

Сервер Электронной Подписи

"КриптоПро DSS"

Руководство разработчика

Содержание

Руководство разработчика

Аннотация

История версий

КриптоПро DSS

КриптоПро SVS

КриптоПро Центр Мониторинга

Аутентификация и подтверждение операций

Общие сведения об OAuth 2.0

OAuth-клиенты

Предварительная конфигурация

Сценарии авторизации

Описание ошибок

Подтверждение произвольных операций

Отображаемые данные

Время жизни транзакции

Внешняя аутентификация

Разное

Производительность

Основные сведения

Аппаратная конфигурация

Пропускная способность сети

Пропускная способность хэширования

Производительность подписи

REST API Сервиса Подписи

Создание запроса на сертификат

Подтверждение операций

Потоковая обработка

Примеры запросов на подпись документа

Отображаемая подпись PDF-документов

Примеры шаблонов отображаемой подписи

Асинхронная подпись

Конечные точки

Типы данных

REST API Сервиса Управления Пользователями (UMS)

- Конечные точки

- Типы данных

REST API Сервиса Аудита

- Конечные точки

- Типы данных

REST API Сервиса Сведений об Операциях

- Конечные точки

- Типы данных

REST API Сервиса Обработки Документов

- Конечные точки

- Типы данных

Sim-auth

- REST API

myDSS

- Лицензирование myDSS

- Назначение и управление аутентификацией через myDSS

- Выпуск сертификата пользователя

- Подтверждение операций через myDSS

- Глубинные ссылки (Deer Links)

DSS SDK

- Назначение и управление аутентификацией через DSS SDK

- Подтверждение операций через DSS SDK

- Выпуск сертификата пользователя

Подтверждение операций по СМС

- Предварительная настройка Администратором DSS

- Назначение и управление аутентификацией

- Подтверждение операций

Разное

- Сервис обнаружения

- Подпись HTTP-сообщений

- Преобразование утверждений

Подсистема оповещения

- Push-плагин

Руководство Разработчика КриптоПро DSS

Настоящий документ содержит Руководство Разработчика Сервера Электронной Подписи (СЭП) «КриптоПро DSS». СЭП «КриптоПро DSS» используется для создания электронной подписи, шифрования документов, а также для централизованного защищенного хранения закрытых ключей Пользователей. Для хранения в СЭП сертификатов и закрытых ключей Пользователей в зашифрованном виде, а также для реализации криптографических операций используется [ПАКМ «КриптоПро HSM»](#).

Документ предназначен для разработчиков как руководство по настройке работы СЭП «КриптоПро DSS» через программные интерфейсы.

Руководство Разработчика КриптоПро DSS

Настоящий документ содержит Руководство Разработчика Сервера Электронной Подписи (СЭП) «КриптоПро DSS». СЭП «КриптоПро DSS» используется для создания электронной подписи, шифрования документов, а также для централизованного защищенного хранения закрытых ключей Пользователей. Для хранения в СЭП сертификатов и закрытых ключей Пользователей в зашифрованном виде, а также для реализации криптографических операций используется [ПАКМ «КриптоПро HSM»](#).

Документ предназначен для разработчиков как руководство по настройке работы СЭП «КриптоПро DSS» через программные интерфейсы.

История версий КриптоПро DSS

Версия 2.0.3272

Сертифицированная сборка СЭП КриптоПро DSS 2.0.3272 для Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, Windows Server 2019.

Новые возможности

- *idsrv*: Добавлена возможность смены логина пользователя (DSS-2573)
- *web, idsrv*: При создании нового экземпляра сервиса автоматически регистрируются плагины для отображения документов (DSS-2558)
- *idsrv*: Добавлена сервис получения статистики операций (v2 API) (DSS-2571)
- *audit*: Упрощено заполнение полей сервиса отчётов (DSS-2564)
- *idsrv*: В Веб-интерфейсе ЦИ добавлена возможность создания пользователя по шаблону (DSS-2563)
- *idsrv*: В Веб-интерфейс ЦИ добавлена возможность выбора системы при создании QR-кода DSS SDK (DSS-2555)
- *web*: В интерфейс шифрования/расшифрования добавлен флаг "Документ ФСС" (DSS-2561?)
- *idsrv*: Добавлен плагин для отображения документов более 15 Mb в мобильном приложении myDSS (DSS-2558)

Ошибки

- *idsrv*: Исправлена ошибка отправки QR-кода myDSS по электронной почте (DSS-2569)
- *idsrv*: Исправлена ошибка отмены входа с подтверждением в Веб-интерфейсе (DSS-2557)
- *signserver*: Исправлена ошибка сервиса хэширования (DSS-2559)
- *idsrv*: Исправлена ошибка управления сертификатами пользователя с большим различительным именем X500 (DSS-2567)

Версия 2.0.3210

Новые возможности

- *idsrv*: Добавлен новый метод аутентификации myDSS Client (DSS-1954)

Примечание

myDSS Client - инструментальный (framework) для быстрой разработки приложений с функциями клиента DSS. SDK предоставляет API и графический интерфейс для:

- Управления сертификатами пользователя
- Подтверждения или отклонения подписи документов при работе в ДБО
- Отправки документов на подпись
- Отображения подписываемых документов любого размера
- Просмотра истории операций пользователя
- и др.

- *dss*: Добавлено API v2.

Примечание

API v2 разрабатывалось для myDSS Client, но поддерживает и другие методы аутентификации myDSS, SIM-аутентификация, одноразовые пароли по SMS и Email и др. API v2 предоставляет больше возможностей по управлению операциями (контролю статуса операций, отмене/повтору операций и т.п.), документами (сохранению временному или постоянному, отображению, работе с документами любого размера).

- *dss*: Добавлены новые компоненты КриптоПро DSS.

Примечание

Для API v2 необходимо вернуть 2-е новых компоненты:

- Сервис Обработки Документов
- Сервис Операций

- *idsrv*: Группы пользователей более не привязаны к Центрам Идентификации (DSS-2115)

Примечание

Добавилась возможность создания Оператора DSS, который может управлять учётными записями пользователей в разных Центрах Идентификации.

- *signserver*: Добавлена поддержка RSA-провайдеров с Мастер ключом (DSS-2424)
- *signserver*: Поддержка шифрования XML-шифрования ФСС (DSS-2419).
- *idsrv*: Расширены возможности утилиты создания файлов персонализации SIM-аутентификации (DSS-1530)
- *audit*: Добавлены командлеты для настройки взаимодействия с сервисом Аудита (DSS-2162)

Примечание

Добавлены командлеты

- New-dssXXXaudit
- Remove-dssXXXAudit
- Set-dssXXXAudit

- *audit*: Добавлен возможность настройки блокирующего аудита (DSS-2162)
- *idsrv*: Добавлена возможность настройки Client Credentials протокола OAuth 2.0 (DSS-2027)
- *signserver*: Добавлена поддержка потоковой загрузки документов (DSS-1578)
- *signserver*: Добавлена возможность поиска сертификата по отпечатку (DSS-2517)
- *idsrv*: Добавлена возможность передать время жизни транзакции в запросе (DSS-1549)
- *signserver*: Добавлена возможность ограничить количество ввода ПИН-кода на закрытый ключ (DSS-1905)
- *signserver*: Добавлена возможность установки сертификата с заменой самоподписанного сертификата (DSS-2095)
- *audit*: Добавлен REST API сервиса Аудита (DSS-2074)
- *cloudcsp*: Добавлена возможность создания ключей подписи через Cloud CSP (DSS-2095)
- *cloudcsp*: Поддержана выработка ключа согласования ГОСТ Р 34.12-2015 (DSS-2511)
- *idsrv,webui*: Добавлена возможность настройки оповещения пользователя в Веб-интерфейсе (DSS-2117)
- *lite*: Исправлены ошибка подписи XML, Office на алгоритмах ГОСТ Р 34.10-2012 (DSS-2305)
- *idsrv,webui*: Добавлена настройка и отображение контактной информации Оператора в личном кабинете (DSS-2187)
- *idsrv*: Добавлена поддержка сервисных OAuth-клиентов (DSS-2179)

Примечание

Сервисный OAuth-клиент позволяет пропустить первичную аутентификацию пользователя. Данный клиент позволяет настроить различную аутентификацию пользователей при работе через Веб-интерфейс КриптоПро DSS и интегрированную ИС (ДБО и т.п.)

- *idsrv*: Добавлена возможность отложенной блокировки пользователя (DSS-2407)
- *idsrv*: Добавлена возможность блокировки операторов (DSS-2010)
- *signserver*: Добавлена возможность блокировки операторов (DSS-2010)
- *idsrv*: Добавлен плагин для отображения в myDSS и Веб-интерфейсе запросов на сертификат в формате PKCS#10
- *idsrv*: Поддержаны implicit/hybrid/code сценарии в OIDC RP (DSS-2216)
- *signserver*: Добавлены шаблоны подписи (DSS-1851)

Примечание

- *idsrv*: Добавлена поддержка подписочных лицензий myDSS (DSS-2046)
- *idsrv*: Добавлен REST API Сервиса Управления Пользователями для пользователей (UMS-U) (DSS-2012)
- *idsrv*: Добавлена возможность отмены транзакций (DSS-2299)
- *dss*: Добавлена возможность разворачивать базы данных в предсозданные "болванки" (DSS-1858)

Примечание

Позволяет разворачивать экземпляры сервисов DSS с минимальными правами на SQL-сервере

- *idsrv*: Добавлена возможность аутентификации OAuth-клиента на конечной точке /confirmation (DSS-2327)
- *idsrv*: Добавлена возможность запомнить выбор Центра Идентификации (DSS-2392)
- *idsrv*: Добавлена возможность не отправлять Push уведомления при создании операций (DSS-2347)
- *idsrv*: Добавлена возможность не создавать QR-код для offline подтверждения операций (DSS-2270)
- *idsrv*: Добавлена возможность аутентификации OAuth-клиентов по сертификату (DSS-2327)
- *dss*: Расширены механизмы переноса мастер-ключей в отказоустойчивых конфигурациях (DSS-2503)
- *dss*: Расширены настройки автоопределения формата при отображении подписываемых данных (DSS-2383)
- *dss*: Добавлена поддержка асинхронной подписи (DSS-2363)
- *idsrv*: Добавлена поддержка аутентификации по сертификату в заголовке (NGate) (DSS-2182)

Ошибки

- *signserver*: Исправлен отказ запуска Сервиса Подписи при истекшей лицензии (DSS-2107)
- *signserver*: Исправлено создание откреплённой параллельной подписи на разных алгоритмах (DSS-2064)
- *lite*: Исправлена ошибка в имени поля DSSPreSignResponse -> `CacheObjectId` (DSS-2078)
- *lite*: Исправлена ошибка в контроле размера загружаемого документа (DSS-2083)
- *signserver*: Исправлена ошибка проверки статуса сертификата, если не было запроса на сертификат в модуле Out-of-Band (DSS-2086)
- *signserver*: Исправлена ошибка проверки статуса сертификата, загруженного из PFX (DSS-2147)
- *webui*: Исправлено кэширование параметров формы создания запроса на сертификат (DSS-2190)
- *idsrv, mydss*: Исправлена ошибка при отклонении операции (DSS-1911)
- *idsrv, mydss*: Исправлена ошибка отправки QR-кода myDSS по электронной почте (DSS-1977)
- *lite*: Исправлены ошибки при удалении экземпляра Lite (DSS-2209)
- *signserver*: Исправлена ошибка CMS расшифрования с неподписанными атрибутами (DSS-2214)
- *audit*: Исправлена отображаемая пользователю ошибка при недоступности сервиса Аудита (DSS-2223, DSS-2230)
- *lite, webui*: Исправлена ошибка обновления сертификата через модуль КриптоПро УЦ 2.0 (DSS-2290)
- *lite*: Исправлено создание подписи формата PDF CAdES-T (DSS-2297)
- *lite, webui*: Исправлена работа Веб-интерфейса в режиме ServerAndClient без установленного CAdES Browser Plugin (DSS-2250)
- *idsrv, signserver*: Исправлены ошибки HttpClient при работе под нагрузкой (DSS-2320)
- *webui*: Исправлена ошибка кэширования ПИН-код на закрытый ключ сертификата подписи (DSS-2322)
- *idsrv*: Исправлено заполнение поля `ExpiresIn` в ответах конечной точки confirmation (DSS-2324)
- *audit*: Добавлена утилита для проверки целостности аудита (DSS-2369)
- *signserver*: Исправлены ошибки ввода/удаления лицензии на доп.сервер (DSS-2058)
- *mydss*: Исправлена ошибка при повторе транзакции (DSS-2382)
- *idsrv*: Исправлена ошибка при многопоточном изменении УЗ пользователей (DSS-2381)
- *mydss*: Исправлена ошибка в командлете `Set-MyDssPushServiceProperties` (DSS-2395)
- *signserver*: Исправлено возобновление сертификата, приостановленного через КриптоПро УЦ 2.0 (DSS)
- *mydss*: Исправлена ошибка при отправке callback при отмене транзакции (DSS-2416)
- *webui*: Исправлены форматы даты запроса на отзыв, приостановление, возобновление (DSS-2429)
- *idsrv*: Добавлен плагин для отображения XML-документов (DSS-2498)
- *mydss*: Исправлена утечка хэндлов крипто-провайдеров (DSS-2105)
- *idsrv*: Исправлена ошибка проверки маркера доступа при работе под нагрузкой (DSS-1698)

- *idsrv*: Исправлены ошибки EF при работе с большими бинарными данными (DSS-2502)
- *idsrv*: Добавлены механизмы повторной отправки сообщений в случае ошибок (DSS-2456)

Изменения

- *idsrv*: Добавлено логирование в SMPP-плагин. Параметр `Logfile` (DSS-2055)
- *lite*: Улучшена обработка ошибок при некорректных параметрах запроса (DSS-2089)
- *lite*: Исправлены ошибки кэширования параметров подписи между запросами PreSign и PostSign (DSS-2081)
- *idsrv, mydss*: Время жизни транзакции на myDSS Server передаётся с Центра Идентификации (DSS-2090)

Примечание

Параметр `SignedHashValue` более не является обязательным параметром запроса.

- *lite*: Исправлено поведение когда при разворачивании экземпляра Lite по умолчанию устанавливается требование двухсторонней аутентификации на IIS (DSS-2041)
- *_lite*: Добавлена демонстрационная лицензия при разворачивании экземпляра Lite (DSS-1646)
- *signserver*: Исправлена ошибка при работе с демонстрационной и полноценной лицензией на Сервис Подписи (DSS-2099)
- *audit*: Настройка всех событий аудита перенесена на сервис Аудита (DSS-2124, DSS-2127)

Примечание

Включение отключение всех событий аудита производится на сервиса Аудита. Командлеты `*-DssAnalyticsEvent`

- *idsrv*: Улучшение контроля отправки одноразовых сообщений (DSS-2070)
- *idsrv*: Улучшено логирование и аудит ошибок подтверждение транзакций (DSS-2052)
- *idsrv*: Единственный заданный параметр `redirect_uri` стал опциональным в запросах аутентификации (DSS-1850)
- *idsrv*: Добавлена поддержка отпечатков сертификатов в формате BASE64URL в JWT-токенах (DSS-1807)
- *audit*: Операторам DSS и Read-Only операторам разрешено формирование отчётов на сервисе Аудита (DSS-2013)
- *idsrv*: Выбранный IDP запоминается только после успешной аутентификации в нём (DSS-2211)
- *lite*: Добавлены более строгие проверки значения подписи в методе PostSign (DSS-2206)
- *idsrv*: `post_redirect_logout_uri` заполняется относительным, а не абсолютным адресом (DSS-2316)
- *webui*: Взаимодействие Веб-интерфейса подписи с Сервисом Подписи переведено на REST API и OAuth 2.0
- *idsrv*: В оповещение оператора добавлены сведения о пользователе, которым он управлял (DSS-2252)
- *idsrv*: Разделяемые секреты OAuth-клиентов создаются в формате в BASE64URL (DSS-2300)
- *signserver*: В настройки модулей УЦ добавлен режим проверки на отзыв `RevocationFlag` (DSS-2339)
- *idsrv*: Политика оповещения настраивается только на уровне групп и пользователей (DSS-2173)
- *webui*: Добавлены сведения о статусе сертификата на страницу расшифрования (DSS-2338)
- *dss*: КриптоПро DSS переведён на .NET Framework 4.8 (DSS-2343)
- *webui*: Отображение результата отправки запроса на отзыв/приостановление/возобновление (DSS-2352)
- *webui*: Добавлена защита от CSRF (DSS-2385)
- *idsrv,signserver*: Улучшена диагностика процесса отправки оповещений (DSS-2175)
- *signserver*: Добавлена поддержка шаблонов сертификатов с одинаковыми именами (КриптоПро УЦ 2.0) (DSS-2077)
- *idsrv*: Убрана зависимость скачивания Jwks от параметра `ShowInUi` (DSS-2066)
- *idsrv*: Исправлена отображаемая ошибка при валидации resource в OAuth 2.0 (DSS-1855)
- *signserver*: Улучшена диагностика ошибок в невалидных запросах (DSS-1671)
- *audit*: Добавлены события об удалении каждого сертификата при удалении всех сертификатов (DSS-2436)
- *signserver*: Добавлена возможность отображения исходного документа при подтверждении операции соподписи (DSS-2505)
- *frontend*: Улучшена фильтрация сертификатов по типу работы сервиса подписи (DSS-2480)
- *signserver*: Выбор ООВ-обработчика УЦ при установке PFX стал детерминированным (DSS-2463)
- *mydss*: Расширена трассировка myDSS Server (DSS-2375)
- *audit*: В аудит добавлена запись идентификатора OAuth-клиента (DSS-1544)

Версия 2.0.2882

Новые возможности

- *idsrv, _powershell*: Введён новый тип лицензий КриптоПро myDSS (DSS-2024)

Примечание

Подробнее лицензирование КриптоПро myDSS описано [здесь](#)

Версия 2.0.2828

Новые возможности

- *powershell*: Добавлена возможность загрузки данных о стороннем ЦИ через метаданные OpenID Connect 1.0 (DSS-1874)
- *idsrv*: Добавлена возможность передавать бинарные данные при подтверждении произвольной операции (DSS-1550)
- *signserver*: Добавлена возможность соподписи по хэш-значению (DSS-1775)
- *signserver*: Добавлена поддержка произвольных преобразований в XMLDSig (DSS-1796)

Примечание

Регистрация преобразований выполняется через командлеты *-DssSignServerTransformPlugin В состав дистрибутива включены реализации трансформов:

- <urn:xml-dsig:transformation:v1.1>
- <http://www.w3.org/2002/06/xmldsig-filter2>
- *audit*: Добавлена поддержка блокирующего аудита (DSS-1885)
- *signserver*: Добавлена поддержка Xml-шифрования (DSS-1802)
- *idsrv*: Sim-auth: добавлена поддержка счётчика использования ключей и имени профиля безопасности (DSS-1882, DSS-1883)
- *idsrv*: Поддержка SAML в протоколе TokenExchange (DSS-1896)
- *signserver*: Поддержка соподписи в пакетном режиме (DSS-1898)
- *idsrv, webui*: Поддержка различных номеров телефонов и адресов электронной почты:

Примечание

- Появилась отдельная страница с контактной информацией пользователя. На ней можно добавлять неограниченное количество контактов: адресов электронной почты и номеров телефона. Лишние или устаревшие контакты можно удалять. Уникальность контактов регулируется параметрами RequireUniqueEmail и RequireUniquePhone в конфигурации ЦИ (эти параметры настраиваются командлетами Get-/Set-DssAccountPolicy);
- Для многих вариантов использования, например, в качестве логина, контакт должен быть подтвержден. Требование подтверждать контакт с помощью одноразового пароля определяется параметрами PhoneConfirmation, PhoneConfirmationByOperator, EmailConfirmation, EmailConfirmationByOperator в конфигурации ЦИ (параметры настраиваются командлетами Get-/Set- DssStsProperties);
- На странице с контактной информацией для каждого контакта можно выбрать будет ли он использоваться для оповещения;
- На странице настроек аутентификации в разделе «Методы первичной аутентификации» в подразделе «Способы входа» один из подтвержденных контактов каждого типа (телефон/e-mail) может быть назначен для использования в качестве логина (если соответствующий тип идентификатора разрешен в настройках ЦИ; параметр AvailableIdentifiers, командлеты Get/Set-DssStsProperties);
- Одноразовые пароли для вторичной аутентификации, код активации MobileAuth, сброшенный пароль пользователя при работе от оператора и т.п. теперь могут быть отправлены только на подтвержденный контакт. Контакт будет предложено выбрать из списка подтвержденных контактов или, в случае отсутствия таковых, будет предложено

перейти на страницу управления контактами пользователя;

- При работе от оператора, в таблице с данными о пользователях теперь отображаются все телефоны и электронные адреса. Контакт, используемый для идентификации, будет выделен жирным шрифтом;
- Для вторичной аутентификации с помощью отправки одноразовых паролей на телефон или почту в DSS создается токен аутентификации, который отображается в таблице на странице «Средства аутентификации» при работе от оператора. Новые типы токенов: SmsOtp и EmailOtp.

- *powershell*: Добавлены командлеты для управления списком TSP-службы (DSS-1908)
- *signserver*: Добавлена возможность для модуля УЦ через powershell задать файл со списком расширений, которые будут добавлены в запрос на сертификат (DSS-1937)

Примечание

В командлеты Set-DssEnrollment*** добавлен параметр ExtensionsConfig

- *signserver*: Добавлена возможность передать список расширений, которые будут добавлены в запрос на сертификат, через REST API (DSS-1937)
- *powershell*: Добавлена возможность группировки и разгруппировки криптопровайдеров (DSS-1940)
- *powershell*: Добавлены командлеты для перезапуска пуллов приложений (DSS-1950)

Примечание

Добавлены командлеты Restart-Dss***Instance

- *ums*: В REST API добавлена возможность получения расширенных сведений о ключе myDSS (DSS-1967)
- *install*: Создание резервных копий баз данных опционально при обновлении DSS (DSS-1801)
- *signserver*: В REST API добавлены методы хэширования документов (DSS-1298)

Ошибки

- *signserver*: Исправлен ошибка NRE при изменении статуса сертификата (DSS-1791)
- *myDSS*: Исправлены ошибки при запуске myDSS Push Service (DSS-1752)
- *myDSS*: Исправлена ошибка настройки FCM-плагины (DSS-1786)
- *signserver*: Исправлены ошибки ввода ПИН-код с кириллическими символами (DSS-1778)
- *signserver*: Исправлен ошибка NRE при создании запроса на сертификата на OOB (DSS-1794)
- *idsrv*: Исправлена ошибка при вводе одноразовых паролей (DSS-1793)
- *signserver*: Исправлено удаление всех запросов и сертификатов через REST API (DSS-1799)
- *webui*: Исправлена ошибка выбора криптопровайдера при создании запроса на сертификат (DSS-1776)
- *idsrv*: Исправлена ошибка создания файла персонализации на мастер ключе ГОСТ 2012 (DSS-1803)
- *idsrv*: Исправлено удаление учётной записи и связанных данных пользователя (DSS-1798)
- *idsrv*: Исправлена ошибка при назначении внешнего логина в OpenID Connect 1.0 провайдере (DSS-1857)
- *idsrv*: Исправлен поиск ключа проверки маркера доступа при входе через Microsoft Account (DSS-1874)
- *audit*: Исправлена ошибка подписи журнала аудита без запуска службы AnalytixService (DSS-1601)
- *audit*: Исправлена ошибка формирования отчёта о пользователях, учитывающего операторов (DSS-1612)
- *idsrv*: Исправлены ошибки формирования событий UserAccountChanged, AuthenticationCompleted, RevokeRequestContentRequested, UserEmailChangeFail, CertificateRenewWithConfirmation, CertificateRenewWithConfirmationFail (DSS-1623, DSS-1757, DSS-1913)
- *idsrv, signserver*: Исправлены ошибки аудита изменения настроек (DSS-1693)
- *signserver*: Исправлена ошибка изменения статуса сертификата, выпущенного на OOB-модуле (DSS-1708)
- *webui*: Исправлено поведение при попытке создать запрос на сертификат, при недоступных модулях УЦ (DSS-1714)
- *ums*: Исправлено сообщение об ошибке при получении/задании номера телефона, если метод аутентификации отключен (DSS-1749)
- *ums*: Исправлена ошибка задание номера телефона пользователя с пробелами (DSS-1756)
- *webui*: Исправлена и улучшена верстка Веб-интерфейса Сервиса Подписи (DSS-1773, DSS-1915, DSS-1945)
- *webui*: Исправлено поведение при попытке создать запрос на сертификат, при недоступных криптопровайдерах (DSS-

1840)

- *idsrv*: Исправлено получение сопровождающего текста при аутентификации через myDSS (DSS-1792)
- *webui*: Не отображается поле для адреса TSP-службы при AllowThirdPartyTsp = false (DSS-1795)
- *powershell*: Добавлены проверки на вводимые параметры в командлетах *-DssConfirmationPolicy или *-DssAccessPolicy (DSS-1806)
- *idsrv*: Исправлена ошибка при отключении лицензии через таблицу с токенами (DSS-1810)
- *signserver, idsrv*: Исправлен сброс максимально допустимого размера сообщения после обновления (DSS-1817)
- *webui*: Исправлена ошибка подписи хэш-значения (DSS-1820)
- *signserver*: Исправлена ошибка CAdES подписи при закрытом списке TSP-служб (DSS-1825)
- *powershell*: Исправлена ошибка запуска командлета Set-DssStsEndpointGlobalSettings (DSS-1837)
- *signserver*: Исправлена ошибка при удалении OOB запроса при удаленном криптопровайдере (DSS-1838, DSS-1839)
- *ums*: Исправлена ошибка кодировки при формировании DN в личном кабинете оператора (DSS-1842)
- *signserver*: Исправлена логика работы с криптопровайдером по умолчанию (DSS-1848)
- *myDSS*: Исправлена повторная регистрация модулей оповещения при разворачивании на доп.сервере (DSS-1856)
- *idsrv*: Исправлено применение Idp ClaimsTransformRules при добавлении внешнего логина через веб-интерфейс (DSS-1857)
- *powershell*: Исправлена ошибка задании/изменении сервисных сертификатов с пробелами (DSS-1877)
- *idsrv*: Исправлена ошибка саморегистрации с пустым различительным именем (DSS-1880)
- *idsrv*: Исправлено поведение формы создания пользователя оператором после выбора группы (DSS-1888)
- *audit*: Исправлено логирование параметра подписи OriginalDocument (DSS-1892)
- *webui*: Исправлен порядок отображения компонентов имени в шаблоне УЦ (DSS-1881)
- *ums*: Исправлено автоматическое подтверждение номера телефона при создании пользователя оператором (DSS-1894)
- *powershell*: Исправлена ошибка при получении сроков действия мастер ключей (DSS-1900)
- *powershell*: Исправлена ошибка спец.символов в файле с шаблонами сертификатов OOB (DSS-1903)
- *signserver*: Исправлено заполнение FaultReason в SOAP Fault (DSS-1904)
- *powershell*: Исправлена ошибка ввода лицензии на доп.сервере (DSS-1910)
- *idsrv*: Исправлена ошибка формирования выжимки для dtbs в кодировке отличной от UTF8 (DSS-1912)
- *powershell*: Исправлена ошибка сохранения настроек события в файл (DSS-1916)
- *idsrv*: Исправлена ошибка сброса пароля при первом входе (DSS-1961)
- *signserver*: Исправлена ошибка подсчёта количества пользователей (DSS-1961)
- *installer*: Исправлена ошибка при которой не учитывались SQL учётные данные при выполнении некоторых командлетов при обновлении (DSS-1960)
- *audit*: Исправлена ошибка записи хэш-значения (DSS-1939)

Изменения

- *eventlog*: добавлены новые события в EventLog (DSS-1874):
 - IssuerSigningTokensDownloadSucceeded (65110),
 - IssuerSigningTokensDownloadFailed (65111),
 - IssuerSigningTokensImportFailed (65112).
- *powershell*: в командлеты Add/Set-DssIdentityProvider добавлен необязательный параметр -AuthEndpointType (DSS-1874)
- *powershell*: Поддержан BASE64-формат отпечатка сертификата Idp
- *signserver*: Расширена валидация параметров PDF-подписи (DSS-1824)
- *signserver*: Параметры запросов к REST API стали регистро независимыми (DSS-1834)
- *signserver*: При попытке подписи документов превышающих заданный размер возвращается ошибка HTTP 400 max_request_length_exceeded (DSS-1671, DSS-1835)
- *idsrv*: Расширено логирование при недоступном BackChannelUrl (DSS-1836)
- *powershell*: Расширен набор проверок в командлете Test-DssCryptoProvider (DSS-1841)
- *signserver, idsrv*: Улучшен контроль сроков действия мастер ключа и ключе пользователей (DSS-1841)
- *myDSS*: Поддержан TLS-соединение между InteractionService и Service (DSS-1842)

- *ums*: Расширена валидация номеров телефона (DSS-1844)
- *idsrv*: В плагин для Callback'a вызывающей системы добавлена поддержку двустороннего TLS (DSS-1852)
- *powerhsell*: Добавлена возможность сбросить значение параметра AuthorityName в Set-DssEnrollment20 (DSS-1948)

Примечание

В настройке плагина добавлена параметр 'ctt', в котором можно указать отпечаток сертификата аутентификации

- *idsrv*: Расширена валидация параметра Resource при попытке получения OAuth токена (DSS-1859)
- *webui*: Добавлена фильтрация доступных криптопровайдеров в зависимости от выбранного шаблона сертификата (DSS-1862)
- *idsrv*: Ускорена аутентификация по сертификату (DSS-1864)
- *signserver*: Улучшена обработка невалидных параметров в запросах к REST API (DSS-1876)
- *signserver*: При работе с XML-документами запрещена загрузка внешних сущностей (DSS-2025)
- *idsrv*: Согласование параметров жизни токена в OAuth-клиенте с глобальными настройками сервиса (DSS-1879)
- *audit*: Добавлено явное задание адреса ЦИ в настройках сервиса (DSS-1906)
- *powershell*: Улучшен поиск операторов по типу в командлете Get-DssIdentityOperators (DSS-1887)
- *myDSS, idsrv*: Улучшена обработка отклонённых операций (DSS-1899)
- *audit*: Расширены параметры события создания запроса на сертификат (DSS-1919)
- *powershell*: Добавлена трассировка миграции базы данных (DSS-1920)
- *signserver, idsrv*: Улучшен мониторинг доступности криптопровайдеров (DSS-1940)

Примечание

Трассировка пишется в каталог %APPDATA%\..\local\temp

- *audit*: При удалении всех сертификатов в аудит пишутся события удаления для каждого из сертификатов (DSS-1952)
- *audit*: Расширен набор данных записываемых для событий подписи и подписи с подтверждением (DSS-1933)
- *signserver*: В оповещение об отзыве сертификата добавлена отображаемая причина отзыва (DSS-1849)

Версия 2.0.2636

Новые возможности

- *idsrv*: Поддержка аутентификации в сторонних ЦИ по протоколу OpenId Connect 1.0 (DSS-1373)
- *webui*: Добавлена возможность зашифрования на собственном сертификате (DSS-1730)
- *ums*: Разрешено добавлять номер телефона или email в идентификатор типа Login (DSS-1717)
- *convert*: Добавлена возможность явно задавать кодировку в плагинах конвертации (DSS-1696)
- *idsrv*: Добавлена новая роль Оператора DSS - ReadOnly-Оператор (DSS-1683)
- *webui*: Контроль формата компоненты имени пользователя "Электронная почта" (DSS-1675)
- *ums*: Возможность повторной отправки кода активации myDSS пользователю (DSS-1668)
- *signserver*: Добавлена возможность импорта сертификата из pfx в REST API (DSS-1596)
- *signserver*: Добавлена возможность указания даты начала и окончания действия сертификата (DSS-1567)
- *idsrv*: Добавлена поддержка долговременных сессий (DSS-1405)
- *powershell*: Унифицированы идентификаторы проверяющих сторон (DSS-1560)
- *powershell*: Добавлена возможность экспорта/импорта конфигурации (DSS-1724)
- *powershell*: Добавлен аудит изменения настроек (DSS-1693)
- *signserver*: Добавлена поддержка открытых и закрытых списков TSP-служб (DSS-1480)

Примечание

Настройка регулируется флагом AllowThirdPartyTsp в командлете Set-DssProperties. Если флаг взведён, то Сервис Подписи может принимать в параметрах подписи произвольный адрес службы TSP. Если флаг сброшен, то адреса TSP служб должны быть явно заданы в настройках Сервиса Подписи.

- *signserver*: Добавлена возможность не отображать модули УЦ в Веб-интерфейсе

Примечание

В настройках модулей УЦ добавлен параметр ShowInUi.

- *signserver*: Добавлена поддержка AllowUserMode при работе с КриптоПро УЦ 2.0 R10 и выше.

Примечание

Режим работы AllowUserMode - подпись запросов к КриптоПро УЦ 2.0 на действующем ключе пользователя. Для версий КриптоПро УЦ 2.0 ниже R10 данный режим не поддерживается.

- *signserver*: В политике Сервиса Подписи передаётся информация о допустимых алгоритмах ключа для каждого шаблона сертификата (DSS-1634)
- *signserver*: Добавлена возможность выполнения соподписи документа с использованием хэш-значения (DSS-1775)
- *webui*: Встроенный Веб-интерфейс SVS переведён на REST API (DSS-1736)

Внимание!

Если в Веб-интерфейсе DSS была настроена интеграция с Сервисом Проверки Подписи, то необходимо обновить SVS до версии 2.0.2626 и старше.

Ошибки

- *powershell*: Исправлена ошибка переназначения плагинов в модуле оповещения (DSS-1771)
- *powershell*: Исправлена ошибка назначения ключа аутентификации FCM (DSS-1786)
- *powershell*: Исправлена работа командлета Get-Dss***Administrator (DSS-1695)
- *powershell*: Исправлена ошибка удаления базы данных Сервиса Аудита (DSS-1755)
- *signserver*: Исправлена ошибка при получении запроса на сертификат по невалидному идентификатору (DSS-1762)
- *idcrv*: Исправлена ошибка формирования события аудита UserPhoneChanged (DSS-1532)
- *webui*: Исправлена и улучшена верстка Веб-интерфейса Сервиса Подписи (DSS-1746, DSS-1732, DSS-1707, DSS-1702, DSS-1676, DSS-1669, DSS-1660, DSS-1656, DSS-1650, DSS-1642, DSS-1631, DSS-1630, DSS-1440, DSS-1399, DSS-1388, DSS-651, DSS-592)
- *webui*: Исправлена ошибка при просмотре сертификатов без расширения EKU (DSS-1745)
- *powershell*: Исправлена ошибка удаления базы данных при удалении экземпляра ЦИ (DSS-1737)
- *ums*: Исправлена регистрация пользователя с указанием группы (DSS-1726)
- *audit*: Исправлена ошибка загрузки Сервиса Аудита при недоступных плагинах (DSS-1721)
- *simauth*: Исправлена ошибка загрузки Центра Идентификации при недоступных криптопровайдерах (DSS-1639)
- *signserver*: Корректная обработка уровня сертификации CERTIFIED_NO_CHANGES_ALLOWED в PDF-подписи (DSS-1719)
- *webui*: Исправлено поведение Веб-интерфейса при недоступных модулях УЦ (DSS-1714, DSS-695)
- *idsrv*: Исправлено перенаправление при подтверждении входа в личный кабинет пользователя (DSS-1659)
- *idsrv*: Исправлены ошибки при записи событий аудита (DSS-1643)
- *signserver*: Исправлена ошибка создания ключа согласования (DSS-1624)
- *idsrv*: Исправлена ошибка проверки сертификата аутентификации (DSS-1622)
- *signserver*: Исправлена ошибка проверки маркера доступа при многопоточной работе (DSS-58)
- *idsrv*: Исправлена форма ввода кода подтверждения myDSS (DSS-1617)
- *powershell*: Исправлена ошибка назначения параметра RequireMutualHttps в командлете Set-DssStsProperties (DSS-1611)
- *idsrv*: Исправлено подтверждение VKO через myDSS (DSS-1609)
- *idsrv*: Корректная обработка параметра response_mode при формировании ответа с ошибками авторизации (DSS-1607)

Изменения

- *powershell*: Подпись по хэш-значению разрешается/запрещается с помощью параметра AllowHashSigning в командлете Set-DssProperties (DSS-1614)
- *signserver*: Улучшена проверка параметров при создании транзакции (DSS-1640)
- *ums*: Улучшена диагностика ошибок (DSS-1667)
- *webui*: Оператору DSS отображаются средства аутентификации пользователей только из своих групп (DSS-1673)
- *idsrv*: Улучшен порядок назначения и освобождения лицензий CloudCsp (DSS-1681, DSS-1655, DSS-1654, DSS-1628, DSS-1626, DSS-1621)
- *ums*: Удаление токенов аутентификации (MobileAuth, SimAuth, CloudCsp) при удалении пользователя (DSS-1692)
- *simauth*: Передача OperationResult в callback'e об истечении срока действия транзакции (DSS-1685)
- *simauth*: Поддержка статуса USER_TIMEOUT от OTA платформы (DSS-1694)
- *simauth*: Номер телефона для отправки SimAuth-сообщений сохраняется в параметрах SIM-карты (DSS-1700)
- *powershell*: В командлетах Set-DssXXXTracing параметры ServiceModelListenerLogFile, ServiceModelMessageLoggingListenerLogFile сделаны опциональными (DSS-1706)
- *webui*: Не отображается список сторонних ЦИ, состоящий из одного элемента (DSS-1710)
- *idsrv*: Определение долговременной сессии в обработке OAuth (DSS-1725)
- *convert*: Улучшена диагностика ошибок при загрузке плагинов преобразования (DSS-1721)
- *signserver*: Добавлен корректный код возврата при отзыве отозванного сертификата: invalid_certificate_status (DSS-1733)
- *powershell*: Улучшено отображение параметра OrcActions в выводе командлета Get-DssConfirmationPolicy (DSS-1744)
- *webui*: Увеличена скорость работы Веб-интерфейса Сервиса Подписи (DSS-1713, DSS-1747)
- *idsrv*: Добавлена возможность преобразования входящих утверждений при аутентификации через сторонний ЦИ.
- *simauth*: Исправлена ошибка при формировании сообщения об истечении транзакции SimAuth (DSS-1685)
- *powershell*: Удалён командлет Set-DssStsWsFederationSettings (DSS-1373)
- *powershell*: Добавлен командлет Set-DssIdentityProviderWsFedEndpoint (DSS-1373)
- *powershell*: Добавлен командлет Set-DssIdentityProviderOidcEndpoint (DSS-1373)
- *powershell*: из командлетов Add/Set-DssIdentityProvider удалён параметр -Uri (DSS-1373)
- *powershell*: из командлетов Add/Set-DssIdentityProvider удалён параметр -NameClaimType (DSS-1373)
- *powershell*: из командлетов Add/Set-DssIdentityProvider удалён параметр -RoleClaimType (DSS-1373)
- *powershell*: из командлетов Add-DssIdentityProvider удалён параметр -ShowInUi (DSS-1373)
- *powershell*: из вывода командлета Get-DssIdentityProvider удалено поле PrivateCabUri (DSS-1373)
- *powershell*: в командлет Set-DssIdentityProvider добавлен параметр -ClaimsTransformationRulesFile (DSS-1373)
- *powershell*: в вывод командлета Get-DssIdentityProvider добавлены поля (DSS-1373):
 - -ClaimsTransformationRules,
 - -AuthEndpointType,
 - -AuthEndpointSettings.
- *eventlog*: добавлены новые события в EventLog (DSS-1373):
 - IssuerDisabled (65101),
 - IssuerSigningKeyLoaded (65102),
 - IssuerSigningKeyMissing (65103),
 - IssuerSigningKeyResolved (65104),
 - IssuerSigningKeyUnResolved (65105),
 - IssuerSigningTokenResolved (65106),
 - IssuerSigningTokenUnResolved (65107),
 - IssuerNameResolved (65108),
 - IssuerNameUnResolved (65109),
 - LexerNoViableAltException (65001),
 - ParserSyntaxError (65002).

История версий КриптоПро SVS

Версия 2.0.2658

Новые возможности

- *svs*: Добавлен REST API (DSS-1479)
- *svs*: Добавлена возможность регистрации произвольных XMLDSig преобразований (DSS-1796)
- *svs*: Добавлена возможность регистрации дополнительных проверок сертификатов (DSS-1833)
- *svs*: Добавлена проверка сертификата требованиям к квалифицированному сертификату (DSS-1833)
- *svs*: Добавлена поддержка групповой политики OCSP Client (CADES-1643, DSS-1302)
- *svs*: Добавлена возможность проверки CAdES подписи по хэш-значению (DSS-1339)
- *svs*: Добавлена поддержка Windows Server 2019
- *tsl*: Добавлена загрузка и установка корневого сертификата Минкомсвязи ГОСТ Р 34.10-2012
- *tsl*: Добавлены различные режимы загрузки, установки и удаления CRL (DSS-1220)
- *tsl*: Добавлена возможность распараллеливания загрузки сертификатов и CRL (DSS-1358)

Ошибки

- Исправлена проверка наличия в подписи исходного документа для документов более 7Мб (DSS-1758)

История версий КристоПро Центр Мониторинга

Версия 2.0.2992

Новые возможности

- Мониторинг счётчика производительности: изменён графический интерфейс теста (DSS-1990)
- Тесты веб-служб (HTTP): в сообщения об ошибках включены адреса тестируемых служб (DSS-2000)
- Тесты веб-служб (HTTP): возможность указания таймаута для запросов (DSS-2006)
- Новые тесты: Тест проверки срока действия ClientSecret (DSS-2111), Тест криптопровайдера Сервиса Аудита (DSS-2026), Тест криптопровайдера Центра Идентификации DSS (DSS-2026), Тест NGate (DSS-1543)
- Возможность экспорта и импорта конфигурации тестов и экземпляров тестирования

Ошибки

- Рассылка уведомлений: Исправлена ошибка при некоторых событиях по email, вызывающие ошибку "значение не может быть неопределённым" (DSS-1965)
- Выбор сертификатов: исправлена ошибка, при которой окно выбора сертификатов не отрисовывалось сверху всех окон
- Исправлена ошибка исчезновения несохранённых тестов, при сохранении другого теста (DSS-2136)
- Тест CRL: Исправлена ошибка, при которой CRL, которые не начали действовать, считались корректными
- Прочие правки: Тест лицензий Сервиса Подписи, Тест лицензий Центра Идентификации, Проверка количества сообщений в системных очередях ЦР, Тест состояния очередей Центра Регистрации УЦ

Известные проблемы

- Тест лицензий Центра Идентификации, Тест лицензий Сервиса Подписи: тест не учитывает встроенные лицензии на некоторых версиях DSS

Аутентификация и подтверждение операций

Раздел содержит справочную информацию о процессах аутентификации и подтверждения операций на Центре Идентификации DSS и включает в себя:

- [краткое описание протокола OAuth 2.0](#);
- [сценарии авторизации на ЦИ DSS по протоколу OAuth 2.0](#);
- вспомогательные статьи.

Список вспомогательных статей:

- [Подсказка логина](#);
- [Режимы возврата ответа о результате авторизации](#);
- [Отмена запроса](#);
- [Ошибки авторизации](#);
- [Взаимодействие с пользователем](#);
- [Обмен маркеров](#).

Протокол авторизации OAuth 2.0

OAuth 2.0 ([RFC 6749](#)) является фреймворком для авторизации, позволяющим получить сторонним приложениям ограниченный доступ к ресурсам HTTP-сервиса.

Стандарт OAuth 2.0 определяет следующие четыре роли:

- *владелец ресурса* - сущность, обладающая правом на выдачу доступа к защищенным ресурсам. В случае, если владелец является человеком, его называют конечным пользователем;
- *сервер ресурсов* - сервер, содержащий защищаемые ресурсы и обладающий возможностью получения и формирования ответа на запросы к защищаемым ресурсам посредством использования маркера доступа;
- *клиент* - приложение, осуществляющее доступ к защищенным ресурсам от имени Владельца. Термин "клиент" явно не определяет какое-либо конкретное исполнение (будь то сервер, персональный компьютер или мобильное приложение);
- *сервер авторизации* - сервер, осуществляющий выпуск маркеров доступа для клиентских приложений после успешной аутентификации и авторизации Владельца ресурсов.

Соответствие сущностей DSS ролям OAuth 2.0 имеет следующий вид:

- в качестве владельца ресурсов выступает Пользователь КриптоПро DSS;
- в качестве сервера ресурсов выступают компоненты DSS, содержащие данные Пользователя (например, Сервис Подписи, Сервис Аудита, Хранилище Документов и т.п.)
- клиентом является приложение, осуществляющее операции в DSS. Приложением может являться браузер, толстый клиент или мобильное приложение.
- ЦИ DSS выступает в роли сервера авторизации.

Согласно OAuth, Клиентское Приложение запрашивает доступ к ресурсу, находящемуся в ведении Пользователя. Вместо использования учетных данных Пользователя, Приложение получает **маркер доступа** - строку, включающую в себя атрибуты доступа к ресурсу. Данные маркеры выпускаются Центром Идентификации КриптоПро DSS с разрешения Пользователя (после его аутентификации).

Протокол OAuth обладает возможностью аутентификации не только Пользователя, но и клиентского приложения, осуществляющего доступ к ресурсам. Для этого протокол предусматривает такие параметры как `client_id` и `client_secret`.

`client_id` - это идентификатор клиентского приложения, используемый Центром Идентификации для поиска информации о клиенте.

`client_secret` является аналогом пароля для клиентского приложения и используется для аутентификации клиентского приложения на ЦИ DSS. Компрометация секрета приводит к компрометации всех клиентских приложений, использовавших `client_id`, связанный с данными секретом.

Абстрактная схема взаимодействия описанных в протоколе ролей выглядит следующим образом:



1. Клиентское приложение запрашивает авторизацию у Пользователя.
2. Приложение получает разрешение на авторизацию (access grant), которое представляет из себя данные, описывающее авторизацию Приложения Пользователем. Разрешение на авторизацию может быть представлено одним из четырех типов, описанных в протоколе;
3. Приложение запрашивает маркер доступа у Центра Идентификации, передавая ему разрешение на авторизацию, полученное от Пользователя;
4. Центр Идентификации аутентифицирует клиента и проверяет валидность разрешения на авторизацию. В случае положительного исхода сервер формирует маркер доступа и передает его клиенту;
5. Приложение запрашивает доступ к защищенному ресурсу (например, Сервису Подписи) и аутентифицируется на нем посредством маркера доступа, полученного ранее.
6. Сервер с защищаемым ресурсом валидирует маркер доступа и в случае положительного исхода предоставляет доступ к запрашиваемым ресурсам.

Протокол определяет следующие типы разрешения на авторизацию:

- код авторизации (authorization code);
- неявный (implicit);
- учетные данные владельца ресурса (resource owner);
- учетные данные клиента (не используется КристоПро DSS).

Работа с ЦИ DSS по протоколу OAuth 2.0 может быть описана одним из представленных ниже сценариев ([подробнее](#)). При этом, все сценарии можно разбить на два класса: требующие взаимодействия с пользователем через браузер и не требующие.

Под взаимодействием с пользователем понимается непосредственное взаимодействие Пользователя DSS с Веб-Интерфейсом Центра Идентификации в рамках авторизации клиентского приложения, в то время как сценарии без взаимодействия с Пользователем могут выполняться по схеме Service-Service.

К сценариям, требующим обязательное взаимодействие с пользователем, относятся:

- [Авторизация с использованием кода авторизации](#)
- [Авторизация с использованием типа разрешения Implicit](#)

К сценариям которые могут выполняться без интерактивного взаимодействия с пользователем относят:

- Авторизация с аутентификацией по сертификату с использованием кода авторизации
- Авторизация с использованием учетных данных владельца ресурса
- Получение операторского делегирующего маркера доступа к ресурсу.

Аутентификация OAuth клиента

ПАК КриптоПро DSS обладает возможностью регистрации конфиденциальных OAuth-клиентов. Под конфиденциальным клиентом понимается аутентифицируемый OAuth клиент, обладающий тем или иным фактором аутентификации. КриптоПро DSS позволяет аутентифицировать OAuth-клиенты следующим образом:

- [Аутентификация с использованием пароля](#)
- [Аутентификация по сертификату](#)

Аутентификация с использованием пароля

Данная схема подразумевает наличие у OAuth-клиента пароля, знанием которого обладают только сервер аутентификации и собственно аутентифицируемый клиент.

Для регистрации конфиденциального клиента с паролем необходимо при выполнении командлета `Add-DssClient` необходимо указать параметр `GenerateSecret`. Пароль также может быть добавлен позднее для уже существующих клиентов при помощи командлета Powershell `Add-DssClientSecret` с типом секрета `SharedSecret`.

При формировании запросов к серверу аутентификации клиент может передавать свои аутентификационные данные двумя способами:

- при помощи заголовка `Authentication` HTTP-запроса.
- в теле HTTP-запроса;

В первом случае для HTTP-запроса необходимо сформировать заголовок `Authentication` следующего вида:

```
Basic dGVzdENSaWVudDp0ZXN0U2VjcmV0
```

Идентификатор клиента и секрет преобразовываются следующим образом: `Base64(client_id:secret)`, где `Base64()` это функция преобразования символьной строки в кодировку Base64.

Пример: для OAuth-клиента с идентификатором `testClient` и секретом `testSecret` необходимо сделать следующее:

1. Объединить идентификатор и секрет в строку с разделителем: `testClient:testSecret`;
2. Преобразовать полученную строку с использованием кодировки Base64:
`Base64(testClient:testSecret)=dGVzdENSaWVudDp0ZXN0U2VjcmV0`;
3. Сформировать готовый заголовок: `Authenctication: Basic dGVzdENSaWVudDp0ZXN0U2VjcmV0`.

Во втором случае идентификатор OAuth-клиента и секрет необходимо передавать в теле запроса на аутентификацию. Для идентификатора OAuth-клиента предназначен параметр `client_id`, для секрета - `client_secret` соответственно.

Примечание

В соответствии с RFC 6749, передача аутентификационных данных клиента в теле запроса должна использоваться только для клиентов, которые не могут в полной мере воспользоваться схемой аутентификации Basic HTTP. Для клиентов, которые обладают возможностью аутентификации через Basic HTTP СТРОГО НЕ РЕКОМЕНДУЕТСЯ передавать данные в теле запроса. Дополнительно следует отметить, что данные НИКОГДА не должны передаваться в строке запроса и должны влючаться ТОЛЬКО в тело.

Аутентификация с использованием сертификата

ПАК КриптоПро DSS поддерживает возможность аутентификации OAuth-клиента посредством установления двустороннего защищенного соединения между клиентом и сервером аутентификации. В качестве фактора аутентификации выступает клиентский сертификат, который используется OAuth-клиентом для установления защищенного соединения.

ПАК предусматривает отдельную конечную точку `oauth/token/cert` для аутентификации OAuth-клиентов по клиентскому

сертификату. Конечная точка настроена таким образом, что обязательно требует клиентский сертификат.

Процедура аутентификации выглядит следующим образом:

1. OAuth-клиент инициирует запрос на авторизацию на указанную выше конечную точку;
2. Между сервером и OAuth-клиентом устанавливается двустороннее защищенное соединение;
3. Сервер извлекает `client_id` из тела запроса для идентификации OAuth-клиента;
4. Затем сервер извлекает клиентский сертификат установленного соединения и убеждается, что найденный на предыдущем этапе OAuth-клиент сдержит среди своих секретов извлеченный сертификат;
5. В случае положительного результата сервер осуществляет дальнейшую аутентификацию пользователя на основании информации в теле запроса.

Для того, чтобы зарегистрировать сертификат в качестве клиентского секрета, необходимо воспользоваться командлетом Powershell `Add-DssClientSecret`, указав в нем тип секрета `Certificate`, а сам сертификат в качестве значения параметра `Certificate`.

Предварительная конфигурация

Для обеспечения взаимодействия с DSS по протоколу OAuth необходимо выполнить ряд предварительных действий, которые включают в себя:

- [Получение идентификатора Сервиса Подписи](#);
- [Регистрация OAuth-клиентов на стороне ЦИ DSS](#);

Получение идентификатора Сервиса Подписи

Данный шаг включает в себя получение идентификатора Сервиса Подписи, к которому будет обращаться OAuth-клиент. Этот идентификатор используется при формировании запросов на аутентификацию к ЦИ DSS.

Значение идентификатора можно получить при помощи командлета PowerShell *Get-DssRelyingPartyTrust*. Данный командлет возвращает информацию о доверенных сторонах ЦИ DSS, среди которых содержится идентификатор Сервиса Подписи.

Формат идентификатора ресурса имеет следующий вид: `urn:cryptopro:dss:<componentName>:<displayName>`, где `<componentName>` - тип компонента (SignServer/Frontend), а `<displayName>` - отображаемое имя экземпляра данного компонента.

Пример вывода командлета:

```
Id                : 3
DisplayName        : SignServer
Description        :
EncryptionCertificate :
DisableTokenEncryption : False
ForOperator        : False
AdministrativeUrl   :
Enabled            : True
DisableActAs        : True
SupportsBackChannel : True
BackChannelUrl      : https://<hostname>/SignServer/rest/api/transactions/token
Identities          : {https://<hostname>/SignServer/rest/api, urn:cryptopro:dss:signserver:signserver}
```

По умолчанию, идентификатор Сервиса Подписи имеет следующий вид: `urn:cryptopro:dss:signserver:signserver`;

Регистрация клиента

В соответствии со спецификацией протокола OAuth, для успешной аутентификации Центр Идентификации должен располагать информацией об OAuth-клиентах, которые будут инициировать процедуру доступа к ресурсам.

Непосредственная конфигурация OAuth-клиентов на стороне ЦИ DSS осуществляется при помощи командлетов PowerShell *Add-DssClient*, *Set-DssClient* и *Remove-DssClient*.

Add-DssClient позволяет зарегистрировать новый OAuth-клиент.

Set-DssClient позволяет изменить конфигурацию существующего OAuth-клиента.

Remove-DssClient позволяет удалить зарегистрированный OAuth-клиент из БД ЦИ DSS.

К информации, которая необходима для регистрации OAuth-клиента, относится:

- Идентификатор - значение, используемое для идентификации OAuth-клиента при формировании запросов к ЦИ;

Примечание

В случае, если при регистрации идентификатор не указан, ЦИ DSS сгенерирует его автоматически.

- Имя - отображаемое имя OAuth-клиента;

- Секрет - закрытая информация, позволяющая аутентифицировать клиента;
- Разрешенные сценарии использования - при регистрации OAuth-клиента на стороне ЦИ DSS необходимо явно задать список сценариев, по которым данный клиент может взаимодействовать с ЦИ;
- Адрес перенаправления - адрес, на который будет осуществляться перенаправление ответа ЦИ с полученным токеном.

Сценарии авторизации

Данный раздел содержит в себе описание типовых сценариев взаимодействия клиентов с Центром Идентификации.

Раздел описывает следующие сценарии авторизации клиентского приложения на Центре Идентификации:

- [Авторизация с использованием кода авторизации по протоколу OAuth 2.0](#)
- [Авторизация с аутентификацией по сертификату с использованием кода авторизации по протоколу OAuth 2.0](#)
- [Авторизация с использованием типа разрешения Implicit по протоколу OAuth 2.0](#)
- [Авторизация с использованием учетных данных владельца ресурса по протоколу OAuth 2.0](#)
- [Получение операторского делегирующего маркера доступа к ресурсу](#)

Центр Идентификации ПАК КриптоПро DSS предоставляет следующие конечные точки для аутентификации по OAuth:

- **oauth/authorize** - используется для инициирования процедуры авторизации;
- **oauth/authorize/certificate** - используется для инициирования авторизации с аутентификацией по сертификату;
- **oauth/token** - используется для получения маркера доступа.

Соответствие между конечными точками и сценариями, их использующими, представлено ниже:

СЦЕНАРИЙ АВТОРИЗАЦИИ	ИСПОЛЬЗУЕМЫЕ КОНЕЧНЫЕ ТОЧКИ
Авторизация с использованием кода авторизации	oauth/authorize - запрос кода авторизации oauth/token - получение маркера доступа по коду авторизации
Авторизация с аутентификацией по сертификату	oauth/authorize/certificate - запрос кода авторизации oauth/token - получение маркера доступа по коду авторизации
Авторизация с использованием типа разрешения Implicit	oauth/token - получение маркера доступа (после аутентификации)
Авторизация с использованием учетных данных владельца ресурса	oauth/token - получение маркера доступа
Получение операторского делегирующего маркера доступа к ресурсу	oauth/authorize/certificate - запрос кода авторизации оператором oauth/token - получение маркера доступа оператора, получение делегирующего маркера

Авторизация с использованием кода авторизации по протоколу OAuth 2.0

Работа протокола OAuth 2.0 в режиме с использованием типа разрешения Authorization Code строится вокруг так называемого кода авторизации, представляющего из себя промежуточное звено между Владелец Ресурсов и Клиентским приложением.

Вместо прямого запроса на авторизацию у Владелец Ресурса, Клиентское приложение перенаправляет его на доверенный Центр авторизации.

Данный сценарий является самым защищенным и наиболее предпочтителен для использования клиентским приложением

Описание сценария может быть найдено [здесь](#).

Авторизация с аутентификацией по сертификату с использованием кода авторизации по протоколу OAuth 2.0

Данный сценарий используется в случае, когда необходимо совместить возможность аутентификации по сертификату и преимущества протокола OAuth 2.0 в режиме работы с типом разрешения Код Авторизации.

Сценарий аналогичен предыдущему, но имеет следующее отличие. В процессе выполнения данного сценария устанавливается двустороннее TLS-соединение. Центр Идентификации в процессе обработки запроса пытается извлечь

клиентский сертификат и, используя его, идентифицировать и аутентифицировать пользователя.

Таким образом, данный сценарий может использоваться тогда, когда отсутствует возможность интерактивного взаимодействия с пользователем.

Описание сценария может быть найдено [здесь](#).

Авторизация с использованием типа разрешения Implicit по протоколу OAuth 2.0

Режим протокола OAuth 2.0 с неявным типом разрешения используется мобильными и веб-приложениями (приложениями, которые работают в веб-браузере), где конфиденциальность секрета клиента не может быть гарантирована. Так же, как и в случае с типом разрешения Код Авторизации, протокол использует механизм перенаправлений пользовательского агента, однако присутствуют ряд значительных отличий:

- неявный тип разрешения не поддерживает механизм маркеров обновления маркера доступа.
- отсутствует аутентификация клиентского приложения.
- адрес перенаправления не передается в запросе, а берется непосредственно из информации о клиенте, зарегистрированном ранее на ЦИ.
- операция получения маркера доступа выполняется в один шаг вместо двух.

Описание сценария может быть найдено [здесь](#).

Авторизация с использованием учетных данных владельца ресурса по протоколу OAuth 2.0

Режим работы протокола OAuth 2.0 с типом разрешения: Учетные данные Владельца ресурса подходит для использования в тех случаях, когда Владелец данных доверяет Клиентскому приложению.

Данный тип можно использовать тогда, когда Клиентское приложение имеет возможность тем или иным образом получить учетные данные Владельца данных (логин и пароль), однако данный режим является наименее защищенным и должен применяться только в случае, если остальные режимы работы недоступны.

Описание сценария может быть найдено [здесь](#).

Получение операторского делегирующего маркера доступа к ресурсу

Данный сценарий работы необходим тогда, когда на Сервисе Подписи Оператором DSS необходимо выполнить операции с сертификатами от имени пользователя.

Сценарий является расширением сценария Авторизация с аутентификацией по сертификату и включает в себя:

- аутентификацию Оператора DSS;
- получение маркера доступа для Оператора DSS;
- проверка права на делегацию полномочий Оператору DSS;
- получение делегирующего маркера доступа к ресурсам Сервиса Подписи.

Описание сценария может быть найдено [здесь](#).

Авторизация с использованием кода авторизации

Примечание

Данный сценарий требует интерактивного взаимодействия Пользователя с Веб-интерфейсом ЦИ DSS.

Спецификация описана в разделе 4.1 [RFC6749](#).

Получение доступа к функциям КриптоПро DSS посредством протокола OAuth с типом разрешения Authorization Code может быть представлено следующей схемой:



1. Запрос авторизации
2. Авторизация пользователя
3. Выдача кода авторизации
4. Запрос маркера доступа
5. Получение маркера доступа

Запрос авторизации

Для инициации запроса на авторизацию клиентское приложение должно сформировать следующий запрос и отправить его на конечную точку **/oauth/authorize**:

Пример

```
GET https://{hostname}/{stsapname}/oauth/authorize?
client_id=sample&response_type=code&scope=dss&redirect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aaob%3Aauto&resource
=urn:cryptopro:dss:signserver:signserver
accept-encoding: gzip, deflate
Connection: close
```

Параметры запроса:

- `client_id` - идентификатор клиента OAuth, зарегистрированный на ЦИ DSS. Для регистрации клиента и его последующей конфигурации можно воспользоваться командами Windows PowerShell **Add-DssClient** и **Set-DssClient** соответственно.

Примечание

При регистрации клиента параметр `AllowedFlow` должен содержать значение `AuthorizationCode`.

- `response_type` - в данном сценарии имеет значение `code`.
- `scope` - области использования маркера. Должен содержать значение `dss`.
- `redirect_uri` - зарегистрированный на Центре Идентификации адрес возврата (по этому адресу будет возвращён запрошенный код авторизации).

Примечание

Значение параметра должно соответствовать значению адреса возврата, заданного при регистрации клиента на ЦИ.

Для случаев, в которых не планируется использование выделенного HTTP-сервиса для обработки URI перенаправления, рекомендуется использовать зарезервированный URI `urn:ietf:wg:oauth:2.0:oob:auto`.

- `resource` - идентификатор ресурса, для доступа к которому выпускается токен. Для Сервиса Подписи идентификатор фиксирован и имеет вид **urn:cryptopro:dss:signserver:<signserverAppName>**.

Авторизация пользователя

В ответ на запрос ЦИ может перенаправить пользователя на страницу аутентификации. Процесс аутентификации может занять несколько шагов и состоять из множества перенаправлений на другие ресурсы.

Выдача кода авторизации

После успешной аутентификации ЦИ сформирует следующий ответ, который отправит на `redirect_uri`:

```
HTTP/1.1 302 Found
Location: urn:ietf:wg:oauth:2.0:oob:auto#code=c86322fd3a4cf85fb51249ab3fb4fcd1
Content-Length: 0
```

Параметр `code` (стоящий после символа #) содержит код авторизации, который необходимо извлечь из заголовка `Location`.

Типовые ошибки

HTTP-код	ошибка	описание
400	invalid_client	OAuth-клиент не зарегистрирован или неверно указан clientID.
400	unauthorized_client	OAuth-клиент использует незарегистрированный сценарий аутентификации (Flow) или был передан некорректный <code>redirect_uri</code> .
400	invalid_request	Неверно сформирован параметр <code>resource</code> .
500	An error has occurred	1. Проверяющая сторона с идентификатором <code>resource</code> не зарегистрирована.

Получение маркера доступа

Для получения маркера доступа используется конечная точка **/token**. Клиент формирует следующий HTTP-запрос:

```
POST https://<hostname>/<stsappname>/oauth/token HTTP/1.1
Content-Type: application/x-www-form-urlencoded
cache-control: no-cache
Accept: */*
content-length: 139

grant_type=authorization_code&code=9e554074113426cb2f4430f51b68170a&redirect_uri=http%3A%2F%2Fgrand-
pc%2FauthorizationCode&client_id=sample
```

Параметры запроса:

- `grant_type` - в данном сценарии имеет значение `authorization_code`.
- `code` - код авторизации, полученный на предыдущем этапе.
- `redirect_uri` - зарегистрированный на Центре Идентификации адрес возврата (по этому адресу будет возвращён запрошенный код авторизации).

Примечание

Значение параметра должно соответствовать значению адреса возврата, заданного при регистрации клиента на ЦИ.

- `client_id` - идентификатор клиента OAuth, зарегистрированный на ЦИ DSS.

Если в рамках сценария необходима аутентификация клиентского приложения и известен его секрет, запрос необходимо модифицировать.

Параметр `client_id` в теле запроса должен был заменен на заголовок Authorization HTTP-запроса, имеющий значение `Basic Base64(client_id:client_secret)`.

В случае успешной обработки запроса Центром Идентификации ответ будет содержать:

- `access_token` - Маркер доступа, выпущенный Центром Идентификации DSS
- `token_type` - Тип токена
- `expires_in` - Время жизни токена в секундах

Значение параметра `access_token` необходимо будет использовать при обращениях к Сервису Подписи.

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 2268
Content-Type: application/json; charset=utf-8
Expires: -1

{"access_token":"eyJ0eXAiOiJKV1Q...LnS1sAunDSE1hh3A5n8W7lhPSM4z_VA","expires_in":300,"token_type":"Bearer"}
```

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_client	OAuth-клиент не зарегистрирован или неверно указан clientID.
400	unauthorized_client	OAuth-клиент использует незарегистрированный сценарий аутентификации (Flow) или был передан некорректный <code>redirect_uri</code> .
400	invalid_request	Неверно сформирован параметр <code>resource</code> .
400	invalid_grant	Невалидный код авторизации.

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
500	An error has occurred	1. Проверяющая сторона с идентификатором resource не зарегистрирована.

Пример сообщения об ошибке

```
HTTP/1.1 400 Bad Request
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Date: Fri, 21 Dec 2018 13:46:42 GMT
Connection: close

{"error":"invalid_client"}
```


Авторизация с аутентификацией по сертификату с использованием кода авторизации

Примечание

Данный сценарий не требует интерактивного взаимодействия Пользователя с Веб-интерфейсом ЦИ DSS.

Для получения доступа к функциям КриптоПро DSS посредством протокола OAuth с типом разрешения Authorization Code необходимо выполнить следующие шаги:

1. [Формирование запроса на код авторизации](#)
2. [Получение кода авторизации](#)
3. [Получение маркера доступа](#)

Формирование запроса на код авторизации

Пример запроса

```
GET https://{hostname}/{stsappname}/oauth/authorize/certificate?
client_id=sample&response_type=code&scope=dss&redirect_uri=urn:ietf:wg:oauth:2.0:oob:auto&resource=urn:crypto
pro:dss:signserver:signserver
accept-encoding: gzip, deflate
Connection: close
```

Параметры запроса:

- `client_id` - идентификатор клиента OAuth, зарегистрированный на ЦИ DSS. Для регистрации клиента и его последующей конфигурации можно воспользоваться командами Windows PowerShell **Add-DssClient** и **Set-DssClient** соответственно.

Примечание

При регистрации клиента параметр `AllowedFlow` должен иметь значение `AuthorizationCode`.

- `response_type` - в данном сценарии имеет значение `code`.
- `scope` - области использования маркера. Должен содержать значение `dss`.
- `redirect_uri` - зарегистрированный на Центре Идентификации адрес возврата (по этому адресу будет возвращён запрошенный код авторизации).

Примечание

Значение параметра должно соответствовать значению адреса возврата, заданного при регистрации клиента на ЦИ.

Для случаев, в которых не планируется использование выделенного HTTP-сервиса для обработки URI перенаправления, рекомендуется использовать зарезервированный URI `urn:ietf:wg:oauth:2.0:oob:auto`.

- `resource` - идентификатор ресурса, для доступа к которому выпускается токен.

Также необходимо указать пользовательский сертификат в качестве клиентского SSL/TLS сертификата, именно по нему будет осуществляться поиск и аутентификация пользователя на ЦИ.

Примечание

В случае, если Центр Идентификации не сможет извлечь клиентский сертификат из запроса и корректно аутентифицировать пользователя, Пользователю будет предложена интерактивная аутентификация в Веб-Интерфейсе ЦИ.

Для запрета интерактивного взаимодействия с Веб-Интерфейсом ЦИ необходимо передать в запросе дополнительный

параметр `prompt=none`. В этом случае при невозможности аутентификации пользователя по сертификату ЦИ вернет ошибку.

Получение кода авторизации

В случае, если пользователь был успешно аутентифицирован с использованием предоставленного сертификата, ЦИ сформирует следующий ответ и отправит его на `redirect_uri`:

```
HTTP/1.1 302 Found
Location: urn:ietf:wg:oauth:2.0:oob:auto#code=c86322fd3a4cf85fb51249ab3fb4fcd1
Content-Length: 0
```

Параметр `code` (стоящий после символа #) содержит код авторизации, который необходимо извлечь из заголовка `Location`.

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_client	OAuth-клиент не зарегистрирован или неверно указан clientID
400	unauthorized_client	OAuth-клиент использует незарегистрированный сценарий аутентификации (Flow)
400	invalid_request	Неверно сформирован параметр resource
500	An error has occurred	1. Проверяющая сторона с идентификатором resource не зарегистрирована.

Получение маркера доступа

Для получения маркера доступа используется конечная точка **/token**. Клиент формирует следующий HTTP-запрос:

```
POST https://<hostname>/<stsappname>/oauth/token HTTP/1.1
cache-control: no-cache

grant_type=authorization_code&code=e1b74adb33d77c51bf0f9121b8b88662&redirect_uri=urn:ietf:wg:oauth:2.0:oob:auto&client_id=sample
```

Параметры запроса:

- `grant_type` - в данном сценарии имеет значение `authorization_code`.
- `code` - код авторизации, полученный на предыдущем этапе.
- `redirect_uri` - зарегистрированный на Центре Идентификации адрес возврата (по этому адресу будет возвращён запрошенный код авторизации).

Примечание

Значение параметра должно соответствовать значению адреса возврата, заданного при регистрации клиента на ЦИ.

- `client_id` - идентификатор клиента OAuth, зарегистрированный на ЦИ DSS.

Если в рамках сценария необходима аутентификация клиентского приложения и известен его секрет, запрос необходимо модифицировать.

Параметр `client_id` в теле запроса должен был заменен на заголовок `Auhtorization` HTTP-запроса, имеющий значение `Basic Base64(client_id:client_secret)`.

В случае успешной обработки запроса Центром Идентификации ответ будет содержать:

- `access_token` - Маркер доступа, выпущенный Центром Идентификации DSS
- `token_type` - Тип токена
- `expires_in` - Время жизни токена в секундах

Значение параметра `access_token` необходимо будет использовать при обращениях к Сервису Подписи.

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 2268
Content-Type: application/json; charset=utf-8
Expires: -1

{"access_token":"eyJ0eXAiOiJKV1Q...LnS1sAunDSE1hh3A5n8W7lhPSM4z_VA","expires_in":300,"token_type":"Bearer"}
```

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_client	OAuth-клиент не зарегистрирован или неверно указан clientID
400	unauthorized_client	OAuth-клиент использует незарегистрированный сценарий аутентификации (Flow)
400	invalid_request	Неверно сформирован параметр resource
400	invalid_grant	Невалидный код авторизации.
500	An error has occurred	1. Проверяющая сторона с идентификатором resource не зарегистрирована.

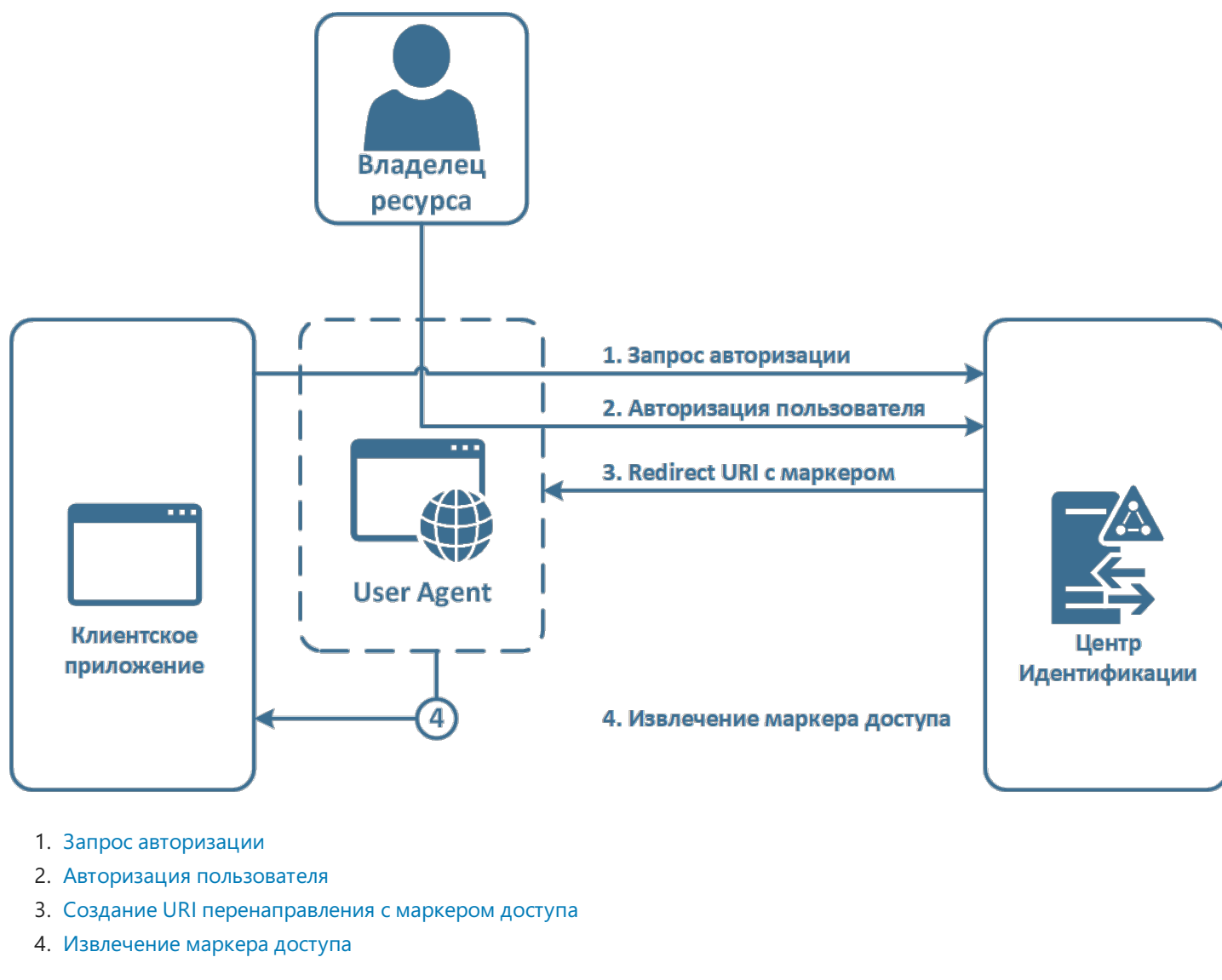
Авторизация с неявным типом разрешения

Примечание

Данный сценарий требует интерактивного взаимодействия Пользователя с Веб-интерфейсом ЦИ DSS.

Спецификация описана в разделе 4.2 [RFC6749](#).

Получение маркера доступа в сценарии с неявным типом разрешения может быть представлено следующей схемой:



Запрос авторизации

Для инициации запроса на авторизацию клиентское приложение должно сформировать следующий запрос и отправить его на конечную точку **/oauth/authorize**:

Пример

```
https://<hostname>/<stsappname>/oauth/authorize?  
client_id=implicitsample&response_type=token&scope=dss&redirect_uri=urn:ietf:wg:oauth:2.0:oob:auto&resource=ht  
tps://grand-pc/signserver/rest/api HTTP/1.1
```

Параметры запроса:

- **client_id** - идентификатор клиента OAuth, зарегистрированный на ЦИ DSS. Для регистрации клиента и его последующей конфигурации можно воспользоваться командами Windows PowerShell **Add-DssClient** и **Set-DssClient** соответственно.

Примечание

При регистрации клиента параметр `AllowedFlow` должен содержать значение `Implicit`.

- `response_type` - в данном сценарии имеет значение `token`.
- `scope` - области использования маркера. Должен содержать значение `dss`.
- `redirect_uri` - зарегистрированный на Центре Идентификации адрес возврата (по этому адресу будет возвращён запрошенный код авторизации).

Примечание

Значение параметра должно соответствовать значению адреса возврата, заданного при регистрации клиента на ЦИ.

Для случаев, в которых не планируется использование выделенного HTTP-сервиса для обработки URI перенаправления, рекомендуется использовать зарезервированный URI `urn:ietf:wg:oauth:2.0:oob:auto`.

- `resource` - идентификатор ресурса, для доступа к которому выпускается токен. В случае Сервиса Подписи идентификатор фиксирован и имеет вид `urn:cryptopro:dss:signserver:<signserverAppName>`.

Если в рамках сценария необходима аутентификация клиентского приложения и известен его секрет, запрос необходимо модифицировать.

Параметр `client_id` в теле запроса должен был заменен на заголовок `Auhtorization` HTTP-запроса, имеющий значение `Basic Base64(client_id:client_secret)`.

Авторизация пользователя

В ответ на запрос ЦИ может перенаправить пользователя на страницу аутентификации. Процесс аутентификации может занять несколько шагов и состоять из множества перенаправлений на другие ресурсы.

Создание URI перенаправления с маркером доступа

В случае успешной аутентификации Центр Идентификации готовит URI перенаправления на основании данных, переданных в запросе. Адрес включает в себя маркер доступа, который должен быть извлечен из ответа сервера

Пример ответа от сервера

```
HTTP/1.1 302 Found
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Location:
urn:ietf:wg:oauth:2.0:oob:auto#access_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUz...&token_type=Bearer&expires_in=300
Date: Fri, 14 Dec 2018 15:15:26 GMT
Content-Length: 0
```

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_client	OAuth-клиент не зарегистрирован или неверно указан clientID
400	unauthorized_client	OAuth-клиент использует незарегистрированный сценарий аутентификации (Flow)
400	invalid_request	Неверно сформирован параметр resource
500	An error has occurred	1. Проверяющая сторона с идентификатором resource не зарегистрирована.

Извлечение маркера доступа

Для извлечения маркера доступа необходимо получить значение заголовка `Location`, содержащегося в HTTP-ответе от Центра Идентификации. Заголовок `Location` будет представлять из себя URI перенаправления, включающий в себя маркер доступа в качестве параметра.

В случае, если в качестве `redirect_uri` был передан валидный HTTP-адрес, маркер доступа (в Base64-формате) может быть извлечен из query-string запроса, содержащегося в `Location_`

В случае служебного URI `urn:ietf:wg:oauth:2.0:oob:auto` данные, стоящие после знака `#` будут представлять из себя маркер доступа к ресурсу, закодированный в Base64 формате.

Авторизация с использованием учетных данных владельца

Примечание

Данный сценарий не требует интерактивного взаимодействия Пользователя с Веб-интерфейсом ЦИ DSS.

Спецификация описана в разделе 4.2 [RFC6749](#).

Сценарий может быть представлен следующей схемой:



Для получения маркера доступа используется конечная точка **/token**. Клиент формирует следующий HTTP-запрос:

Пример запроса

```
POST http://<hostname>/<stsappname>/oauth/token HTTP/1.1
```

```
Content-Type: application/x-www-form-urlencoded
```

```
content-length: 133
```

```
Connection: keep-alive
```

```
grant_type=password&username=Test1&client_id=TestClient&resource=urn:cryptopro:dss:signserver:signserver&password=Test1Test1
```

Примечание

Параметры передаются в теле запроса, но не в query-string.

Параметры запроса:

- `grant_type` - в данном сценарии имеет значение `password`.
- `username` - логин Владельца ресурса.
- `password` - пароль Владельца ресурса.
- `resource` - идентификатор ресурса, для доступа к которому выпускается токен. Для Сервиса Подписи идентификатор фиксирован и имеет вид `urn:cryptopro:dss:signserver:<signserverAppName>`.

- `client_id` - идентификатор клиента OAuth, зарегистрированный на ЦИ DSS.

Примечание

Для регистрации клиента и его последующей конфигурации можно воспользоваться командами Windows PowerShell **Add-DssClient** и **Set-DssClient** соответственно.

Примечание

При регистрации клиента параметр `AllowedFlow` должен иметь значение `ResourceOwner`.

Если в рамках сценария необходима аутентификация клиентского приложения и известен его секрет, запрос необходимо модифицировать.

Параметр `client_id` в теле запроса должен был заменен на заголовок Authorization HTTP-запроса, имеющий значение `Basic Base64(client_id:client_secret)`.

В случае успешного выполнения данного запроса ЦИ вернет следующий ответ:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 2196
Content-Type: application/json; charset=utf-8
Expires: -1
Date: Fri, 14 Dec 2018 12:35:53 GMT
Connection: close

{"access_token":"eyJ0eXAiOiJKV1Qi...","expires_in":300,"token_type":"Bearer"}
```

Типовые ошибки

HTTP-код	ошибка	описание
400	invalid_client	OAuth-клиент не зарегистрирован или неверно указан clientID
400	unauthorized_client	OAuth-клиент использует незарегистрированный сценарий аутентификации (Flow) или был передан некорректный <code>redirect_uri</code> .
400	invalid_request	Неверно сформирован параметр resource
500	An error has occurred	1. Проверяющая сторона с идентификатором resource не зарегистрирована.

Пример сообщения об ошибке

```
HTTP/1.1 400 Bad Request
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Date: Fri, 21 Dec 2018 13:46:42 GMT
Connection: close

{"error":"invalid_client"}
```


Авторизация с использованием двухфакторной аутентификации

В силу того, что двухфакторная аутентификация на ЦИ DSS требует интерактивного взаимодействия с пользователем, сценарий протокола OAuth с утверждениями типа Resourceowner не может быть выполнен для пользователей, которым назначен второй фактор аутентификации.

В случае, если таких пользователей необходимо авторизовать, следует использовать описанный ниже сценарий.

Шаг 1. Формирование запроса к ЦИ на инициализацию процедуры аутентификации

Для инициирования процедуры двухфакторной аутентификации необходимо сформировать запрос к ЦИ, включающий в себя идентификатор ресурса и базовые аутентификационные данные пользователя (логин/пароль).

Пример запроса

```
POST https://host/STS/confirmation HTTP/1.1
Content-Type: application/json
Authorization: Basic VGVzdDE6VGZdDFUZXN0MQ==

{
  "Resource": "urn:cryptopro:dss:signserver:signserver",
  "ClientId": "oauth-client-id",
  "ClientSecret": "oauth-client-secret"
}
```

Примечание

Логин и пароль пользователя передаются в заголовке `Authorization` в виде `Basic BASE64(Login:Password)`, где `BASE64` - функция кодирования строки в формат BASE64, `Login` и `Password` - логин и пароль пользователя соответственно.

E.g: Логин: Test1, Пароль: Test1Test1, значение заголовка - `Basic VGVzdDE6VGZdDFUZXN0MQ==`.

В случае, если пароль у пользователя отсутствует, заголовок должен иметь значение `Basic BASE64(Login:)`.

- `ClientId` - идентификатор OAuth клиента.
- `ClientSecret` - пароль OAuth клиента (для неконфиденциальных клиентов данный параметр не указывается).

В зависимости от количества подключенных способов подтверждения операций возможно два варианта ответа на запрос:

- Если у Пользователя подключен один метод подтверждения операций, то при успешном выполнении запроса он получит одноразовый пароль на указанный номер мобильного телефона (*Пример ответа 1*).
- Если у Пользователя подключено несколько методов подтверждения операций, то при успешном выполнении запроса он получит доступные методы подтверждения операции (*Пример ответа 2*). В этом случае необходимо выбрать [метод подтверждения операции](#).

В ответ на этот запрос ЦИ пришлет приглашение к вводу информации по второму фактору аутентификации (формат зависит от выбранного метода аутентификации)

Далее, на примере вторичной аутентификации по СМС будут приведены примеры запросов и ответов от ЦИ.

Пример ответа 1

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
  "Challenge": {
    "Title": {
      "Value": "На ваш номер отправлено SMS сообщение с одноразовым паролем"
    },
    "TextChallenge": [
      {
        "AuthnMethod": "http://dss.cryptopro.ru/identity/authenticationmethod/otpviasms",
        "RefID": "82d2cd39-81e1-4a59-904c-93047006cd95",
        "Label": "Подтвердите операцию входа пользователя. Идентификатор запроса ngwsouyu",
        "MaxLenSpecified": false,
        "HideTextSpecified": false,
        "ExpiresIn": 300,
        "ExpiresInSpecified": true
      }
    ],
    "ContextData": {
      "RefID": "82d2cd39-81e1-4a59-904c-93047006cd95"
    }
  },
  "IsFinal": false,
  "IsError": false
}
```

Пример ответа 2

Необходимо запомнить идентификатор транзакции *RefID* для выбора метода ее подтверждения в пункте 2.1.1.

```

HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 6736
Content-Type: application/json; charset=utf-8
Expires: -1

{
  "Challenge": {
    "Title": {
      "Value": "Для подтверждения операции необходимо выбрать способ аутентификации"
    },
    "ChoiceChallenge": [
      {
        "Choice": [
          {
            "RefID": "http://dss.cryptopro.ru/identity/authenticationmethod/otpviasms",
            "Label": "Аутентификация с помощью СМС"
          },
          {
            "RefID": "http://dss.cryptopro.ru/identity/authenticationmethod/mobile",
            "Label": "Аутентификация с помощью мобильного приложения"
          }
        ],
        "RefID": "7406c54e-0ba5-4340-bbaa-3c2feebdc852",
        "Label": "Для подтверждения операции необходимо выбрать способ аутентификации",
        "ExactlyOne": true,
        "ExactlyOneSpecified": true,
        "ExpiresIn": 86400,
        "ExpiresInSpecified": true
      }
    ],
    "ContextData": {
      "RefID": "7406c54e-0ba5-4340-bbaa-3c2feebdc852"
    }
  },
  "IsFinal": false,
  "IsError": false
}

```

POST-запрос на выбор метода подтверждения транзакции (только если подключено несколько методов подтверждения)

В заголовке Authorization HTTP-запроса клиент должен указать `AccessToken`, полученный при аутентификации:

```
Authorization: Bearer \<access_token>
```

В запросе необходимо указать:

- `ChoiceSelected.RefID` – Идентификатор выбранного метода подтверждения транзакции
- `RefId` – Идентификатор транзакции, созданной на Сервисе Подтверждения Операций (пункт 2.1).
- `Resource` – Идентификатор Сервиса Подписи в формате `urn:cryptopro:dss:signserver:<signserver app name>` (по умолчанию имеет значение **`urn:cryptopro:dss:signserver:signserver`**).

```

POST https://host/STS/confirmation HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJHM ... 5aPB98A3NAVduJbtz5Wti-H8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 246
Expect: 100-continue
{
  "Resource" : "urn:cryptopro:dss:signserver:signserver",
  "ClientId": "oauth-client-id",
  "ClientSecret": "oauth-client-secret",
  "ChallengeResponse" : {
    "ChoiceChallengeResponse" : [ {
      "RefId" : "49744464-5cd7-419d-891e-2495b8f49539",
      "ChoiceSelected": [{
        "RefID": "http://dss.cryptopro.ru/identity/authenticationmethod/otpviasms",
      } ]
    } ]
  }
}

```

Пример ответа

При успешном выполнении запроса Пользователь должен получить на свой номер мобильного телефона одноразовый пароль. Необходимо запомнить *новый* идентификатор транзакции **RefID** для ее подтверждения в пункте 2.2.

```

HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 6736
Content-Type: application/json; charset=utf-8
Expires: -1

{
  "Challenge": {
    "Title": {
      "Value": "Код подтверждения отправлен на ваш номер мобильного телефона"
    },
    "TextChallenge": [
      {
        "AuthnMethod": "http://dss.cryptopro.ru/identity/authenticationmethod/otpviasms",
        "RefID": "5460867e-be6a-4940-bebc-9aeb516fa448",
        "Label": "02.11.2018 9:47:05. Подпись документа. testPdf.pdf. Идентификатор операции gcoaryoy. Пользователь Test2. Сертификат: test2.",
        "MaxLenSpecified": false,
        "HideTextSpecified": false,
        "ExpiresIn": 86400,
        "ExpiresInSpecified": true
      }
    ],
    "ContextData": {
      "RefID": "5460867e-be6a-4940-bebc-9aeb516fa448"
    }
  },
  "IsFinal": false,
  "IsError": false
}

```

Примечание

RefId - Идентификатор транзакции, созданной на Сервисе Подтверждения Операций. Его необходимо будет использовать при следующем обращении на конечную точку /confirmation (пункт 2.2).

Шаг 2. Формирование ChallengeResponse и получение маркера безопасности

Следующим этапом является формирование объекта Challenge Response, включающего в себя данные для вторичной аутентификации (например, QR-code, одноразовый пароль и т.п.).

Формат данных зависит от выбранного метода вторичной аутентификации и информацию о нем можно получить в соответствующих разделах документации.

Пример запроса

```
POST https://host/STS/confirmation HTTP/1.1
Authorization: Basic VGVzdDE6VGZdDFUZXN0MQ==
Content-Type: application/json
{
  "Resource" : "urn:cryptopro:dss:signserver:signserver",
  "ClientId": "oauth-client-id",
  "ClientSecret": "oauth-client-secret",
  "ChallengeResponse":
  {
    "TextChallengeResponse":
      [
        {
          "RefId": "ccb39b68-acd4-437f-9eef-563738987011",
          "Value": "62985"
        }
      ]
  }
}
```

Если были переданы корректные аутентификационные данные, ЦИ в ответе вернет маркер безопасности:

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
  "AccessToken": "eyJ0eXAiOiJKV1QiLCJh...",
  "ExpiresIn": 600,
  "IsFinal": true,
  "IsError": false
}
```

Сервисный OAuth-клиент

В случае если интегрируемая система не предусматривает интерактивного взаимодействия с пользователем для получения учётных данных или не хранит учётные данные, то взаимодействие с DSS может быть реализовано по схеме Сервис-Сервис.

В схеме Сервис-Сервис не требуется аутентификация пользователя, необходима лишь аутентификация сторонней системы. Аутентификация сторонней системы может быть осуществлена двумя способами:

- по паролю
- по сертификату

При использовании данной схемы сторонняя система может вызывать API DSS от имени пользователя.

Примечание

Сервисный OAuth-клиент не позволяет выполнить действия, требующие двухфакторной аутентификации, без подтверждения пользователя. Исключение составляет операция подтверждения входа.

Ограничения

При работе по схеме Сервис-Сервис на сценарий аутентификации накладывается ряд ограничений. Для того, чтобы сторонняя система смогла корректно работать в рамках данной схемы, необходимо следующее:

- Запрос на авторизацию должен проходить по сценарию `Пароль (password)` или `Обмен токена (Token Exchange)`
- Клиент должен быть конфиденциальным, то есть обладать назначенным ему секретом

Примечание

При выполнении сценариев аутентификации, отличных от описанных выше, данный клиент работает в соответствии со спецификацией.

Регистрация

Для регистрации клиента в режиме сервисного необходимо воспользоваться командлетом Powershell `Add-DssClient` с параметром `SuppressAuthForIssue`, а также параметром `GenerateSecret` для регистрации конфиденциального клиента.

Пример команды:

```
Add-DssClient -Identifier test_id -Name testId -AllowedFlow ResourceOwner -GenerateSecret -  
SuppressAuthForIssue 1
```

Для изменения режима работы существующего клиента необходимо сделать следующее:

- Выполнить назначение секрета модифицируемому клиенту при помощи командлета `Add-DssClientSecret`.

Пример команды:

```
Add-DssClientSecret -ClientId test_id -Type SharedSecret -Value 12345
```

Примечание

В случае, если клиент уже является конфиденциальным, данный шаг можно пропустить.

- Задать параметр `SuppressAuthForIssue` OAuth-клиента в значение `true` посредством командлета PowerShell `Set-DssClientSecret`.

Пример команды:

```
Set-DssClient -ClientId test_id -SuppressAuthForIssue 1
```

Примеры

В данном примере использован метод аутентификации сторонней системы по паролю.

Пример запроса на аутентификацию:

```
POST /STS/oauth/token HTTP/1.1
Host: grand-pc
Content-Type: application/x-www-form-urlencoded
Authorization: Basic dGVzdF9pZDoxMjM0NQ==

grant_type=password&username=Test1&resource=urn%3Acryptopro%3Adss%3Asignserver%3ASignServer&password=
```

Примечание

Не смотря на то, что пароль пользователя не проверяется, параметр `password` должен обязательно присутствовать в запросе

Пример ответа:

```
{
  "access_token": "eyJ0eXAiOiJK...",
  "expires_in": 300,
  "token_type": "Bearer"
}
```

Получение операторского маркера доступа для управления пользователями

Примечание

Данный сценарий не требует интерактивного взаимодействия Пользователя с Веб-интерфейсом ЦИ DSS.

Данный раздел включает в себя описание операций, которые необходимо выполнить клиентскому приложению для получения маркера доступа Оператора, позволяющего выполнять на Сервисе Подписи операции с сертификатами и запросами от имени пользователя.

Перед началом интеграции Администратору DSS необходимо:

- Выпустить и зарегистрировать на DSS сертификат аутентификации Оператора DSS
- Зарегистрировать OAuth-клиента

Для получения маркера доступа необходимо выполнить следующие шаги:

1. [Инициация аутентификации.](#)
2. [Получение кода авторизации.](#)
3. [Получение маркера доступа Оператора.](#)
4. [Получение делегирующего маркера доступа.](#)

Инициация аутентификации

Данный шаг заключается в отправке GET-запроса на аутентификацию на конечную точку **/authorize/certificate**. Примера запроса представлен ниже.

Пример запроса

```
https://<hostname>/<stsappname>/oauth/authorize/certificate?  
client_id=sample&response_type=code&scope=dss&redirect_uri=urn:ietf:wg:oauth:2.0:oob:auto&resource=urn:crypto  
pro:dss:signserver:signserver
```

Параметры запроса:

- `client_id` - идентификатор клиента OAuth, зарегистрированный на ЦИ DSS. Для регистрации клиента и его последующей конфигурации можно воспользоваться командами Windows PowerShell **Add-DssClient** и **Set-DssClient** соответственно.

Примечание

При регистрации клиента параметр `AllowedFlow` должен включать в себя разрешения `AuthorizationCode` и `ResourceOwner`.

- `response_type` - в данном сценарии имеет значение `code`.
- `scope` - области использования маркера. Должен содержать значение `dss`.
- `redirect_uri` - зарегистрированный на Центре Идентификации адрес возврата (по этому адресу будет возвращён запрошенный код авторизации).

Примечание

Значение параметра должно соответствовать значению адреса возврата, заданного при регистрации клиента на ЦИ.

Для случаев, в которых не планируется использование выделенного HTTP-сервиса для обработки URI перенаправления, рекомендуется использовать зарезервированный URI `urn:ietf:wg:oauth:2.0:oob:auto`.

- `resource` - идентификатор ресурса, для доступа к которому выпускается токен.

Если в рамках сценария необходима аутентификация клиентского приложения и известен его секрет, запрос необходимо модифицировать.

Параметр `client_id` в теле запроса должен был заменен на заголовок `Auhtorization` HTTP-запроса, имеющий значение `Basic Base64(client_id:client_secret)`.

Также необходимо указать операторский сертификат в качестве клиентского SSL/TLS сертификата, по нему будет осуществляться аутентификация оператора на ЦИ.

Получение кода авторизации

В случае успешной аутентификации

- ответ сервера будет иметь статус HTTP 302
- В заголовке `Location` будет содержаться код авторизации

В примере используется специальное значение `redirect_uri`, клиенту необходимо из заголовка `Location` извлечь значение параметра `code`. Значение параметра `code` будет использовано для получения `AccessToken` на следующем шаге.

Пример ответа

```
HTTP/1.1 302 Found
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Location: urn:ietf:wg:oauth:2.0:oob:auto?code=65e4322a9751cf9ba43012692ce02ec1
Date: Fri, 07 Sep 2018 10:30:24 GMT
Content-Length: 0
```

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_client	OAuth-клиент не зарегистрирован или неверно указан clientID
400	unauthorized_client	OAuth-клиент использует незарегистрированный сценарий аутентификации (Flow)
400	invalid_request	Неверно сформирован параметр resource
500	An error has occurred	1. Проверяющая сторона с идентификатором resource не зарегистрирована.

Получение операторского маркера доступа

Для получения маркера доступа используется конечная точка **/token**. Клиент формирует следующий HTTP-запрос:

```
POST https://<hostname>/<stsappname>/oauth/token HTTP/1.1
cache-control: no-cache

grant_type=authorization_code&code=e1b74adb33d77c51bf0f9121b8b88662&redirect_uri=urn:ietf:wg:oauth:2.0:oob:auto&client_id=sample
```

Параметры запроса:

- `grant_type` - в данном сценарии имеет значение `authorization_code`.
- `code` - код авторизации, полученный на предыдущем этапе.
- `redirect_uri` - зарегистрированный на Центре Идентификации адрес возврата (по этому адресу будет возвращён запрошенный код авторизации).

Значение параметра должно соответствовать значению адреса возврата, заданного при регистрации клиента на ЦИ.

- `client_id` - идентификатор клиента OAuth, зарегистрированный на ЦИ DSS.

В случае успешной обработки запроса Центром Идентификации ответ будет содержать:

- `access_token` - Маркер доступа, выпущенный Центром Идентификации DSS
- `token_type` - Тип токена
- `expires_in` - Время жизни токена в секундах

Значение параметра `access_token` необходимо будет использовать при обращениях к Сервису Подписи.

Примечание

Данный `access_token` не даёт право Оператору DSS выполнять операции на Сервисе Подписи от имени пользователей.
`access_token` может быть использован для получения [Политики Сервиса Подписи](#).

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 2268
Content-Type: application/json; charset=utf-8
Expires: -1
```

```
{"access_token":"eyJ0eXAiOiJKV1Q...LnS1sAunDSE1hh3A5n8W7lhPSM4z_VA","expires_in":300,"token_type":"Bearer"}
```

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_client	OAuth-клиент не зарегистрирован или неверно указан clientID
400	unauthorized_client	OAuth-клиент использует незарегистрированный сценарий аутентификации (Flow)
400	invalid_request	Неверно сформирован параметр resource
400	invalid_grant	Невалидный код авторизации.
500	An error has occurred	1. Проверяющая сторона с идентификатором resource не зарегистрирована.

Получение делегирующего маркера доступа

Для получения AccessToken для делегирования используется конечная точка `oauth/token`. Подробная информация по протоколу получения AccessToken для делегирования: [OAuth 2.0 Token Exchange](#).

Параметры запроса:

- `grant_type` - тип разрешения, в данном сценарии равен **urn:ietf:params:oauth:grant-type:token-exchange**.
- `resource` – идентификатор Сервиса Подписи.
- `actor_token` - Маркер доступа, полученный на предыдущем шаге
- `actor_token_type` – тип маркера доступа, должен иметь значение **urn:ietf:params:oauth:token-type:jwt**.
- `subject_token_type` – тип маркера доступа, должен иметь значение **urn:ietf:params:oauth:token-type:jwt**.
- `subject_token` – неподписанный JWT-токен, содержащий логин управляемого пользователя.

В декодированном виде `subject_token` имеет вид:

```
{
  "alg": "none",
  "typ": "JWT"
}.
{
  "unique_name": "mydss",
  "nbf": 1488312889,
  "exp": 1488316489,
  "iat": 1488312889
}
.
```

Пример кодирования JWT-токена можно посмотреть по [ссылке](#).

Первая часть (до точки) называется header, вторая – payload. Для получения закодированного значения необходимо выполнить следующее преобразование:

```
Base64UrlEncode(Utf8GetBytes(header)) + "." + Base64UrlEncode(Utf8GetBytes(payload)) + "."
```

Примечание

Поле `unique_name` должно содержать логин пользователя, для которого осуществляется делегация прав.

Поля `nbf`, `exp` и `iat` представляют из себя даты в формате Unix Epoch и задают дату начала действия токена, дату истечения срока действия и дату подписания токена соответственно.

Внимание!

Символ "." в конце получившегося значения является обязательным.

В заголовке Authorization HTTP-запроса клиент должен передать идентификатор OAuth-клиента и секрет (если используется): Authorization: Basic Base64(<client_id>:<secret>)

Пример запроса

```
POST https://<hostname>/<stsappname>/oauth/token HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Authorization: Basic c2FtcGx1Og==

accept-encoding: gzip, deflate
content-length: 2908

grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Atoken-
exchange&resource=urn%3Acryptopro%3Adss%3Asignserver%3Asignserver&actor_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Im5mNzJmKE4VnJqWUzJQkVSLW9ycm9GNWpTcyJ9.eyJ1bm1ldWVfbmFtZSI6ImFkbWluIiwibmFtZWI6IjoiNGYyZjc2OTktODBkMS00YmQwLWlIXZWEtNjQ4MWE2NjIyZjJhIiwiaXNzX2l2cyI6InJlYWxzZHMlLCJkc3NfdXVpZCI6IiN1WGNDs3lGUjFBVUVPWwtbnWhycw1qZ3BFND0iLCJodHRwOi8vZHNzLmNyeXB0b3Byby5ydS9pZGVudG10eS9jbGFpbXMvZGlzcGxheW5hbWUiOiJhZG1pbiIsImh0dHA6Ly9zY2h1bWVzLnhtbHNvYXAub3JnL3dzLzIwMDUvMDUvawRlbnRpdHkvY2xhaw1zL3g1MDBkaXN0aw5ndWlzaGVkbmFtZSI6IknOPURTU0FkbWluIiwicm9sZSI6IkFkbWlucyIsImh0dHA6Ly9kc3MuY3J5cHRvcHJvLnJlL2lkZW50aXR5L2NsYWltcy9hY3Rpb24vaXNzdWUiOiJ0ZmE6ZmFsc2UiLCJodHRwOi8vZHNzLmNyeXB0b3Byby5ydS9pZGVudG10eS9jbGFpbXMvYWN0aw9uL3NpZ25kb2N1bWVudCI6InRmYTpmYWxzZSI6Imh0dHA6Ly9kc3MuY3J5cHRvcHJvLnJlL2lkZW50aXR5L2NsYWltcy9hY3Rpb24vc2lnbmRvY3VtZW50cyI6InRmYTpmYWxzZSI6Imh0dHA6Ly9kc3MuY3J5cHRvcHJvLnJlL2lkZW50aXR5L2NsYWltcy9hY3Rpb24vZGVjcmlwdGRvY3VtZW50IjojdGZhOmZhbHNIiwiawHR0cDovL2Rzcy5jcmlwdG9wcm8ucnUvaWRlbnRpdHkvY2xhaw1zL2FjdGlvbi9jcmlwdGVyZXZlZXN0IjojdGZhOmZhbHNIiwiawHR0cDovL2Rzcy5jcmlwdG9wcm8ucnUvaWRlbnRpdHkvY2xhaw1zL2FjdGlvbi9jaGFuZ2Vaw4iOiJ0ZmE6ZmFsc2UiLCJodHRwOi8vZHNzLmNyeXB0b3Byby5ydS9pZGVudG10eS9jbGFpbXMvYWN0aw9uL3JlbnM9rZWNlcnRpmjYXRlIjojdGZhOmZhbHNIiwiawHR0cDovL2Rzcy5jcmlwdG9wcm8ucnUvaWRlbnRpdHkvY2xhaw1zL2FjdGlvbi9ob2xkY2VydGlmawNhdGUiOiJ0ZmE6ZmFsc2UiLCJodHRwOi8vZHNzLmNyeXB0b3Byby5ydS9pZGVudG10eS9jbGFpbXMvYWN0aw9uL3VuaG9sZGnlcnRpmjYXRlIjojdGZhOmZhbHNIiwiawHR0cDovL2Rzcy5jcmlwdG9wcm8ucnUvaWRlbnRpdHkvY2xhaw1zL2FjdGlvbi9kZWxldGVjZXJ0awZpY2F0ZSI6InRmYTpmYWxzZSI6Imh0dHA6Ly9kc3MuY3J5cHRvcHJvLnJlL2lkZW50aXR5L2NsYWltcy9hY3Rpb24vcHJpdmF0ZWtleWFjY2VzcyI6InRmYTpmYWxzZSI6ImRzc19ncm91cCI6IkRlZmF1bHQiLCJodHRwOi8vZHNzLmNyeXB0b3Byby5ydS9pZGVudG10eS9jbGFpbXMvYWNjZXNzcG9saWN5IjoimCIsImNlcnR0aHVtYnByaw50IjoimZUwMUQ0MzUzRjg2N0JGMzU3NDZERDU4MkFCNTJBRDk1RDlIMTVBMyIsImh0dHA6Ly9kc3MuY3J5cHRvcHJvLnJlL2lkZW50aXR5L2NsYWltcy9nb3N0LXRodW1icHJpbnQiOiJ1SF1Pa055eGx1L3YwOUhPL2NTeVpXcmdScnlMcEx2QUoyMwpyY3N2NG9nPSIsIm1zcyI6InJlYWxzZHMlLCJleHAiOiJlNDUyMjAyODgsIm5iZiI6MTU0NTIxOTk4OH0.d39MMW4P0iItKwStWHQg_y1wInPaDy4TzLYfaLeZDgIqv3uEG4MjC11sA8va5aJxiIwxBkdkXYKwmHs1em0k3J00hsEJxFKkaYe8T9iBV93DlCAyoI_r60y9C006-7I936uCQ0vBNEdk0-qg-
iT4STK_qpHtPhTIlloN0kjGjNDKyvRhH8wUDjz_08j3MfQ4B31QGuz81K70rM_G4l3DmckTvi7TG6n_PVm1VUqxYyyQpID7NmeJ6RoLjv-axKrk1Zl3J3q6aIbcuWlRrnsq6VP0j1ok4EXOVtR1ak57GUZHus9WxiZmvOMKqeTyHmsLpF-fIznS0lCdj0byGFA&actor_token_type=urn%3Aietf%3Aparams%3Aoauth%3Atoken-type%3Ajwt&subject_token_type=urn%3Aietf%3Aparams%3Aoauth%3Atoken-type%3Ajwt&subject_token=ewogICAgImFsZyI6ICJub25lIiwKICAgICJ0eXAiOiAiSldUIGp9.ewogICAgInVuaXF1ZV9uYW1lIjojIlRlc3QxIiwKICAgICJ0eXAiOiAxNTQ1MDU2ODc2LAogICAgImV4cCI6IDE1NzY3NjU2NzYsCiAgICAiaWF0IjojMTU0NTIyOTY3Ngp9.
```

В случае успешной обработки запроса Центром Идентификации ответ будет содержать:

- `access_token` - делегирующий AccessToken, выпущенный Центром Идентификации DSS
- `token_type` - Тип токена
- `expires_in` - Время жизни токена в секундах

Значение параметра `access_token` необходимо будет использовать при обращениях к Сервису Подписи во время выполнения операции оператором от имени пользователя.

Пример ответа

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 2704
Content-Type: application/json; charset=utf-8
Expires: -1
Date: Wed, 19 Dec 2018 11:46:31 GMT
Connection: close

{"access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Im5mNzJmKE4VnJqWUzJQkVSLW9ycm9GNWpTcyJ9.eyJ1bm1ldWVfbmFtZSI6ImFkbWluIiwibmFtZWI6IjoiNGYyZjc2OTktODBkMS00YmQwLWlIXZWEtNjQ4MWE2NjIyZjJhIiwiaXNzX2l2cyI6InJlYWxzZHMlLCJkc3NfdXVpZCI6IiN1WGNDs3lGUjFBVUVPWwtbnWhycw1qZ3BFND0iLCJodHRwOi8vZHNzLmNyeXB0b3Byby5ydS9pZGVudG10eS9jbGFpbXMvZGlzcGxheW5hbWUiOiJhZG1pbiIsImh0dHA6Ly9zY2h1bWVzLnhtbHNvYXAub3JnL3dzLzIwMDUvMDUvawRlbnRpdHkvY2xhaw1zL3g1MDBkaXN0aw5ndWlzaGVkbmFtZSI6IknOPURTU0FkbWluIiwicm9sZSI6IkFkbWlucyIsImh0dHA6Ly9kc3MuY3J5cHRvcHJvLnJlL2lkZW50aXR5L2NsYWltcy9hY3Rpb24vaXNzdWUiOiJ0ZmE6ZmFsc2UiLCJodHRwOi8vZHNzLmNyeXB0b3Byby5ydS9pZGVudG10eS9jbGFpbXMvYWN0aw9uL3NpZ25kb2N1bWVudCI6InRmYTpmYWxzZSI6Imh0dHA6Ly9kc3MuY3J5cHRvcHJvLnJlL2lkZW50aXR5L2NsYWltcy9hY3Rpb24vZGVjcmlwdGRvY3VtZW50cyI6InRmYTpmYWxzZSI6Imh0dHA6Ly9kc3MuY3J5cHRvcHJvLnJlL2lkZW50aXR5L2NsYWltcy9hY3Rpb24vZGVjcmlwdGRvY3VtZW50IjojdGZhOmZhbHNIiwiawHR0cDovL2Rzcy5jcmlwdG9wcm8ucnUvaWRlbnRpdHkvY2xhaw1zL2FjdGlvbi9jcmlwdGVyZXZlZXN0IjojdGZhOmZhbHNIiwiawHR0cDovL2Rzcy5jcmlwdG9wcm8ucnUvaWRlbnRpdHkvY2xhaw1zL2FjdGlvbi9jaGFuZ2Vaw4iOiJ0ZmE6ZmFsc2UiLCJodHRwOi8vZHNzLmNyeXB0b3Byby5ydS9pZGVudG10eS9jbGFpbXMvYWNjZXNzcG9saWN5IjoimCIsImNlcnR0aHVtYnByaw50IjoimZUwMUQ0MzUzRjg2N0JGMzU3NDZERDU4MkFCNTJBRDk1RDlIMTVBMyIsImh0dHA6Ly9kc3MuY3J5cHRvcHJvLnJlL2lkZW50aXR5L2NsYWltcy9nb3N0LXRodW1icHJpbnQiOiJ1SF1Pa055eGx1L3YwOUhPL2NTeVpXcmdScnlMcEx2QUoyMwpyY3N2NG9nPSIsIm1zcyI6InJlYWxzZHMlLCJleHAiOiJlNDUyMjAyODgsIm5iZiI6MTU0NTIxOTk4OH0.d39MMW4P0iItKwStWHQg_y1wInPaDy4TzLYfaLeZDgIqv3uEG4MjC11sA8va5aJxiIwxBkdkXYKwmHs1em0k3J00hsEJxFKkaYe8T9iBV93DlCAyoI_r60y9C006-7I936uCQ0vBNEdk0-qg-
iT4STK_qpHtPhTIlloN0kjGjNDKyvRhH8wUDjz_08j3MfQ4B31QGuz81K70rM_G4l3DmckTvi7TG6n_PVm1VUqxYyyQpID7NmeJ6RoLjv-axKrk1Zl3J3q6aIbcuWlRrnsq6VP0j1ok4EXOVtR1ak57GUZHus9WxiZmvOMKqeTyHmsLpF-fIznS0lCdj0byGFA&actor_token_type=urn%3Aietf%3Aparams%3Aoauth%3Atoken-type%3Ajwt&subject_token_type=urn%3Aietf%3Aparams%3Aoauth%3Atoken-type%3Ajwt&subject_token=ewogICAgImFsZyI6ICJub25lIiwKICAgICJ0eXAiOiAiSldUIGp9.ewogICAgInVuaXF1ZV9uYW1lIjojIlRlc3QxIiwKICAgICJ0eXAiOiAxNTQ1MDU2ODc2LAogICAgImV4cCI6IDE1NzY3NjU2NzYsCiAgICAiaWF0IjojMTU0NTIyOTY3Ngp9.", "expires_in": 300, "token_type": "Bearer"}
```

Перевыпуск маркера доступа

Сценарий описывается в разделе 1.5 [RFC6749](#).

Данный раздел описывает сценарий организации взаимодействия с ЦИ DSS с использованием маркеров обновления маркеров доступа.

В целях повышения безопасности протокол OAuth 2.0 выпускает маркеры доступа, имеющие ограниченное время жизни.

От времени жизни маркера напрямую зависят безопасность информации, хранящейся на сервере, и удобство пользования оным. Уменьшение времени жизни маркера приводит к сокращению размера окна, в течении которого злоумышленник, перехвативший маркер, имеет доступ информации. Одновременно с этим увеличивается количество запросов на авторизацию, которые должен подтверждать пользователь.

Компромиссным решением в такой ситуации является использование т.н. **маркера обновления** (refresh_token). Маркером обновления называется долговременный маркер безопасности, выпускаемый Центром Идентификации. Он позволяет осуществлять прозрачный для пользователя перевыпуск маркера доступа.

Сценарий взаимодействия с ЦИ DSS, включающий в себя механизм маркеров обновления, может быть изображен следующей схемой:



1. Клиентское приложение осуществляет запрос маркера доступа и для этого передает разрешение на авторизацию Центру Идентификации ([подробнее](#)).
2. Центр Идентификации передает клиентскому приложению маркер доступа и маркер обновления для последующего перевыпуска маркеров доступа ([подробнее](#)).
3. Пользователь обращается к Сервису Подписи с полученным маркером доступа.
4. Сервис Подписи возвращает защищенный ресурс клиентскому приложению.
5. Пользователь продолжает совершать запросы к Сервису до тех пор, пока маркер не перестанет быть валидным.
6. Сервис Подписи возвращает ошибку, сигнализирующую о том, что маркер доступа перестал быть валидным.
7. Клиентское приложение инициирует процедуру перевыпуска маркера доступа, используя имеющийся у него маркер обновления ([подробнее](#)).
8. Центр Идентификации после проверки маркера обновления выпускает приложению новый маркер доступа и, опционально, новый маркер обновления.

Запроса маркеров доступа и обновления

Для запроса маркера доступа с маркером обновления приложению необходимо сформировать запрос на получение кода авторизации, имеющий следующий вид:

Пример запроса

```
GET https://<hostname>/<stsappname>/oauth/authorize?
response_type=code&scope=dss+offline_access&redirect_uri=urn:ietf:wg:oauth:2.0:oob:auto&resource=urn:cryptopro
:dss:signserver:signserver&client_id=TestClient HTTP/1.1
cache-control: no-cache
Accept: */*
accept-encoding: gzip, deflate
Connection: close
```

Параметры запроса:

- `client_id` - идентификатор клиента OAuth, зарегистрированный на ЦИ DSS. Для регистрации клиента и его последующей конфигурации можно воспользоваться командлетами Windows PowerShell **Add-DssClient** и **Set-DssClient** соответственно.

Примечание

При регистрации клиента параметр `AllowedFlow` должен содержать значение `AuthorizationCode`.

Примечание

Для включения возможности получения маркера обновления используется параметр `-AllowedFlow RefreshToken`. (В версиях КриптоПро DSS 2882 и более ранних данный параметр назывался `-AllowOfflineAccess`).

Параметр используется только в Исполнениях «DSS + myDSS», «DSS SDK» и «DSS Client SDK», «DSS + AirKey Lite».

- `response_type` - в данном сценарии имеет значение `code`.
- `scope` - области использования маркера. Должен содержать значение `dss`, а также значение `offline_access`, сообщающее ЦИ о необходимости включения маркера обмена при выдаче маркера доступа.
- `redirect_uri` - зарегистрированный на Центре Идентификации адрес возврата (по этому адресу будет возвращён запрошенный код авторизации).

Примечание

Значение параметра должно соответствовать значению адреса возврата, заданного при регистрации клиента на ЦИ.

Для случаев, в которых не планируется использование выделенного HTTP-сервиса для обработки URI перенаправления, рекомендуется использовать зарезервированный URI `urn:ietf:wg:oauth:2.0:oob:auto`.

- `resource` - идентификатор ресурса, для доступа к которому выпускается токен. В случае Сервиса Подписи идентификатор фиксирован и имеет вид `urn:cryptopro:dss:signserver:<signserverAppName>`.

Если в рамках сценария необходима аутентификация клиентского приложения и известен его секрет, запрос необходимо модифицировать.

Параметр `client_id` в теле запроса должен был заменен на заголовок `Auhtorization` HTTP-запроса, имеющий значение `Basic Base64(client_id:client_secret)`.

В ответ на запрос ЦИ может перенаправить пользователя на страницу аутентификации. Процесс аутентификации может занять несколько шагов и состоять из множества перенаправлений на другие ресурсы.

После успешной аутентификации ЦИ сформирует следующий ответ, который отправит на `redirect_uri`:

```
HTTP/1.1 302 Found
Location: urn:ietf:wg:oauth:2.0:oob:auto#code=c86322fd3a4cf85fb51249ab3fb4fcd1
Content-Length: 0
```

Параметр `code` (стоящий после символа #) содержит код авторизации, который необходимо извлечь из заголовка

Location.

Типовые ошибки

HTTP-код	ошибка	описание
400	invalid_client	OAuth-клиент не зарегистрирован или неверно указан clientID.
400	unauthorized_client	OAuth-клиент использует незарегистрированный сценарий аутентификации (Flow) или был передан некорректный <code>redirect_uri</code> .
400	invalid_request	Неверно сформирован параметр <code>resource</code> .
500	An error has occurred	1. Проверяющая сторона с идентификатором <code>resource</code> не зарегистрирована.

Для получения маркера доступа используется конечная точка **/token**. Клиент формирует следующий HTTP-запрос:

```
POST https://<hostname>/<stsappname>/oauth/token HTTP/1.1
Content-Type: application/x-www-form-urlencoded
cache-control: no-cache
Accept: */*
content-length: 139

grant_type=authorization_code&code=9e554074113426cb2f4430f51b68170a&redirect_uri=urn:ietf:wg:oauth:2.0:oob:auto&client_id=sample
```

Параметры запроса:

- `grant_type` - в данном сценарии имеет значение `authorization_code`.
- `code` - код авторизации, полученный на предыдущем этапе.
- `redirect_uri` - зарегистрированный на Центре Идентификации адрес возврата (по этому адресу будет возвращён запрошенный код авторизации).

Примечание

Значение параметра должно соответствовать значению адреса возврата, заданного при регистрации клиента на ЦИ.

- `client_id` - идентификатор клиента OAuth, зарегистрированный на ЦИ DSS.

Если в рамках сценария необходима аутентификация клиентского приложения и известен его секрет, запрос необходимо модифицировать.

Параметр `client_id` в теле запроса должен быть заменен на заголовок Authorization HTTP-запроса, имеющий значение `Basic Base64(client_id:client_secret)`.

Получение маркера доступа и маркера обновления

В случае успешной обработки запроса Центром Идентификации ответ будет содержать:

- `access_token` - Маркер доступа, выпущенный Центром Идентификации DSS
- `token_type` - Тип токена
- `expires_in` - Время жизни токена в секундах
- `refresh_token` - Маркер обновления, выпущенный Центром Идентификации DSS

Внимание!

В силу того, что маркер обновления является долговременным и позволяет легко получить маркер доступа на его основе, факт компрометации маркера обновления клиентским приложением приводит к компрометации всей пользовательской информации.

Необходимо применять усиленные меры безопасности при работе с данным маркером и в случае установления факта компрометации данный маркер должен быть инвалидирован.

Значение параметра `access_token` необходимо будет использовать при обращениях к Сервису Подписи.

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 2268
Content-Type: application/json; charset=utf-8
Expires: -1

{
  "access_token": "eyJ0eXAiOiJKV1Q...LnS1sAunDSE1hh3A5n8W7lhPSM4z_VA",
  "expires_in": 300,
  "token_type": "Bearer",
  "refresh_token": "774fae55a7e64c8238c35695c4198198"
}
```

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_client	OAuth-клиент не зарегистрирован или неверно указан clientID.
400	unauthorized_client	OAuth-клиент использует незарегистрированный сценарий аутентификации (Flow) или был передан некорректный <code>redirect_uri</code> .
400	invalid_request	Неверно сформирован параметр <code>resource</code> .
400	invalid_grant	Невалидный код авторизации.
500	An error has occurred	1. Проверяющая сторона с идентификатором <code>resource</code> не зарегистрирована.

Пример сообщения об ошибке

```
HTTP/1.1 400 Bad Request
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Date: Fri, 21 Dec 2018 13:46:42 GMT
Connection: close

{"error": "invalid_client"}
```

Запрос перевыпуска маркера доступа

Для того, чтобы клиентское приложение, в случае истечения срока валидности маркера доступа, могло запросить у Центра Идентификации новый, необходимо сформировать и отправить на ЦИ следующий запрос:


```
POST http://<hostname>/<stsappname>/oauth/token HTTP/1.1
Content-Type: application/x-www-form-urlencoded
accept-encoding: gzip, deflate
content-length: 92
Connection: keep-alive

grant_type=refresh_token&client_id=TestClient&refresh_token=774fae55a7e64c8238c34995c4198198
```

Параметры запроса:

- `grant_type` - в данном сценарии имеет значение `refresh_token`.
- `refresh_token` - маркер обмена, полученный от Центра Идентификации.
- `client_id` - идентификатор клиента OAuth, зарегистрированный на ЦИ DSS.

Если в рамках сценария необходима аутентификация клиентского приложения и известен его секрет, запрос необходимо модифицировать.

Параметр `client_id` в теле запроса должен был заменен на заголовок Authorization HTTP-запроса, имеющий значение `Basic Base64(client_id:client_secret)`.

Пример ответа сервера

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 2268
Content-Type: application/json; charset=utf-8
Expires: -1

{
  "access_token": "eyJ0eXAiOiJKV1Q...LnS1sAunDSE1hh3A5n8W71hPSM4z_VA",
  "expires_in": 300,
  "token_type": "Bearer",
  "refresh_token": "774fae55a7e64c8238c35695c4198198"
}
```

Маркер обновления и вторичная аутентификация

Использование маркера обновления позволяет пропускать аутентификацию пользователя (в т.ч. и вторичную) при получении маркера доступа через конечную точку 'oauth/token', при этом подтверждение операций, требующих доступ к ЗК, а также подтверждение произвольных операций всё равно будет требовать использования одного из методов вторичной аутентификации.

Описание ошибок протокола OAuth 2.0

В данном разделе приводится описание ошибок, возникающих при выполнении запросов в рамках протокола OAuth 2.0 и OpenId Connect 1.0.

В случае возникновения ошибок сервер возвращает информацию в двух полях:

- `error` - код ошибки.
- `error_description` - описание ошибки.

Далее будет представлена информация по кодам ошибок.

Ошибки конечной точки `/authorize`

`invalid_request`

Неправильный запрос.

В запросе не передан обязательный параметр, либо значения переданного параметра некорректно, либо параметр присутствует в запросе несколько раз, либо весь запрос имеет неправильный формат.

Возможные причины

- Отсутствие обязательных параметров в запросе (`client_id`, `redirect_uri`, `resource`, `response_type`).
- Переданы некорректные значения параметров (неправильный формат `redirect_uri`, `resource`).
- Переданы незарегистрированные значения параметров (незарегистрированный `resource`).
- Переданы неподдерживаемые значения параметров (неподдерживаемый `response_mode`).
- Передано некорректное значение параметра `id_token_hint`.
- Маркер, указанный в значении параметра `id_token_hint`, невалиден.
- Сервер не настроен на обработку параметра `id_token_hint`.

`unauthorized_client`

Неавторизованный клиент.

Клиент с указанным в запросе идентификатором не зарегистрирован, отключён, либо клиенту запрещено получения маркера доступа в рамках данного сценария.

Возможные причины

- По переданному в запросе `client_id` не найдено зарегистрированных клиентов.
- Переданный в запросе `client_id` принадлежит заблокированному клиенту.
- Используемый в запросе `redirect_uri` не зарегистрирован для используемого клиента.
- Используемый сценарий не разрешён для используемого клиента.

`unsupported_response_type`

Тип ответа не поддерживается.

Указанный в запросе `response_type` не поддерживается.

Возможные причины

- В запросе указан параметр `response_type` со значениями отличными от
 - `code`,
 - `token`,
 - `id_token`,

- id_token token,
- code id_token
- code token
- code id_token token.

invalid_scope

Неправильная область использования.

Указанный в запросе scope не зарегистрирован на сервере.

Возможные причины

- В запросе указан параметр scope, значение которого не зарегистрировано на сервере.
- В запросе указан параметр scope, заблокированный на сервере.
- Среди значений параметра scope указано больше одной области использования, требующей подтверждения.
- Переданные параметры dss_scope_params имеют неправильный формат.

login_required

Требуется аутентификация.

Запрос не может быть выполнен в интерактивном режиме (с указанием параметра prompt со значением none).

Возможные причины

- Запрос не может быть выполнен в неинтерактивном режиме, так как требуется аутентификация пользователя.

Ошибки конечной точки /token

invalid_requeset

Неправильный запрос.

В запросе не передан обязательный параметр, либо значения переданного параметра некорректно, либо параметр присутствует в запросе несколько раз, либо весь запрос имеет неправильный формат.

Возможные причины

- Не удалось получить идентификатор клиента из запроса.
- Отсутствие обязательных параметров в запросе (для сценария обмена маркеров: subject_token, actor_token_type, resource, grant_type, refresh_token).
- Переданы некорректные значения параметров (resource).
- Переданы незарегистрированные значения параметров (незарегистрированный resource).
- Переданы некорректные значения параметров (для сценария обмена маркеров: неправильный формат subject_token, actor_token).
- Маркеры доступа, указанные в значениях параметров subject_token, actor_token, не действительны.
- Переданы неподдерживаемые значения параметров (для сценария обмена маркеров: неподдерживаемый тип subject_token_type, subject_token_type).
- Сервер не настроен на поддержку сценария обмена маркеров.

invalid_client

Неправильный клиент.

Не удалось осуществить аутентификацию клиента.

Возможные причины

- Не удалось получить идентификатор клиента из запроса.
- Клиент с указанным в запросе идентификатором не зарегистрирован или отключён.

`invalid_grant`

Неправильное разрешение.

Разрешение, используемое клиентом, не является действительным.

Возможные причины

- Не передан параметр `password` в сценарии с использованием учётных данных владельца ресурсов.
- Сервер не настроен на обработку сценария с использованием учётных данных владельца ресурсов.
- Не удалось аутентифицировать пользователя по переданным `username` и `password`.
- Сценарий с использованием учётных данных владельца ресурсов не может быть использован для данной учётной записи пользователя из-за включенной вторичной аутентификации.
- Не передан код авторизации в сценарии с кодом авторизации.
- Переданный код авторизации истёк или не действителен.
- Переданный код авторизации был получен другим клиентом.

`unauthorized_client`

Неавторизованный клиент.

Клиент с указанным в запросе идентификатором не зарегистрирован, отключён, либо переданы неверные учётные данные клиента.

Возможные причины

- Используется тип учётных данных клиента, отличный от разделяемого секрета.
- Переданы неверные учётные данные.
- Переданные учётные данные клиента истекли.

`unsupported_grant_type`

Неподдерживаемый тип разрешения.

Разрешение, используемое клиентом, не поддерживается сервером.

Возможные причины

- Передан тип разрешения отличный от
 - `code`,
 - `password`,
 - `urn:ietf:params:oauth:grant-type:token-exchange`,
 - `refresh_token`.

Подтверждение произвольных операций

Центр Идентификации КриптоПро DSS позволяет подтверждать действия, требующие доступ к закрытому ключу сертификата пользователя. Информацию о параметрах выполняемой операции ЦИ получает от Сервиса Подписи через так называемый токен транзакции, идентификатор которого вызывающее приложение передаёт в запросе на подтверждении транзакции.

Однако ЦИ позволяет подтверждать не только операции доступа к закрытому ключу, но и произвольные действия. Это даёт мощный механизм подтверждения, использующий существующие способы аутентификации, реализованные в DSS.

Области использования маркера

Результатом подтверждения любой операции в ЦИ является **маркер безопасности** (токен). Для того чтобы указать, как будет применяться выпущенный маркер, используют т.н. **области использования маркера** (scope). Область использования маркера показывает для каких целей можно использовать выпущенный маркер. Однако следует понимать, что контроль за областью использования ложится на проверяющую сторону.

Шаблоны подтверждения

Каждой области использования маркера в DSS соответствует некоторый шаблон сообщения, используемый для отображения информации о действии пользователю. Шаблоны сообщений могут быть заданы для всех способов аутентификации (в терминологии системы оповещений DSS различным способам аутентификации соответствуют различные типы назначений - **destination**, а также различные типы получателей сообщений - **recipient**).

На данный момент в DSS существуют следующие типы назначений:

- **SMS** - соответствует способу аутентификации Otp-via-Sms.
- **Email** - соответствует способу аутентификации Otp-via-Email.
- **MobileAuth** - соответствует способу аутентификации через мобильное приложение myDSS.
- **MyDssAuth** - соответствует способу аутентификации через DSS Client SDK.
- **SimAuth** - соответствует способу аутентификации с помощью апплета на SIM карте.
- **Audit** - данный тип назначения используется для записи сообщения в журнал аудита
- **Challenge** - данный тип используется для форматирования сообщения, отображаемого пользователю в интерфейсе.

Типы получателей

- User - пользователь.
- Admin - оператор.
- Service - система (используется только для назначения **Аудит**)

Управление областями использования маркера

Для управления областями использования маркера администратору предоставляются следующие PowerShell команды:

- Add-DssScope - для добавления области использования маркера.
- Set-DssScope - для изменения параметров уже добавленной области использования.
- Remove-DssScope - для удаления области использования.

Форматы запросов

DSS предоставляет три возможности для передачи области использования маркера в зависимости от применяемого протокола:

- Передача в запросе на маркер безопасности (Request Security Token) для WS-Trust.
- Передача в запросе на авторизацию OAuth 2.0.
- Передача в запросе на подтверждение операции.

Рассмотрим каждый из трёх способов в отдельности.

WS-Trust

Для передачи области использования в запросе на выпуск маркера используется расширение протокола WS-Trust `AdditionalContext`, описание которого содержится в спецификации [WS-Federation](#):

```
<wst:RequestSecurityToken>
...
<auth:AdditionalContext>
  <auth:ContextItem Name="xs:anyURI" Scope="xs:anyURI" ? ...>
    (<auth:Value>xs:string</auth:Value> |
     xs:any ) ?
  </auth:ContextItem> *
  ...
</auth:AdditionalContext>
...
</wst:RequestSecurityToken>
```

Для указания области использования и параметров используются следующие имена элементов контекста:

`http://dss.cryptopro.ru/identity/claims/confirmationscope` и

`http://dss.cryptopro.ru/identity/claims/confirmationscopeparam`

Пример

```
<trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
  <wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
    <wsa:EndpointReference xmlns:wsa="http://www.w3.org/2005/08/addressing">
      <wsa:Address>https://relying-party/</wsa:Address>
    </wsa:EndpointReference>
  </wsp:AppliesTo>
  <trust:KeyType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/SymmetricKey</trust:KeyType>
  <trust:RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue</trust:RequestType>
  <auth:AdditionalContext xmlns:auth="http://docs.oasis-open.org/wsfed/authorization/200706">
    <auth:ContextItem Name="http://dss.cryptopro.ru/identity/claims/confirmationscope">
      <auth:Value>test-confirmation-scope</auth:Value>
    </auth:ContextItem>
    <auth:ContextItem Name="http://dss.cryptopro.ru/identity/claims/confirmationscopeparam">
      <auth:Value>Param1.ValueOfParam1</auth:Value>
    </auth:ContextItem>
  </auth:AdditionalContext>
</trust:RequestSecurityToken>
```

OAuth 2.0

В протоколе OAuth 2.0 область использования маркера указывается в `authorize` запросе в параметре `scope`, а подстановочные значения в параметре `dss_scope_params`.

Пример

```
GET /STS/oauth/authorize?client_id=eea2fd3f-5c70-4d74-a594-f1e7bf81b4d7
&response_type=code
&scope=openid+offline_access+dss+test-confirmation-scope
&redirect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aauto%3Aauto
&resource=https%3A%2F%2Fsimdss.cryptopro.ru%2FSignServer%2Frest%2Fapi
&dss_scope_params=eyJDCFRpbWUiOiIxNy4wMS4yMDE4IDE0OjQ5OjU1In0 HTTP/1.1
```

Подстановочные параметры должны быть предварительно закодированы по следующему алгоритму:

1. Все параметры и их значения представить в виде JSON объекта, в котором название свойства соответствует названию параметра, а значение свойства – значению параметра.
2. Полученный JSON-объект представить, как строку в кодировке UTF-8, и преобразовать в массив байтов.

3. Полученный массив байтов преобразовать в строку с помощью алгоритма BASE64URL.

Пример

Пусть в шаблоне требуется задать параметр `CpTime` со значением `17.01.2018 14:49:55`. Соответствующий этим данным JSON объект будет выглядеть `{"CpTime": "17.01.2018 14:49:55"}`.

В кодировке BASE64URL `eyJDcFRpbWUiOiIxNy4wMS4yMDE4IDE0OjQ5OjU1In0.`

При использовании следующего шаблона

Подтверждение тестовой операции. Время {0:CpTime}

на устройство пользователя будет отправлено сообщение:

Подтверждение тестовой операции. Время 17.01.2018 13:20:27

ЦИ проверяет, что все указанные в шаблоне параметры переданы в запросе, и, если это не так, вернёт ошибку. В запросе может быть указана только одна область использования маркера.

Строгое подтверждение

Для подтверждения произвольных операций через конечную точку **StrictConfirmation** необходимо передавать идентификатор области использования и параметры шаблона сообщения в запросе **RequestConfirmation**.

Пример

```
POST /STS/confirmation HTTP/1.1
Authorization: Bearer eyJ0...
Content-Type: application/json; charset=utf-8
Host: dss.cryptopro.ru

{
  "Resource": "https://dss.cryptopro.ru/SignServer/rest/api",
  "ClientId": "oauth-client-id",
  "ClientSecret": "oauth-client-secret",
  "ConfirmationScope": "test-confirmation-scope",
  "ConfirmationParams": {
    "CpTime": "17.01.2018 17:54:02"
  }
}
```

В данном примере в качестве идентификатора области использования передаётся значение **test-confirmation-scope**, а в качестве значения параметра **CpTime** - **17.01.2018 17:54:02**. При использовании следующего шаблона

Подтверждение тестовой операции. Время {0:CpTime}

на устройство пользователя будет отправлено сообщение:

Подтверждение тестовой операции. Время 17.01.2018 13:20:27

ЦИ проверяет, что все указанные в шаблоне параметры переданы в запросе, и, если это не так, вернёт ошибку.

В запросе может быть указана только одна область использования маркера.

Отмена операции

Для отмены существующей операции необходимо передавать идентификатор области использования, идентификатор операции и значение "Cancel" для действия управления состоянием операции.

Пример

Пример

```
POST /STS/v2.0/confirmation HTTP/1.1
Authorization: Bearer eyJ0...
Content-Type: application/json; charset=utf-8
Host: dss.cryptopro.ru

{
  "Resource": "https://dss.cryptopro.ru/SignServer/rest/api",
  "ClientId": "oauth-client-id",
  "ClientSecret": "oauth-client-secret",
  "ChallengeResponse": {
    "ControlChallengeResponse": {
      "RefId": "a7b55177-a974-46a4-ad0d-046b14f2654e",
      "ControlAction": "Cancel"
    }
  }
}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
  "IsFinal": true,
  "IsError": true,
  "Error": "authentication_cancelled"
}
```

Передача бинарных данных

При подтверждении произвольной операции можно в качестве параметров передать бинарные данные, которые будут преобразованы на стороне Центра Идентификации в [DTBS](#).

Для передачи бинарных данных необходимо расширить запрос на выпуск маркера параметрами `ConfirmationData` и `ConfirmationDataType`. В первом параметр передаётся содержимое бинарных данных транзакции в BASE64, во втором тип бинарных данных. На основе типа данных ЦИ выполнит преобразование в DTBS.

Пример

В состав DSS входит плагин для преобразования бинарных данных в DTBS. Этот плагин осуществляет преобразование 1 к 1. На вход такому плагину в качестве бинарных данных следует передавать массив байтов, представляющий XML в кодировке UTF8:

Подтверждение операций в версии API 2.0

Версия API 2.0 позволяет подтверждать операции с несколькими бинарными данными, предварительно загруженными в Сервис Обработки Документов КриптоПро DSS (далее СОД).

Для этого в запрос на создание операции добавляется параметр `ConfirmationDataRefs`, значение которого представляет собой массив идентификаторов документов в СОД:

```
{
  "Resource": "{{resource}}",
  "ClientId": "oauth-client-id",
  "ClientSecret": "oauth-client-secret",
  "ConfirmationScope": "Test",
  "CallbackUri": "{{callback_uri}}",
  "ConfirmationParams": {
    "Param1": "Подстановочный параметр 1"
  },
  "ConfirmationDataRefs" : [
    "31FA0009-0968-4E5F-9B66-A1B6B53BA5C7",
    "DB1655AF-647E-44A4-8AB2-FF2B039B3BE1"
  ]
}
```

Примечание

Параметры `ConfirmationDataRefs` и `ConfirmationData` (вместе с `ConfirmationDataType`) являются взаимоисключающими.

Рассмотрим подтверждение произвольной операции через API 2.0 на следующем примере:

Исходные данные

- УЗ пользователя DSS с логином: `ee8473f2-f6c8-4527-a7bb-f0ec3d9f4131`
- Первичная аутентификация не требуется.
- Идентификатор произвольной операции: `CertificateConfirmOperation`.
- Во всех примерах переводы строк добавлены для удобства чтения.

Первичная аутентификация

Сначала получаем маркер доступа для отправки запроса на создание операции подтверждения.

Запрос

```
POST /STS/oauth/token HTTP/1.1
Host: dss-sdk.cryptopro.ru
Content-Type: application/x-www-form-urlencoded
Authorization: Basic MTJjZjR...

grant_type=password
&username=ee8473f2-f6c8-4527-a7bb-f0ec3d9f4131
&resource=urn:cryptopro:dss:signserver:SignServer
&password=
```

Ответ

```
{
  "access_token": "eyJ0...",
  "expires_in": 300,
  "token_type": "Bearer"
}
```

Загрузка документа в СОД

Бинарные данные для произвольной операции загружаются в СОД. Для аутентификации использует `access_token`, полученный на предыдущем шаге.

Запрос

```
POST /documentstore/api/documents HTTP/1.1
Host: dss-sdk.cryptopro.ru
Content-Type: application/octet-stream
CPDSS-POSTDOC: eyAiRmlsZU5hbWUiOiJ0ZXN0LnJ4dCIgfQ==
Authorization: Bearer eyJ0eXA...

"<file contents here>"
```

Ответ

```
{
  "DocumentId": "303306b3-c8df-4947-a9ce-9c4be24482bd"
}
```

Отправка запроса на создание операции подтверждения

Используя идентификатор документа, полученный на предыдущем шаге, формируем запрос на создание операции подтверждения.

Запрос

```
POST /STS/v2.0/confirmation HTTP/1.1
Host: dss-sdk.cryptopro.ru
Content-Type: application/json
Authorization: Bearer eyJ0eX...

{
  "Resource" : "urn:cryptopro:dss:signserver:SignServer",
  "ClientId": "oauth-client-id",
  "ClientSecret": "oauth-client-secret",
  "ConfirmationScope": "CertificateConfirmOperation",
  "ConfirmationDataRefs" : [
    "303306b3-c8df-4947-a9ce-9c4be24482bd"
  ]
}
```

Ответ

```
{
  "Challenge": {
    "Title": {
      "Value": "Подтвердите операцию на устройстве с помощью приложения."
    },
    "TextChallenge": [
      {
        "Label": "Подтверждение операции выпуск сертификата.",
        "ExpiresIn": 300,
        "CreatedAt": 1579670047,
        "ExpiresInSpecified": true,
        "IsHidden": false,
        "AuthnMethod": "http://dss.cryptopro.ru/identity/authenticationmethod/mydss",
        "RefID": "d4b63163-50ae-4b67-8a5a-50f15fe2fe5c",
        "Title": "Подтвердите операцию на устройстве с помощью приложения."
      }
    ]
  },
  "IsFinal": false,
  "IsError": false
}
```

Завершение подтверждения операции

После подтверждения операции в мобильном приложении требуется завершить подтверждение, отправив запрос:

Запрос

```
POST /STS/v2.0/confirmation HTTP/1.1
Host: dss-sdk.cryptopro.ru
Content-Type: application/json
Authorization: Bearer eyJ0eXA...

{
  "Resource" : "urn:cryptopro:dss:signserver:SignServer",
  "ClientId": "oauth-client-id",
  "ClientSecret": "oauth-client-secret",
  "ChallengeResponse" : {
    "TextChallengeResponse" : [ {
      "RefId" : "d4b63163-50ae-4b67-8a5a-50f15fe2fe5c"
    } ]
  }
}
```

Ответ

```
{
  "AccessToken": "eyJ0eXAi...",
  "IsFinal": true,
  "IsError": false
}
```

Значение поля `IsFinal` равно `true` и значение поля `IsError` равно `false` свидетельствуют о том, что подтверждение завершилось успешно.

Полчение информации о подтверждённых действиях

В случае успешного подтверждения информацию об операции можно получить с помощью следующего запроса:

Запрос

```
GET /STS/operations/d4b63163-50ae-4b67-8a5a-50f15fe2fe5c HTTP/1.1
Host: dss-sdk.cryptopro.ru
Authorization: Bearer eyJ0eXAiO...
```

Здесь `d4b63163-50ae-4b67-8a5a-50f15fe2fe5c` является идентификатором операции, он совпадает с идентификатором операции подтверждения.

Ответ

```
{
  "Id": "8cd47c3b-897c-4cc8-bf85-49dc432dd50d",
  "Type": "CertificateConfirmOperation",
  "Parameters": null,
  "Description": "Подтверждение операции выпуск сертификата.",
  "State": "Confirmed",
  "CreatedAt": 1579677157,
  "CompleteBefore": 0,
  "ConfirmBefore": 1579677457,
  "ConfirmedAt": 1579677182,
  "CompletedAt": 0,
  "UpdatedAt": 1579677182,
  "UserId": "4e22dad2-ffd1-4bdd-8451-1a29e2cf3a22",
  "Context": null,
  "Proof": null,
  "AuthenticationType": "http://dss.cryptopro.ru/identity/authenticationmethod/mydss",
  "ExternalId": "8cd47c3b-897c-4cc8-bf85-49dc432dd50d",
  "Actions": [
    {
      "Id": "4c7edc0f-2479-4bcd-8550-590c53b781ab",
      "DocumentId": "303306b3-c8df-4947-a9ce-9c4be24482bd",
      "OriginalDocumentId": null,
      "Status": "Approved",
      "State": "Pending",
      "ResultValue": null,
      "Error": null,
      "ErrorDescription": null
    }
  ]
}
```

Статус операции подтверждения произвольной операции после успешного подтверждения всегда `Confirmed`, если в операции участвовали документы, то соответствующие действия имеют статус `Approved` (а состояние `Pending`, так как документы не обрабатывались после подтверждения никакими сервисами).

Тип операции совпадает со значением `ConfirmationScope`, указанном в запросе на создание операции подтверждения.

Ограничения областей использования

Для OAuth клиента можно задать список допустимых областей использования, которые разрешено использовать при запросе маркера доступа во всех OAuth-сценариях и в сценарии строгого подтверждения. Если допустимые области использования ограничены для клиента, то только они могут быть переданы в запросах в соответствующих каждому сценарию параметрах.

Для того чтобы задать способ список допустимых областей использования, необходимо выполнить следующую PowerShell команду:

```
Set-DssClient -ClientId ScopeRestrictedClientId -AllowedScopes scope1,scope2
```

здесь

- `ScopeRestrictedClientId` - идентификатор OAuth-клиента, для которого настраивается ограничение;
- `scope1,scope2` - идентификаторы областей использования, разрешённые для данного OAuth-клиента.

Запоминание подтверждения

Существует возможность запомнить факт подтверждения операции после первого успешного прохождения процедуры вторичной аутентификации. Для этого в свойствах подтверждаемой области использования необходимо выставить свойство `RememberConsent` в значение `true` с помощью команды:

```
Set-DssScope -ScopeName RememberConsentScope -RememberConsent true
```

здесь

- `RememberConsentScope` - название области использования, для которой меняется настройка.

После применения настройка факт первого успешного подтверждения будет автоматически запомнен.

Разрешение на выполнение действий

С помощью одновременного применения ограничения на допустимые области использования маркера и запоминания факта подтверждения можно организовать сценарий взаимодействия с DSS, при котором у пользователя однократно будет запрошено разрешение на выполнение какого-то действия для конкретного OAuth- клиента. Далее этого сценарий будет рассмотрен подробнее.

Допустим, что существует некоторое веб-приложение `DemoBank`, вход в которое осуществляется через КриптоПро DSS. Есть требование запрашивать перед первым использованием приложения разрешение у пользователя на доступ к учётной записи ЦИ КриптоПро DSS.

Для решения данной задачи администратор DSS создает новую область использования `dss-access` с шаблоном "Доступ к учётной записи КриптоПро DSS". Для данной области использования включается требование двухфакторной аутентификации при подтверждении и возможность сохранения факта подтверждения:

```
Add-DssScope -ScopeName "dss-access" -RequireConfirmation true -RememberConsent true -DisplayName "Доступ к учётной записи КриптоПро DSS"
```

Далее для OAuth-клиента `DemoBank` включается требование подтверждения области использования и список допустимых областей:

```
Set-DssClient -ClientId DemoBank -RequireConsent 1 -AllowedScopes "dss-access"
```

После применения настроек любой запрос на получения маркера доступа без указания параметра `scope` или с указанием значения, отличного от `dss-access` приведёт к ошибке `invalid_scope`. При этом запрос с указанием `dss-access` приведёт к ошибке `consent_required`, означающей, что не было получено разрешение пользователя на область использования `dss-access` для OAuth-клиента `DemoBank`.

Для получения разрешения необходимо отправить следующий запрос на конечную точку `/confirmation`:

Запрос

```
POST /STS/confirmation HTTP/1.1
Authorization: Basic c2dhOg==
Content-Type: application/json

{
  "Resource" : "urn:cryptopro:dss:signserver:SignServer",
  "ConfirmationScope" : "dss-access",
  "ClientId" : "cryptopro.dss.samples",
  "ClientSecret" : "WbENKmg-a0Kada6d0PpsPzJ6LIVT26b7"
}
```

После подтверждения успешный результат будет запомнен и приложение сможет получать маркеры доступа, указывая в запросах область использования `dss-access`.

Отображаемые данные

Сервис Подписи «КриптоПро DSS» позволяет формировать отображаемое представление обрабатываемого документа, которое может быть показано пользователю в процессе подтверждения операции.

Для этого на Сервисе Подписи предусмотрен механизм плагинов, преобразующих документы в XML особого вида. Схема XML-документа приведена ниже:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema targetNamespace="http://www.cryptopro.ru/schemas/2014/08/dtbs"
  elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:dtbs="http://www.cryptopro.ru/schemas/2014/08/dtbs">
  <xs:element name="dtbs" type="dtbs:dtbsType" />
  <xs:complexType name="dtbsType">
    <xs:sequence>
      <xs:element ref="dtbs:row" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="unattendedSign" type="xs:boolean">
      <xs:annotation>
        <xs:documentation>
          Если 'true', то документ подписывается без отображения и
          подтверждения пользователем. В этом случае элементы row
          должны отсутствовать.
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>

  <xs:element name="row" type="dtbs:rowType" />
  <xs:complexType name="rowType">
    <xs:sequence>
      <xs:element name="name" type="xs:string">
        <xs:annotation>
          <xs:documentation>
            Название ключевого поля выжимки документа
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="value" type="xs:string">
        <xs:annotation>
          <xs:documentation>
            Значение ключевого поля выжимки документа
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Отображение подписываемых данных происходит, когда на Сервисе Подписи включен режим подтверждения операций. Во время создания транзакции клиентское приложение передаёт на сервер содержимое документа и его тип. Если по типу документа удалось подобрать соответствующий плагин для формирования отображаемого представления, то сервис подписи сохранит их вместе со сведениями об операции.

Центр Идентификации при подтверждении операции, используя отображаемые данные, формирует текст, отображаемый пользователю в ходе процедуры аутентификационного испытания. Например, при формировании SMS сообщения с одноразовым паролем или отображении на Веб-интерфейсе пользователя к XML-документу применяется XSL-преобразование, преобразующее его в текст вида:

```
Name1 : Value1, Name2 : Value2, ... , NameN : ValueN.
```


Пример отображаемых данных:

```
<?xml version="1.0" encoding="utf-8"?>
<dtbs xmlns="http://www.cryptopro.ru/schemas/2014/08/dtbs">
  <row>
    <name>Наименование документа</name>
    <value>Платёжное поручение</value>
  </row>
  <row>
    <name>Банк получателя</name>
    <value>АКБ "Рога и копыта"</value>
  </row>
  <row>
    <name>Счёт получателя</name>
    <value>40781032100000000000</value>
  </row>
  <row>
    <name>Сумма платежа</name>
    <value>100 RUB</value>
  </row>
</dtbs>
```

Время жизни транзакции

Вызывающая система может передать в запросе на создание транзакции желаемый срок её действия.

Срок действия транзакции не может превышать максимально допустимое значение, задаваемое администратором сервиса.

Указание желаемого срока действия транзакции зависит от используемого протокола.

OAuth 2.0

При создании транзакции через конечную точку `authorize` желаемый срок действия передаётся через параметр запроса `dss_transaction_lifetime`. Срок действия транзакции указывается в секундах.

Пример

```
GET /STS/oauth/authorize?client_id=eea2fd3f-5c70-4d74-a594-f1e7bf81b4d7
&response_type=code
&scope=openid+offline_access+dss
&redirect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aaob%3Aauto
&resource=https%3A%2F%2Fsimdss.cryptopro.ru%2FSignServer%2Frest%2Fapi
&dss_transaction_lifetime=60 HTTP/1.1
```

Строгое подтверждение

При создании транзакции в протоколе строгого подтверждения для указания желаемого срока действия транзакции используется параметр `Tt1`. Срок действия транзакции указывается в секундах.

Пример

```
{
  "Resource": "{{resource}}",
  "ClientId": "oauth-client-id",
  "ClientSecret": "oauth-client-secret",
  "CallbackUri": "{{callback_uri}}",
  "Tt1" : "60"
}
```

Срок действия транзакции будет возвращён в элементе `ExpiresIn` соответствующего испытания.

Пример

```
{
  "Challenge": {
    "Title": {
      "Value": "Подтвердите операцию на устройстве с помощью приложения."
    },
    "TextChallenge": [
      {
        "AuthnMethod": "http://dss.cryptopro.ru/identity/authenticationmethod/mobile",
        "RefID": "3c672c35-947d-4f46-bd9e-5dd948a2107d",
        "MaxLenSpecified": false,
        "HideTextSpecified": false,
        "ExpiresIn": 60,
        "ExpiresInSpecified": true
      }
    ],
    "ContextData": {
      "RefID": "3c672c35-947d-4f46-bd9e-5dd948a2107d"
    }
  },
  "IsFinal": false,
  "IsError": false
}
```

WS-Trust

При получении маркера доступа через протокол **WS-Trust** желаемый срок действия транзакции можно передать через параметр контекста аутентификации `http://dss.cryptopro.ru/identity/claims/transactionlifetime`

Пример

```
<trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
  <wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
    <wsa:EndpointReference xmlns:wsa="http://www.w3.org/2005/08/addressing">
      <wsa:Address>https://relying-party/</wsa:Address>
    </wsa:EndpointReference>
  </wsp:AppliesTo>
  <trust:KeyType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/SymmetricKey</trust:KeyType>
  <trust:RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue</trust:RequestType>
  <auth:AdditionalContext xmlns:auth="http://docs.oasis-open.org/wsfed/authorization/200706">
    <auth:ContextItem Name="http://dss.cryptopro.ru/identity/claims/transactionlifetime">
      <auth:Value>60</auth:Value>
    </auth:ContextItem>
  </auth:AdditionalContext>
</trust:RequestSecurityToken>
```

Аутентификация через сторонние ЦИ

СЭП КриптоПро DSS поддерживает аутентификацию в сторонних ЦИ по двум протоколам:

1. WS-Federation (далее [WsFed](#))
2. OpenId Connect 1.0 (далее [Oidc](#))

Настройка доверия к стороннему ЦИ

Настройка доверия к стороннему ЦИ осуществляется Администратором DSS через PowerShell с помощью следующих командлетов:

- `Add-DssIdentityProvider` - добавление нового доверенного ЦИ.
- `Set-DssIdentityProvider` - изменение параметров существующего доверенного ЦИ.
- `Get-DssIdentityProvider` - отображение списка доверенных ЦИ.
- `Remove-DssIdentityProvider` - удаление доверенного ЦИ из списка.
- `Disable-DssIdentityProvider` - отключение доверенного ЦИ.
- `Enable-DssIdentityProvider` - включение доверенного ЦИ.
- `Set-DssIdentityProviderOidcEndpoint` - изменение настроек аутентификации протокола Oidc.
- `Set-DssIdentityProviderWsFedEndpoint` - изменение настроек аутентификации протокола WsFed.

Настройка аутентификации по протоколу OpenId Connect 1.0


Для подключения стороннего ЦИ по протоколу Oidc необходимы следующие данные:

1. URL адрес конечной точки авторизации (AuthorizationEndpoint).
2. URL адрес конечной точки распространения набора ключей (JwksUri).
3. Самоназвание стороннего ЦИ (Issuer).
4. Идентификатор клиента (ClientId).
5. Секрет клиента (ClientSecret).
6. Запрашиваемые области использования (Scopes).

Использование Google в качестве стороннего ЦИ

Для примера подключим в качестве стороннего ЦИ Google.

Сначала необходимо зарегистрировать DSS, как OAuth-клиент в сервисе Google. Для этого требуется зайти в [консоль разработчика Google](#) и создать новый проект, нажав на кнопку **Create Project**:

 A project is needed to view enabled APIs and services

[Create Project](#)

Popular APIs and services

[VIEW ALL \(210\)](#)**Google...**

Google

The Google Drive API allows clients to access resources from Google Drive

**Gmail...**

Google

Flexible, RESTful access to the user's inbox

После создания проекта нужно включить API, к которому DSS в качестве OAuth 2.0 клиента будет запрашивать доступ. Таким API является **Google+**. Для включения API нужно нажать кнопку **ENABLE APIS AND SERVICES**

No APIs or services are enabled

Browse the [Library](#) to find and use hundreds of available APIs and services

Popular APIs and services

[VIEW ALL \(210\)](#)**Google...**

Google

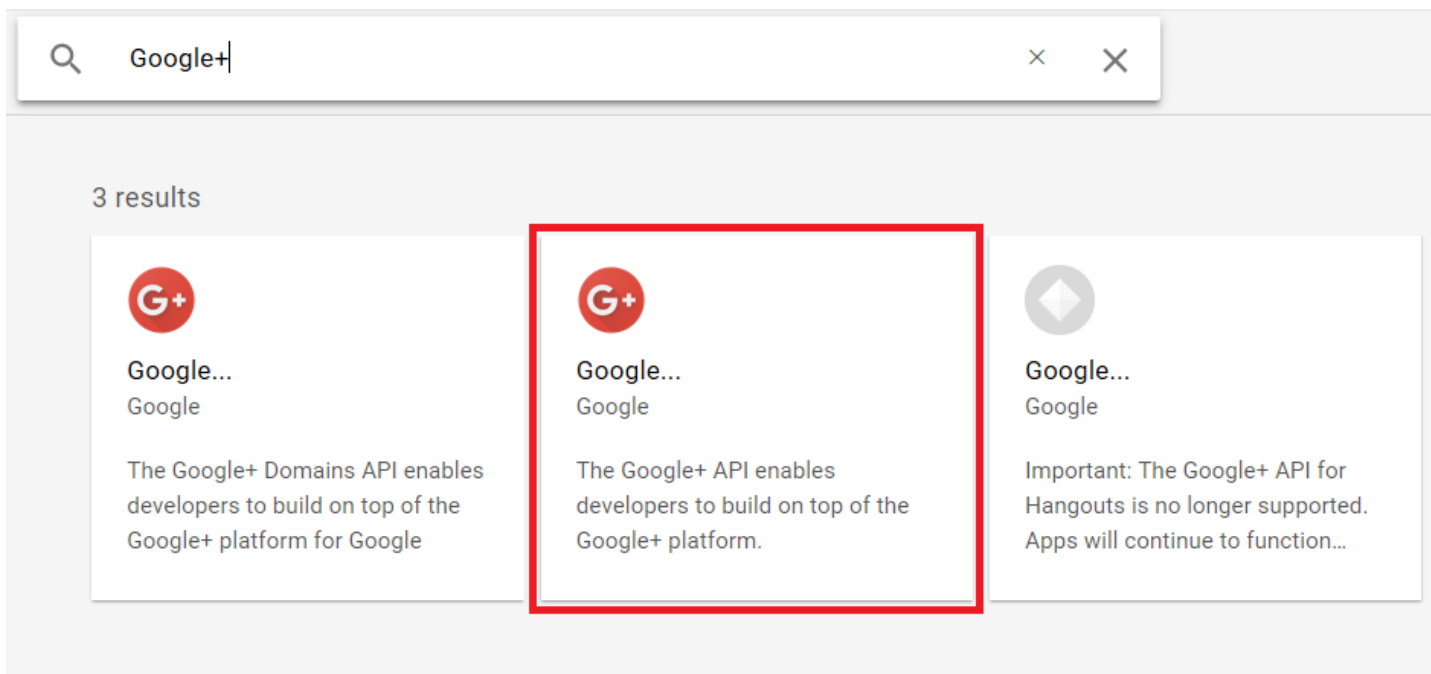
The Google Drive API allows clients to access resources from Google Drive

**Gmail...**

Google

Flexible, RESTful access to the user's inbox

и найти в поиске **Google+ API**



Следующим шагом является добавление учётных данных клиента для доступа к API. Для добавления необходимо нажать на кнопку **CREATE CREDENTIALS**

i To use this API, you may need credentials. Click 'Create credentials' to get started.

[CREATE CREDENTIALS](#)

Details

Name
Google+ API

By
Google

Service name
plus.googleapis.com

Overview
The Google+ API enables developers to build on top of the Google+ platform.

Activation status
Enabled

Tutorials and documentation

[Learn more](#)

[Try in API Explorer](#)

Traffic by response code

Request/sec (2 hr average)

No data is available for the selected time frame.

Aug 26 Sep 02 Sep 09 Sep 16

[View metrics](#)

В появившемся мастере можно сразу перейти к настройке Client Id, нажав на ссылку **client ID** в первом пункте

Add credentials to your project

1 Find out what kind of credentials you need

We'll help you set up the correct credentials

If you wish you can skip this step and create an API key, **client ID** or service account

Which API are you using?

Different APIs use different auth platforms and some credentials can be restricted to only call certain APIs.

Google+ API

Where will you be calling the API from?

Credentials can be restricted using details of the context from which they're called. Some credentials are unsafe to use in certain contexts.

Web server (e.g. node.js, Tomcat)

What data will you be accessing?

Different credentials are required to authorize access depending on the type of data that you request.

☒ User data

Access data belonging to a Google user, with their permission

☐ Application data

Access data belonging to your own application

What credentials do I need?

2 Get your credentials

Cancel

Прежде, чем ввести параметры OAuth клиента требуется задать данные, отображаемые на экране согласия (consent screen). На этом экране пользователю будет предложено предоставить доступ DSS к запрашиваемым данным при аутентификации. Обязательными для заполнения являются адрес электронной почты владельца DSS и отображаемое название сервиса.

Email address ?

Product name shown to users ?

Homepage URL (Optional)

Product logo URL (Optional) ?



This is how your logo will look to end users
Max size: 120x120 px



The consent screen will be shown to users whenever you request access to their private data using your client ID. It will be shown for all applications registered in this project.

You must provide an email address and product name for OAuth to work.

Privacy policy URL

Optional until you deploy your app

Terms of service URL (Optional)

Save Cancel

После указания данных для экрана согласия можно указать параметры сервиса для протокола OAuth. Из обязательных полей имя клиента (отображается в списке подключенных клиентов в консоли разработчика) и адрес перенаправления. Также необходимо указать тип клиента **Web Application**. Адрес перенаправления имеет вид `https://<host>/<sts_apname>/Authentication/External`.

Application type

- ☒ Web application
- ☐ Android [Learn more](#)
- ☐ Chrome App [Learn more](#)
- ☐ iOS [Learn more](#)
- ☐ PlayStation 4
- ☐ Other

Name ?

Restrictions

Enter JavaScript origins, redirect URIs, or both

Authorized JavaScript origins

For use with requests from a browser. This is the origin URI of the client application. It can't contain a wildcard (`https://*.example.com`) or a path (`https://example.com/subdir`). If you're using a nonstandard port, you must include it in the origin URI.

Authorized redirect URIs

For use with requests from a web server. This is the path in your application that users are redirected to after they have authenticated with Google. The path will be appended with the authorization code for access. Must have a protocol. Cannot contain URL fragments or relative paths. Cannot be a public IP address.

Create Cancel


После создания клиента появится диалог, отображающий присвоенный `client_id` и `client_secret`, данный диалог можно игнорировать. Вместо этого удобно скачать параметры подключения в виде json-документа.

Create credentials ▾

Delete

Create credentials to access your enabled APIs. [Refer to the API documentation](#) for details.

OAuth 2.0 client IDs

<input type="checkbox"/> Name	Creation date ▾	Type	Client ID	
<input type="checkbox"/> cryptopro-dss	Sep 20, 2018	Web application	111672840743-cgvqrk67rtqo6ma2qn52vp1cd26ankvj.apps.googleusercontent.com	

Внутри JSON-документ содержит параметры OAuth клиента и адреса конечных точек, из которых интересует конечная точка авторизации (параметр `auth_uri`) <https://accounts.google.com/o/oauth2/auth> и URL для получения сертификатов подписи маркеров безопасности (параметр `auth_provider_x509_cert_url`) <https://www.googleapis.com/oauth2/v1/certs>.

По ссылке `auth_provider_x509_cert_url` можно получить список сертификатов в формате BASE64. Их необходимо установить в хранилище **Доверенные лица** локального компьютера, а их отпечатки прописать в параметрах подключенного стороннего ЦИ.

Однако следует отметить, что Google меняет свои сертификаты достаточно часто (примерно раз в 3 недели), поэтому можно настроить динамическое обновление ключей проверки маркеров. Для этого в параметрах ЦИ следует указать адрес распространения набора ключей подписи маркеров `JwksUri`, для Google этот адрес имеет значение <https://www.googleapis.com/oauth2/v3/certs>.

Для проверки маркера КриптоПро DSS пытается найти сертификат ключа проверки среди зарегистрированных сертификатов, если это не удаётся сделать, из маркера извлекается "самоназвание" ЦИ (для Google это `accounts.google.com`) и по нему находится зарегистрированный ЦИ. Если в настройках ЦИ задан `JwksUri`, то DSS загрузит текущие ключи подписи и с помощью них осуществит проверку пришедшего маркера.

У стороннего ЦИ можно попросить положить в выпускаемый маркер определённые утверждения, делается это через параметр `Scope`, в случае Google имеет смысл попросить добавить в маркер утверждение, содержащее адрес электронной почты (обратите внимание, что для Oidc протокола в `Scope` обязательно наличие значения `openid`).

С учётом всего вышеизложенного команда по настройке аутентификации через Oidc примет вид:

```
Set-DssIdentityProviderOidcEndpoint `
-IssuerName google `
-AuthorizationEndpoint https://accounts.google.com/o/oauth2/auth `
-JwksUri https://www.googleapis.com/oauth2/v3/certs `
-Issuer accounts.google.com `
-ClientId 23459005498-m89eo2br5q92nooqongdnm98qs8e5e4n.apps.googleusercontent.com `
-ClientSecret 102M4R6M8m9b9uF76Xij0B `
-Scopes "openid email"
```

ЦИ КриптоПро DSS ожидает, что в маркере будет соержаться определённый набор утверждений, но во многих случаях на срдержимое маркера, возвращаемого сторонним ЦИ, повлиять нельзядаже с помощью параметра `Scopes`. Для решения этой проблемы можно использовать правила преобразования утверждений. Подробно они рассматриваются в разделе [Правила преобразования утверждений](#). Для Google в качестве уникального идентификатора можно использовать email пользователя, а роль `Users` добавлять безусловно, тогда правила преобразования будут задаваться следующим образом:

```
[
  {
    "Name": "Pass through email",
    "Rules" : [
      "c:
[type==\"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress\"]=>issue(type=\"http://schemas.xml
Isoap.org/ws/2005/05/identity/claims/name\", value=c.value);",
      "=>issue(type=\"http://schemas.microsoft.com/ws/2008/06/identity/claims/role\",
value=\"Users\");",
    ]
  }
]
```

Настройка аутентификации по протоколу WS-Federation

Для подключения стороннего ЦИ по протоколу WsFed необходимы URL адрес конечной точки для обработки пассивного сценария (WsFedEndpointUri).

Настройка Microsoft ADFS в качестве стороннего ЦИ

Для ADFS адрес WsFedEndpointUri имеет вид https://<adfs-host>/adfs/ls. Тогда команда по настройке ADFS в качестве стороннего ЦИ примет вид

```
Set-DssIdentityProviderWsFedEndpoint -IssuerName ADFS -WsFedEndpointUri "https://<adfs-host>/adfs/ls"
```

Параметр login_hint

Часто клиентское приложение заранее знает логин пользователя, и было бы удобно сразу подставить значение логина в форме аутентификации. В качестве такой подсказки спецификация OpenID Connect 1.0 предлагает необязательный параметр **login_hint**.

Данный параметр представляет собой подсказку для Центра Идентификации, которую он использует на странице аутентификации. Если указанный параметр передан и пользователь с таким логином зарегистрирован и не заблокирован в DSS, то его значение будет использовано в качестве логина, тем самым пользователю останется только ввести свой пароль. Если же пользователя с таким логином нет или его учётная запись заблокирована, то параметр будет проигнорирован.

Параметр **login_hint** определён для двух сценариев: с использованием кода авторизации и с неявным разрешением (см. [Authorization Code Flow](#) и [Implicit Flow](#)).

Примеры запросов

Рассмотрим пример запроса, содержащего параметр **login_hint** (переносы строк добавлены для наглядности):

```
GET /authorize?
  client_id=eea2fd3f-5c70-4d74-a594-f1e7bf81b4d7
  &resource=https:%2f%2fdss.cryptopro.ru%2fSignServer%2frest%2fapi
  &redirect_uri=http:%2f%2f127.0.0.1:9158%2f
  &response_type=code
  &login_hint=sga HTTP/1.1
Host: dss.cryptopro.ru
```

Получение имени пользователя из id_token

Клиентское приложение может получить имя пользователя из id_token. Рассмотрим пример содержимого id_token:

```
{
  "unique_name": "sga",
  "dss_iss": "realsts",
  "dss_uuid": "49pDZfxUwFjeHhYSsrSX2LvdSxA=",
  "role": "Users",
  "dss_group": "Default",
  "at_hash": "yQAbEebC9sgmXrjonanKPQ",
  "iss": "realsts",
  "aud": "cryptopro.cloud.csp",
  "exp": 1527844466,
  "nbf": 1527844166
}
```

В id_token логин пользователя содержится в утверждении **unique_name**. В данном примере это **sga**. Однако клиентскому приложению следует обратить внимание не только на значение этого утверждения, но и на значение утверждения **dss_iss**. В нём содержится идентификатор Центра Идентификации, осуществившего аутентификацию пользователя. Использовать значение **unique_name** в качестве параметра **login_hint** можно только в случае, если **dss_iss** равен **realsts**. Значение **realsts** говорит о том, что в **unique_name** находится логин пользователя в ЦИ DSS, а не логин из стороннего ЦИ. Например, **unique_name** из такого маркера использовать нельзя:

```
{
  "unique_name": "sga@dss.cryptopro.ru",
  "dss_iss": "adfs",
  "dss_uuid": "49pDZfxUwFjeHhYSsrSX2LvdSxA=",
  "role": "Users",
  "dss_group": "Default",
  "at_hash": "yQAbEebC9sgmXrjonanKPQ",
  "iss": "realsts",
  "aud": "cryptopro.cloud.csp",
  "exp": 1527844466,
  "nbf": 1527844166
}
```

Режимы возврата ответа о результате авторизации

В OAuth 2.0 сценариях использования с кодом авторизации (AuthorizationCode) и с неявным разрешением (Implicit Flow) возврат ответа может осуществляться различными способами. В СЭП КriptoПро DSS поддерживаются три:

1. Возврат результата в строке запроса (`query`).
2. Возврат результата во фрагменте URL-адреса (`fragment`).
3. Возврат результата через отправку HTML-формы (`form_post`).

В скобках приведены значения параметра `response_mode` запроса на авторизацию, который необходимо указать для того, чтобы сервер вернул ответ в соответствующем режиме.

Возврат результата в строке запроса

При использовании данного метода сервис авторизации возвращает HTTP ответ со статусом `302 Found`, в заголовке `Location` указывается `redirect_uri`, переданный в запросе на авторизацию, все параметры ответа помещаются в строку запроса (query string).

Если в запросе на авторизацию явно не указан режим возврата ответа, то по умолчанию используется данный режим.

Пример 1

```
GET /STS/oauth/authorize?
client_id=eea2fd3f-5c70-4d74-a594-f1e7bf81b4d7
&resource=https%3A%2F%2Fdss.cryptopro.ru%2FSignServer%2Frest%2Fapi
&redirect_uri=http%3A%2F%2F127.0.0.1%3A9158%2F
&response_type=code HTTP/1.1

HTTP/1.1 302 Found
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Location: http://127.0.0.1:9158/?code=399c0f1dfc694f34f9d10ce7c5bc28a8
Content-Length: 0
```

Значения параметров кодируются с помощью кодировки `application/x-www-form-urlencoded` (см. [RFC-6749](#)).

Возврат результата в URL-фрагменте

Данный метод совпадает с предыдущим за исключением того, что параметры возвращаются в URL-фрагменте:

Пример 2

```
GET /STS/oauth/authorize?
client_id=eea2fd3f-5c70-4d74-a594-f1e7bf81b4d7
&resource=https:%2f%2fdss.cryptopro.ru%2fSignServer%2frest%2fapi
&redirect_uri=http:%2f%2f127.0.0.1:9158%2f
&response_type=token&response_mode=fragment HTTP/1.1

HTTP/1.1 302 Found
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Location: http://127.0.0.1:9158/#access_token=eyJ0e&token_type=Bearer&expires_in=300
Content-Length: 0
```

Возврат результата через отправку HTML формы

Данный режим подходит для возврата результата в том случае, когда значения параметров представляют собой длинные строки, так как часто браузеры и ОС накладывают ограничения на длину заголовков HTTP ответа, в том числе и заголовка `Location`.

В данном режиме сервис авторизации в ответ на запрос авторизации возвращает HTML страницу, содержащую HTML форму и JavaScript, который автоматически отправляет данную форму после загрузки страницы. Параметр `action` данной формы (т.е. адрес, куда она будет отправлена) устанавливается равным значению `redirect_uri` из запроса на авторизацию. Сама форма содержит поля (элементы `html input` элементы), соответствующие параметрам ответа.

Пример 3

```
GET /STS/oauth/authorize?
client_id=eea2fd3f-5c70-4d74-a594-f1e7bf81b4d7
&resource=https:%2f%2fdss.cryptopro.ru%2fSignServer%2frest%2fapi
&redirect_uri=http:%2f%2f127.0.0.1:9158%2f
&response_type=token
&response_mode=form_post HTTP/1.1

HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 2818
Content-Type: text/html
Expires: -1
Date: Wed, 15 Aug 2018 07:37:56 GMT

<!DOCTYPE html><html><head><title>Submit this form</title><meta name='viewport' content='width=device-width,
initial-scale=1.0' /></head>
<body><form method='post' action='http://127.0.0.1:9158/'>
<input type="hidden" name="access_token" value="eyJ0eXAi" />
<input type="hidden" name="token_type" value="Bearer" />
<input type="hidden" name="expires_in" value="300" />
</form><script>(function(){document.forms[0].submit();})();</script></body></html>
```

Возврат ошибок

Так как ошибки авторизации возвращаются как параметры ответа, для них используются те же режимы.

Пример 4

```
GET /STS/oauth/authorize?
client_id=eea2fd3f-5c70-4d74-a594-f1e7bf81b4d7
&resource=https:%2f%2fdss.cryptopro.ru%2fSignServer%2frest%2fapi
&redirect_uri=ru.cryptopro.dss:%2f%2foauth2redirect%2fcloudcsp
&response_type=token
&response_mode=form_post HTTP/1.1

HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 358
Content-Type: text/html
Expires: -1
Date: Wed, 15 Aug 2018 07:45:58 GMT

<!DOCTYPE html><html><head><title>Submit this form</title><meta name='viewport' content='width=device-width,
initial-scale=1.0' /></head><body>
<form method='post' action='ru.cryptopro.dss://oauth2redirect/cloudcsp'>
<input type="hidden" name="error" value="login_required" />
</form><script>(function(){document.forms[0].submit();})();</script></body></html>
```

Наличие ошибки можно определить по существованию параметра `error`, дополнительно в ответе может содержаться параметр `error_description`, содержащий более подробное описание ошибки.

Отмена запроса

Пользователь может решить отказаться от подтверждения той или иной операции. Для этого в веб-интерфейсе ЦИ DSS предусмотрена кнопка **Отмена**. При нажатии на эту кнопку ЦИ формирует следующий HTTP ответ (формат ответа может меняться в зависимости от режима формирования ответа см. [Возвращение результата авторизации](#)):

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 358
Content-Type: text/html
Expires: -1
Date: Wed, 15 Aug 2018 07:45:58 GMT

<!DOCTYPE html><html><head><title>Submit this form</title><meta name='viewport' content='width=device-width,
initial-scale=1.0' /></head><body>
<form method='post' action='https://dss.cryptopro.ru/oauth2redirect/cloudcsp'>
<input type="hidden" name="error" value="access_denied" />
<input type="hidden" name="error_description" value="cancelled_by_user" />
</form><script>(function(){document.forms[0].submit();})();</script></body></html>
```

В параметре `error` будет содержаться значение `access_denied` (это стандартный код ошибки, означающий, что пользователь отказался от предоставления доступа к своим ресурсам данному клиенту), а в `error_description` - `cancelled_by_user`.

Ошибки авторизации

Помимо стандартных значений параметра `error` ответа сервиса авторизации, описанных в [RFC-6749](#), в КриптоПро DSS вводятся два специальных значения.

Отмена операции

Пользователь СЭП может отменить процедуру многофакторной аутентификации. В этом случае на клиент будет возвращена со значением параметра `error` равным `access_denied` и фиксированным значением параметра `error_description` равным `cancelled_by_user`. Подробнее см. [Отмена запроса](#).

Недействительная лицензия

Данная ошибка возникает только при использовании КриптоПро Cloud CSP в случае, когда на клиенте и на сервере не было найдено действительной лицензии для использования данного продукта. Значения параметра `error` в этом случае `invalid_license`.

Взаимодействие с пользователем

Иногда OAuth 2.0 клиент заранее знает, что пользователь, работающий с ним, имеет активную сессию с сервисом авторизации. В этом случае клиент может попытаться получить авторизацию в автоматическом режиме без необходимости явного взаимодействия с пользователем через браузер.

Для выполнения запроса в автоматическом режиме необходимо в его параметрах передать дополнительный параметр `prompt` со значением `none`.

Пример

```
GET /STS/oauth/authorize?
  client_id=eea2fd3f-5c70-4d74-a594-f1e7bf81b4d7
  &resource=https:%2f%2fdss.cryptopro.ru%2fSignServer%2frest%2fapi
  &scope=dss
  &redirect_uri=http:%2f%2f127.0.0.1:9158%2f
  &response_type=code
  &prompt=none
Host: simdss.cryptopro.ru
```

При указании `prompt=none` результат будет возвращён сразу в HTTP-ответе. Если это невозможно сделать, в HTTP-ответе будет содержаться ошибка `login_required`, означающая, что запрос необходимо повторить но уже без параметра `prompt`.

Пример

```
HTTP/1.1 302 Found
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Location: http://127.0.0.1:9158/?error=login_required#=_
Content-Length: 0
```

Протокол обмена маркеров

Если клиентское приложение получило маркер от стороннего ЦИ, то оно может обменять этот маркер на маркер от ЦИ DSS.

Эта процедура описана в [OAuth20TE](#).

Для обмена маркерами клиент должен сформировать следующий запрос

```
POST /STS/oauth/token HTTP/1.1
Host: simdss.cryptopro.ru
Content-Type: application/x-www-form-urlencoded
cache-control: no-cache

grant_type=urn:iETF:params:oauth:grant-type:token-exchange
&client_id=12cf4adb-ba3e-40d0-a5d9-fa507ef26932
&resource=urn:ACryptopro:Adss:ASignServer:ASignServer
&subject_token=eyJ0
&subject_token_type=urn:iETF:params:oauth:token-type:jwt
```

В параметре `grant_type` передаётся значение `urn:iETF:params:oauth:grant-type:token-exchange`, в параметре `subject_token_type` указывается тип исходного маркера, в параметре `subject_token` передаётся маркер доступа от стороннего ЦИ. Остальные параметры являются стандартными для протокола OAuth: `resource` содержит идентификатор Сервиса Подписи, для которого необходимо выпустить результирующий маркер доступа, `client_id` - идентификатор зарегистрированного OAuth-клиента.

Допустимые значения `subject_token_type`:

- `urn:iETF:params:oauth:token-type:jwt`,
- `urn:iETF:params:oauth:token-type:saml1`,
- `urn:iETF:params:oauth:token-type:saml2`.

В случае `urn:iETF:params:oauth:token-type:saml1` или `urn:iETF:params:oauth:token-type:saml2` в `subject_token` передаётся закодированное в BASE64URL содержимое соответствующего маркера доступа.

Примеры

Далее рассматриваются различные сценарии использования протокола обмена маркерами на примере использования в качестве стороннего Центра Идентификации **Microsoft Active Directory Federation Service (ADFS)**.

ADFS поддерживает выпуск JWT маркеров, начиная с версии ОС Windows Server 2012 R2.

Пример 1. OAuth 2.0: код авторизации

Начиная с версии ОС Windows Server 2012 R2 служба ADFS, входящая в её состав, поддерживает один из сценариев протокола [OAuth20](#): сценарий с кодом авторизации.

Процесс авторизации состоит из следующих шагов.

1. Формирование запроса на авторизацию в ADFS.
2. Получение кода авторизации на ADFS.
3. Получение маркера доступа на ADFS.
4. Обмен маркера доступа ADFS на маркер доступа ЦИ DSS.

Во всех примерах переносы в теле запроса или URL добавлены для удобства чтения. Маркеры доступа и код авторизации сокращены.

Формирование запроса на авторизацию

Запрос на авторизацию представляет собой URL следующего вида

```
https://adfs.test-dss.local/adfs/oauth2/authorize
?client_id=oauth-client
&redirect_uri=http%3A%2F%2Flocalhost
&response_type=code
&resource=https%3A%2F%2Fdss.test-dss.local%2FSTS%2FIssue
```

- *client_id* - идентификатор OAuth-клиента, зарегистрированного в ADFS.
- *redirect_uri* - адрес перенаправления для возврата кода авторизации.
- *response_type* - тип возвращаемого результата. Для сценария с кодом авторизации всегда `code`.
- *resource* - идентификатор проверяющей стороны DSS, зарегистрированной в ADFS.

Получение кода авторизации

После формирования запроса на авторизацию, сформированный URL необходимо открыть в браузере пользователя.

После этого пользователь должен пройти аутентификацию (если пользователь в домене и использует браузер IE, то это произойдёт автоматически без необходимости вводить доменные учётные данные).

После успешной аутентификации ADFS вернёт следующий HTTP-ответ

```
HTTP/1.1 302 Found
Location: http://localhost/?code=X01...
Content-Length: 0
```

В параметре `code` находится код авторизации, используемый на следующем шаге для получения маркера доступа.

Получение маркера доступа

Для получения маркера доступа необходимо выполнить следующий запрос.

```
POST /adfs/oauth2/token HTTP/1.1
Host: adfs.test-dss.local
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code
&client_id=mf-oauth-client
&resource=https%3A%2F%2FDSS-2012.test-dss.local%2FSTS%2Fsts%2Fissue
$code=X01...
&redirect_uri=http%3A%2F%2Flocalhost
```

Параметры запроса аналогичны запросу на авторизацию. В параметре `code` передаётся код авторизации.

В ответ ADFS вернёт следующий JSON, содержащий маркер доступа.

```
{
  "access_token": "eyJ0eXAiOi...",
  "token_type": "bearer",
  "expires_in": 3600
}
```

Обмен маркера доступа ADFS на маркер доступа ЦИ DSS

Полученный на предыдущем шаге маркер доступа ADFS необходимо обменять на маркер ЦИ DSS с помощью запроса, указанного в начале статьи.

```
POST /STS/oauth/token HTTP/1.1
Host: simdss.cryptopro.ru
Content-Type: application/x-www-form-urlencoded
cache-control: no-cache

grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Atoken-exchange
&client_id=12cf4adb-ba3e-40d0-a5d9-fa507ef26932
&resource=urn%3Acryptopro%3Adss%3Asignserver%3ASignServer
&subject_token=eyJ0
&subject_token_type=urn%3Aietf%3Aparams%3Aoauth%3Atoken-type%3Ajwt
```

В ответ ЦИ вернёт новый маркер, который необходимо использовать для авторизации запросов к REST API Сервиса Подписи

```
{
  "access_token": "eyJ0eXAi...",
  "expires_in": 300,
  "token_type": "Bearer"
}
```

Пример 2. OAuth 2.0: учётные данные владельца ресурсов

Начиная с версии ОС Windows Server 2016 служба ADFS, входящая в её состав, поддерживает один из сценариев протокола [OAuth20](#): сценарий с учётными данными владельца ресурсов.

Процесс авторизации состоит из следующих шагов.

1. Формирование запроса на авторизацию в ADFS.
2. Получение маркера доступа от ADFS.
3. Обмен маркера доступа ADFS на маркер доступа ЦИ DSS.

Формирование запроса на авторизацию

Запрос на авторизацию имеет следующий вид:

```
POST /adfs/oauth2/token HTTP/1.1
Host: adfs.test-dss.local
Content-Type: application/x-www-form-urlencoded

grant_type=password
&client_id=mf-oauth-client
&resource=https%3A%2F%2FDSS-2012.test-dss.local%2FSTS%2Fsts%2Fissue
$username=test%40tdss.local
&password=1qaz%40WSX
```

Параметры запроса:

- *client_id* - идентификатор OAuth-клиента, зарегистрированного в ADFS.
- *redirect_uri* - адрес перенаправления для возврата кода авторизации.
- *resource* - идентификатор проверяющей стороны DSS, зарегистрированной в ADFS.
- *grant_type* - тип разрешения, в данном сценарии всегда имеет значение `password`.
- *username* - логин пользователя.
- *password* - пароль пользователя.

Получение маркера доступа

В ответ ADFS вернёт ответ в формате JSON, содержащий маркер доступа (в зависимости от настроек ADFS в ответе могут быть и другие маркеры, например, `refresh_token` и `id_token`).

```
{
  ...
  "access_token": "eyJ0eXAiOi...",
  "token_type": "bearer",
  "expires_in": 3600
  ...
}
```

Обмен маркера доступа ADFS на маркер доступа ЦИ DSS

Полученный на предыдущем шаге маркер доступа ADFS необходимо обменять на маркер ЦИ DSS с помощью запроса, указанного в начале статьи.

```
POST /STS/oauth/token HTTP/1.1
Host: simdss.cryptopro.ru
Content-Type: application/x-www-form-urlencoded
cache-control: no-cache

grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Atoken-exchange
&client_id=12cf4adb-ba3e-40d0-a5d9-fa507ef26932
&resource=urn%3Acryptopro%3Adss%3Asignserver%3ASignServer
&subject_token=eyJ0
&subject_token_type=urn%3Aietf%3Aparams%3Aoauth%3Atoken-type%3Ajwt
```

В ответ ЦИ вернёт новый маркер, который необходимо использовать для авторизации запросов к REST API Сервиса Подписи

```
{
  "access_token": "eyJ0eXAiOi...",
  "expires_in": 300,
  "token_type": "Bearer"
}
```

Пример 3. WS-Trust 1.3: маркер доступа SAML

Маркер доступа SAML может быть получен в ADFS по протоколу WS-Trust.

Процесс авторизации состоит из следующих шагов.

1. Формирование запроса на авторизацию в ADFS.
2. Получение маркера доступа на ADFS.
3. Преобразование содержимого маркера в BASE64URL.
4. Обмен маркера доступа ADFS на маркер доступа ЦИ DSS.

Формирование запроса на авторизацию

Запрос в рамках WS-Trust протокола имеет следующий вид:

```

<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:a="http://www.w3.org/2005/08/addressing"
xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <s:Header>
    <a:Action s:mustUnderstand="1">http://docs.oasis-open.org/ws-sx/ws-trust/200512/RST/Issue</a:Action>
    <a:MessageID>urn:uuid:cec1b3b2-be45-4ac6-9a4e-9dd4f00d2bce</a:MessageID>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To s:mustUnderstand="1">https://dss-adfs20.dss.cp.ru/adfs/services/trust/13/usernamemixed</a:To>
    <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
      <u:Timestamp u:Id="_0">
        <u:Created>2019-02-08T12:10:23.864Z</u:Created>
        <u:Expires>2019-02-08T12:15:23.864Z</u:Expires>
      </u:Timestamp>
      <o:UsernameToken u:Id="uuid-d5f0d96d-adee-439b-9b8d-e32728a2c101-2">
        <o:Username>sampleuser2@dss.cp.ru</o:Username>
        <o:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">P@ssw0rd</o:Password>
      </o:UsernameToken>
    </o:Security>
  </s:Header>
  <s:Body>
    <trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
      <wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
        <wsa:EndpointReference xmlns:wsa="http://www.w3.org/2005/08/addressing">
          <wsa:Address>http://simdss.cryptopro.ru/STS/Active.svc/token</wsa:Address>
        </wsa:EndpointReference>
      </wsp:AppliesTo>
      <trust:KeyType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Bearer</trust:KeyType>
      <trust:RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue</trust:RequestType>
    </trust:RequestSecurityToken>
  </s:Body>
</s:Envelope>

```

Получение маркера доступа

В ответ сервис вернёт следующие данные:

```

<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:a="http://www.w3.org/2005/08/addressing"
xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <s:Header>
    <a:Action s:mustUnderstand="1">http://docs.oasis-open.org/ws-sx/ws-
trust/200512/RSTRC/IssueFinal</a:Action>
    <a:RelatesTo>urn:uuid:cec1b3b2-be45-4ac6-9a4e-9dd4f00d2bce</a:RelatesTo>
    <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
      <u:Timestamp u:Id="_0">
        <u:Created>2019-02-08T12:09:37.994Z</u:Created>
        <u:Expires>2019-02-08T12:14:37.994Z</u:Expires>
      </u:Timestamp>
    </o:Security>
  </s:Header>
  <s:Body>
    <trust:RequestSecurityTokenResponseCollection xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-
trust/200512">
      <trust:RequestSecurityTokenResponse>
        <trust:Lifetime>
          <wsu:Created xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">2019-02-08T12:09:37.993Z</wsu:Created>
          <wsu:Expires xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">2019-02-08T13:09:37.993Z</wsu:Expires>
        </trust:Lifetime>
        <wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
          <wsa:EndpointReference xmlns:wsa="http://www.w3.org/2005/08/addressing">

```

```

        <wsa:Address>http://simdss.cryptopro.ru/STS/Active.svc/token</wsa:Address>
    </wsa:EndpointReference>
</wsp:AppliesTo>
<trust:RequestedSecurityToken>
    <xenc:EncryptedData Type="http://www.w3.org/2001/04/xmenc#Element"
xmlns:xenc="http://www.w3.org/2001/04/xmenc#">
        <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmenc#aes256-cbc"/>
        <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
            <e:EncryptedKey xmlns:e="http://www.w3.org/2001/04/xmenc#">
                <e:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmenc#rsa-oaep-
mgf1p">
                    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                </e:EncryptionMethod>
                <KeyInfo>
                    <o:SecurityTokenReference xmlns:o="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
                        <X509Data>
                            <X509IssuerSerial>
                                <X509IssuerName>CN=STS</X509IssuerName>
                            <X509SerialNumber>34807263445560130926767860687597093486</X509SerialNumber>
                            </X509IssuerSerial>
                        </X509Data>
                    </o:SecurityTokenReference>
                </KeyInfo>
                <e:CipherData>
                    <e:CipherValue>0lBqsS2HNg5P/86T7hhpRoh4NTopYIFJYKmA3KWuZD01C8kT83sgd9CePNhtASLVkxOkCRHENiDIfXNwzcq7hJdumAniRN
kDjNcmkWPg5DksoggA8Nq83B8avaTic5hA2e9g/L84VDjZQGOu+mM1QR1MerlYmxmMvw80AVqbFZOGAM0Wpqpnh0SbDMtTJXVM7Fkcm4Ihkg5
Chvqo3IKlUOpodr4o/8D+1IC7d6t0YcqCHTYUUK0/mmUvqfa5cUUQILC94dMHLSScPhkZG0kpdZ0CmZONr/G3yHXpi3zCnGv/myBlgE0ItGym
bmp/k/6Y2NsKU4U4mAIaDYAthTv+wg=</e:CipherValue>
                </e:CipherData>
            </e:EncryptedKey>
        </KeyInfo>
    </xenc:CipherData>

<xenc:CipherValue>xc8zbNou9JXvQjR61BQK0oHzHSn3yLz0DCQph4u/M9M4y+wx8RLWi+0230gutQWcwmU/MffwXrxJAXaCF3iAnCVT108
Sb10UwXLF9B4RmLt+asFYFwNlt7jikeBoetl0E341J6xwnLgg+zeC9ieTcbxFptCERlPQTEmVr8F45Ln/8ZDnvyPVCTZ5YoeDpLE9o0S6rvGz
/yV8VQ0yDyKLSpjDxZFT88+JdL8mf76cajjBKtGjKjVgxmCWCDFVJom24rFf1YpY1laCM350ccZTTh+kwU+m3TYlV53feASPLs2b61BuJzRvv
n01NmFYwuzhJ0qeHqReAvCDQqT7Z8G7nxm6X0ppog11LZ/4YPERLTCMKp1B2pGbDCZwL/XXJzHDWMU4GnWznAITvztFPQ0/TYywC18wEKenS
65bhd9C8TUJocyr++WYwfhcqrMSwVXeSp8HQWDZqWdXddGHZ06Iu9dtRat5X3VG0iez4sww1vru6oCnvvAsGLbadPDh1yOwkcQizr1HB7Fyo7
e53XiE9/5DATpA++I9ltZ1nrYl0i9lGQ//Y2K+ZEUdX4IHM04c14yCbODNMmc3KtWffwlvU0U808ADaLFnuv4N/gwsEldd0GUbSFKvkiqNTcm/
qE9WcZ4hTvK9F2IBhcnb91naq0CQJrtXf7grU5vrpEvTrvKa2zFB1jUT46qByMeNGS1StroFutiYLjh6jcPfkeTyBTXEq23szhda4gmVyb9X7
2MjFPvEb4fCGbQ4MhGoDNwEmm5qdUXEA4Qz0c8dfjQj/og1ds+QdlfdW3JOrToL3l16tBLCqzmZJrllB0jAQwi4ZnddS+c1R0E4FG+Dsft5yyH
IIXTXyEvCAjDzuqkbyK4GbrKIr56GbN0uR7ogWRLZ5cWAqivFa5tmUb/KN/aknYPRvox0EenxCry9W0c1jkIOLRZXxLUQnrv1fkdnPhv4P0Fu
wVDK0Jr1Df0Jr7Hwof6C6cdpzuDrT2td6gs4ZiMKHvWPGHlUaBAK3U4m4EQk7k8ZgBes13zDTP5G1z7+TBLP/80GWH1iRU1WrxsTU9pas0+8Y
HVVA5/h//uApMcsZ1xaA70dyuG0n43Wdp17bhHW20rVKU0uS4hcBLHXWRL2aP7Q1Pxe76v87W1Cr5d1eG5Lkoqdaa8wkY+giPX+N5ZKuXA34j
ETGG07h+6QtTjcsMjAnpYmUEIL5VrZSONKH2aN/5NA8ZMPHau8YqYACpXy/AOZdyH4peItycZOY9cvJm1TUWj0m68wIkds20C5/LwmjNRCMXE
ZXJUWrwVuwGnkZiZpQ4BWZFFG1iE+E8mw7YY5S0MBp0cr3mpT10iFMya9EKwghv82qa+ynFBDNF0K1zNuAQgBvB/nwceQHSbsvZuj2deeX0/
J1LQ1z0Cc5ajNPFdVfL0gx9Q47cilygomebYUKfVhB+kdhAT49+1aw6l0UqKB75LrJERRiK/gxI9Zqnjfl1iYvZkxRzgU8UMugoZU6XLpKsf
JRxoQ9z12BI0Cqi741xetVvWqU0skFxnGmZVW1CSmNwvYTa1aSDS3ci5I8PjKr8N1YegcXJlNcuLp1woUKUDhRkEUEh0/2xyMc9FuyOMT077Y
301XktMuDGVd/6WmNLRBcLqPMIIE54Jpn8VOyco3jw5c1fNHtKgC1Pim6mr4VNusOU4KXkuRvf7aoPeK9U6Jiv01+Vnc2/uBKMtS47H1Xt2
Nz5gew+zeUP9H+7yOpXyoZnfJv9gXxoLXJJkdhoimrtimVp1UVYnErp6MfT/IhNPouQ2+hqcs1C06m1f8+ZvP5a981IDj3nDqhDvzU23aXH8H
9oDcwnienBNAejOxupJm3gtP+XQDK8RKdLygKK5Kqm6UijjZ1kvv07em088ebGVnitteY4rHQZskRsC1WZAIPakPLxnLSc6+i6IYZDu/sQcZD
tcP5sdIhLg48Whur0fIIQrh3SkSTLGI5xsxD1WP83e5ljWjaEZeDQ7qEZfF20qE0FnaPTw014rmt2sKubw+mnd+ozuwoJ2I09xNIZXMbLX3hK
emN76542QmA9CLgIoB8WxGCMLanvd7NOT19Q4b0UqeyIo66MM/J3I71PdvoZwysxZU3j4i+VaeZ7+QSeqfh/5JkH1bdoRcmZ3Dn2Q1z+K4ksp
zIC0gp9DoJugZ1dGVTAUaxcj5ZrgDt9qkajwb2g+he1dEZjr5SZZ8jqevnwWY4wZgk84mvIXHRHG3kno9NUHNOZ9bX26FDRuw2k3kxEE5Pq1J
++J+yqQg/9o1rg1PVERZB1G9wpqxwe60bJcX4Io5apHpI5vyseENSffphxZZaXYiMaFuX0u5Rg2/qAkt5g+raR7V+1bvLfy9tEE1Pbvgo9RGq
zeY1sYKGZiJeohx2DDgk3XYLbNn3H0jXTkXZamPkzU7RUOYfeWENJZEwWxc+wPZnb9gqVpLwWj1ZziIngibrU/YRNYWEZV72vY8tfs4zDDpp
Fs0g5uqsk1pF1IqS5ZpUdu1U1+4t45Tk09vTBeJNQhVXXCxpVGJUkAxxrIU4x3RoLcPfVEudfSPx7qGT9hLWwdKxShgnwLS5HmkdDLpFFqj
1Nv3o4AmdnMj011jm5EfiLXSB1jQFJIMD100DApVl7Wi5owPmDGT2mLbhJrUM/G53h0iIrI62nT0MU+HA7Fxfjhw82Ym3zVJvV4WjLkRJi3zx
L411HKZY6ftwLWv+AbG80J/dKqkmM/Ch0NTZxgGuY5Jbsdcw1b0EcTQybIcnskd56kse9PBzZsm0j8B6EnRZbD0pU3N1D5+7QzR/1p935JVXT
gLqs2FtrPKE6i+KENP+kIhjJnv4PT84409jV4oAcZN/+SsPDeBfwS6nddjLYLkHoyprtQ3AsOTFXCjmk/t3e5qn0V52hkgr8gCtu3dCqX7QM
iKLLF5q1MCuVnpZHNwY1LMwcnF0++DF1XS3hVPShj30Q/dhyKrwy9C5dTIAZGi7jTOnC+4TVOES1BnJrJq+5jjNgbbg6nRCN1R61edbhIZ2X
2dB2Sj1gbITQD36NHKRNrGoUza0bZQorr4aJtMcG+feqhUOWLFy8Cu6PwwTyFVMY2dChBE0p1s4kt3HLLswPK6R8gDDf0ca8tYSJLRYsNgeto

```



```
Djg8ua2Ou6VpWP1Q4EHUKSGQ+ckjYDPA4LEtnKpMJR/YE9Mq+cNw4/42U0WchH0MQulV8I4FIjaKPAXUGJjJlWt+i4RWz+1qCDPJpgZGYkiFF
cz9r12AfojFvdMFrTk7q9oonbDH5uIf1fIJDGu5JjJLkaK7/YHibOGWuFxSs85HV4iKRXXQ9ZotnmiOj5pTiyHNC5+D9978UVnNXAAC1jlrVp
x3Ftz1COFdqta13MnvVTZ+VegSA2mJPrSfIDRaEDEApnoBYdNTRUgwdtXB7POH4gTiHKGBPrf6Ku5U8mf5XByUNiKhW4sj8P1WZex63yVHI1x
b1kTbF3AN3NYn/mpughyVia4wL0MOJ/7DemE/IRQ4OK5UWxoupkqMr3R2tsB4CthHW2yJ06ZQaFcXJ/N80PbhAGRwhv8WCXVvMPK9oPhbMYi
viZuXvm9bVLVq5Ucv2JdN808T67X5aBjTqoHGNU5bkqlf9USgpB1htGgyCjbz6Uje2RCQJITlKtC+7gdhTGTNETzwOIffp8F/zrqEPKV48M6E
up30SNYL3kl+mgpMg1QYX5mHwvWHReyCzYJXeQv1pZfLY1s+mWnmFs4eeGv0qRW57D3UfpI2AYNbMmyDG3c+UW5APYgt71s7FbbLj/b30WWQI
dtrKBXgMx1S02LjEpWC4mUNvi6mBpLgIiVTTnn2Z5hy8xnLNYtdaSVV/U0OCZ0AR+6M667Cpfp1W/QTX2f+joKVMcYcCX93zpqY+uGEUi6TWt
MprLndxEDOJbnEMztVJ4rEdadP40JtQQp6nqXdnXcqWU5tTOKAm6PoJQ5NCjGuIE3P2SWgrl6jfH/Edqw4Zg8oUgqSerWFS0284g2vcjkGBR1
PKK18BIDj6w09GHseWKCgpscBiZISP1tb00U7C0Tra56hPzKrBUvbCYfFLixvRHU0eVw1XXLVg6ndILDyB2hy/0CQC0G1YkBWgIzQ2v17Dnqt
j/zSWqgzvAsx8mzZRYp5UunTod8Da0aPnUHnUm6ax4cDrSa73Wbz/XE7CIjf8zhKgDgdXDq9/mNHDj4zuLLAKMR7UjpmPpbsCq67ouJg+edd
n9n+GNTiuFRmB1nLW/nt+ONHWWWHtkhJi8AuDuvwpF6rc4FGTPFg+w3dYMDIROt1aFPn1J4exNbyfgXNJCEURQdV+gZQ+89Gg+duy7nJ6EWh0
N9yU/4+kKrjoEhZl1pIs4Fhi+WqTkP60u84wK+f0IH2nLYjO6aW8V2o7ZEXKbrt72ZRP03fo7Bm69cpOprP3xnooE6T7FdnYIj6fWgGYTKG+h
5b6edwqOJ2md99IG5w71HcnM11INFLUHg5RPMQJ0CfuUoV3KhPWZr5LyuAbPph4tkc0/SrArFaZ4yfdFR2DkGg8X1fNdc1o3Zodqn0LrIa5Ab
6GMCpxJ0v9tIBX00QoPPfWUj1r4Gv1XTNjh0z5Jj9LYcvlnBv2r30FzRNVZzC9qAJ3N00yu1svRP++AMo950a5Im/24pPu2MJsRii0VATCFmg
KRfTJk3h40YippyEgod9bhdWY4Zqt4Rntw123S1RXEvYLK9JrdEiw9BaiJd1lpy5GwcAde7gEGWmgSNRMcV4pggz3Px+NV0TTVm/tt0Ta9p7
9QH5b/JzRaNqJ519HFFn4YtqC8ujhrHEk8HsdzBPxdd0Uqr5K5CMbICmN83yd4RNPmsc3PyPNxoetjVihZv1uVYbbRtVF0zX08BKWBeBJ60qj
z1cg859UPqUdSmUcuy0EuVR8BCIUgrVGbeux/ZVSjI/Z/9k/+ksoNb7L2XGn0+EKDNYPNAL3fYXB//Zx30a27wH01muyFif1/CUqu6+o0jvD
VwpmDT7IkVwrJ3r0UKSUCwhECvLg0T4jt2v9pz+XL1bsqm5FjJ02bjJ480jL</xenc:CipherValue>
</xenc:CipherData>
</xenc:EncryptedData>
</trust:RequestedSecurityToken>
<trust:RequestedAttachedReference>
  <o:SecurityTokenReference xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
    <o:KeyIdentifier ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
profile-1.0#SAMLAssertionID">_b3eff871-f427-4715-b69e-20176cbd4342</o:KeyIdentifier>
  </o:SecurityTokenReference>
</trust:RequestedAttachedReference>
<trust:RequestedUnattachedReference>
  <o:SecurityTokenReference xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
    <o:KeyIdentifier ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
profile-1.0#SAMLAssertionID">_b3eff871-f427-4715-b69e-20176cbd4342</o:KeyIdentifier>
  </o:SecurityTokenReference>
</trust:RequestedUnattachedReference>
<trust:TokenType>urn:oasis:names:tc:SAML:1.0:assertion</trust:TokenType>
<trust:RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue</trust:RequestType>
<trust:KeyType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Bearer</trust:KeyType>
</trust:RequestSecurityTokenResponse>
</trust:RequestSecurityTokenResponseCollection>
</s:Body>
</s:Envelope>
```

Преобразование содержимого маркера в BASE64URL

Содержимое маркера расположено в узле `RequestedSecurityToken`, если кодировка ответа UTF-8, то содержимое в BASE64URL примет вид (переносы строк добавлены для удобства чтения):

PPh1bmM6RW5jcnlwdGVKRGF0YSBUeXB1PSJodHRWoi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGVuYyNFbGVtZW50IiB4bWxuczp4ZW5jPSJodHRWoi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGVuYyMiPjx4ZW5jOkVuY3J5cHRpb25NZXRob2QgQWxb3JpdGhtPSJodHRWoi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGVuYyNhZXMyNTYtY2JjIiAvPjxLZX1JbmVzIHhtbG5zPSJodHRWoi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjIj48ZTpFbmNyeXB0ZWRLZXkgeG1sbmM6ZT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjIj48ZTpFbmNyeXB0aw9uTWV0ag9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjcjhNlw9hZXAtbWdmMXAiPjxEawdlc3RNZXRob2QgQWxb3JpdGhtPSJodHRWoi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjcz2hhMSIgIz48L2U6RW5jcnlwdGlvbkl1dGhvZD48S2V5SW5mbz48bzPTZWNIcm10eVRva2VuUmVmZX1JbmNlIHhtbG5zOm89Imh0dHA6Ly9kb2NzLm9hc21zLW9wZW4ub3JnL3dzcy8yMDA0LzAxL29hc21zLTlTWMDQwMS13c3Mt d3NzZWNIcm10eS1zZWNIeHQtMS4wLnhzZCI-PfG1MD1EYXRhPjxYNTA5SXNzdWVyU2VyaWFsPjxYNTA5SXNzdWVyTmFtZT5DTj1TVFM8L1g1MD1Jc3N1ZXJOYW11PjxYNTA5U2VyaWFsTnVtYmVyPyJM0ODA3MjYzNDQ1NTYwMTMwOTI2NyZ3ODYwNjg3NTk3MDkzNDg2PCYNTA5U2VyaWFsTnVtYmVyPjpwWDUwOUlzc3Vlc1Nlcm1hbD48L1g1MD1EYXRhPiwbzbPTZWNIcm10eVRva2VuUmVmZX1JbmNlPjw

vS2V5SW5mbz48ZTPDaXBoZXJEYXRhPjx10kNpcGh1c1ZhbHVlPk9sQnFzUzJITm
c1UC84NlQ3aGhwUm9oNE5Ub3BZSUZKWUttM0FLV3VaRE8xQzhrVDgzc2dkOUNlU
E5odEFTTHZreE9rQ1JIRW5pRElmeWE53emNxN2hKZHVtQW5pUk5rRGP0Y21rV3BH
NURrc29nZ0E4TnE4M0I4YXZhVG1jNWhBMmU5Zy9MODRWGRGaUUDPdSttTFRUjF
NZXJswW14bU12dzgWQVZxYkZaT0dBtTBXcHFwaG4wU2JETXRUSlhWTTdGa2NtNE
1oa2c1Q2h2cW8zSutsVU9wb2RyNG8vOEQRmULDN2Q2dDBZY3FDSFR5VVVLMC9tb
VV2cWZhNWNVVFJTEM5NGRNSExTc0NQAgtArZBrCGRab0NtWk90ci9HM3lIWHBp
M3pDbkd2L215QmxnRTBJdEd5bWJtcC9rLzZzMk5zS1U0VTRtQUlhRf1BdGhUdnc
rZz09PC9l0kNpcGh1c1ZhbHVlPjwvZTPDaXBoZXJEYXRhPjwvZTPFbmNyeXB0ZW
RLZXk-PC9LZXlJbmZvPjx4ZW5jOkNpcGh1ckRhdGE-PHh1bmM6Q2lwaGVyVmFsd
WU-eGM4emJOb3U5S1h2UWpSNjFCUUtPb0h6SFNum3lMek9EQ1FwaDR1L005TTR5
K3d40FJMV2krTzIzT2d1dFFXY3dtVS9NZmZXWHJ4Skf4YUNGm2lBbkNWVDEwOFN
iMTBVd1hMRjlCNFJtThQRyXNGWUZ3Tmx0N2ppa2VCb2V0bDBFMzQxSjZ4d25MZ2
cremVDOWl1VGNieEzwdENFUmXQUVRfVbVzY0EY0NUxulZhaRG52eVBWQ1RaNVlvZ
URwTEU5bzBTNnJ2R3oveVY4VlEweUR5a0xTcGpEeHpGVDg4K0pkTDhtZjc2Y2Fq
akJLdGdLalZneG1DV0NER0ZWsm9NMjRyRmYxWBZbGxhQ00zNU9jY1pUVGgra1d
VK20zVf1sdlMzZmVBU1BMczJiNjFCdUp6UnZ2bk8xTm1GWXd1emhKT3F1SHFSZU
F2Q0RRcVR2N1o4RzdueG02eE9wcW9nbDFswi80WVBFUkxUQ01LcGxCMnBHYKRdW
ndML1hYSnpIRfDNVTRHb1d6bkfJVHZ6dGZUTAvVF15d0MxOHdFS2VuUzY1YmhE
OUM4VFVKb3ljcisrV3lXZmhjcXJNc1dWwGVTcDhIUvdEwnFXRHhkZEdIwK82SXU
5ZHRsYXQ1WdNWRR09pZXo0c3d3MXZydTzVQ252dkFzR0xiYWRQRGgxeU93a2NRaX
pyMUhCN0Z5bzd1NTNYaUU5LzVEQVRwQSSrSTlSdFoxbnJZbE9pOWxHUS8vWTJLK
1pFVURYNElITU08YzE0eUniT0ROTWMzS3RXZmZ3bHZVT1U4MDhBRGFMRm51djRO
L2d3c0VsZGQWR1ViU0ZLdmtpcU5UY20vcUU5V2NaNGhUdms5RjJJQmhhbmI5MW5
hcTBDUUpydFhmN2dyVTV2cnBFd1RSdkthMnpGQmxqVQ0NnFCeU11TkdtMVNUcm
9GdXRpWUxqaDZqY1Bma2VUeWJUUEVxMjNzemhkyTRnbVZ5Yj1YNzJNakZQdmViN
GZDR2JRNE1oR29ETndFbW01cWRVWEVBNFF6MGM4ZGZqUWkvb2cxZHMURWrsZmRX
M0pPc1RvTDNsbDZ0QkxDcW16SnJsTGJPakFRd2k0Wm5kZFMrY2xST0U0RkcrRHN
mdDV5eUhJSXhUWf1FdKnbAkR6dXFrY1lrNEdicktJcjU2R2JOMHVS29nV1JMWj
Vjv0FxaXZGYTV0bVViL0tOL2Frbl1QUUnZveE9FZW54Q1JZOVdPYzFqa01PTFJaw
HhMVVFucnzZsZmtkb1BodjRQT0Z1d1ZESzBKcmxERm9KcjdId29GNkM2Y2RwenVE
UnRYMnRkNmdzNFppTUtIdldQR0hsdUJBa1gzVW00RVFRn2s4WmdCZXNsM3pEVHA
1RzF6NytUQkxQLzhPR1dIbG1SVTFXcnhzVFU5cGFzTys4WUhwVke1L2gvL3VBCE
1jc1pseGFBnzBkeXVHT240M1dkcGw3YmhIVzIwc1ZLVXUwcZRoY0JMSFhXckwyY
VA3UTFQWGU3NnY4N1cxQ3I1ZDF1RzVMA29xZGFhOHdrWStnaVBYK041Wkt1WEEz
NGpFVEdHTzdoKzZRDHRKY3NNakFucF1NdWVjTDVWclpTb05LSDJhTi81TkE4Wk1
QSGF10F1xWUFDcFh5L0FpWmR5SDRWZUL0eWNaT1k5Y3ZKbTFUVVdqT2020HdJa2
RzMjBDNS9Md21qTlJjTVhFWlhKVVDyd1Z1d0dua1ppWnBRNEJXWmZGRzFpRStFO
G13N11ZNV1zME1CcDBjCjNtcFQxT2lGTX1h0UvLd2d2aDgyCUEreW5GQkRORjBL
bHpOdUFRZ0J2Qi9ud2N1UUhZYNn2WnVqMmR1ZVgwL0oxTFFswNJPQ2M1YwpOUEZ
kVkZsMGd40VE0N2NpbH1nb21lY1lVS2ZWSGIra2RIQVQ00StsYXc2bDBVCUtCNz
VMckpFUlJpSy9neEk5WnFumZMaTFZdlpreFJ6Z1U4VU11Z29aVTZYTHBLU2ZKU
nhRbz16bDJCSTBDcWk3NDF4ZXRWdldxVTBza0Z4bkdtWlZXbENTbU5Xdl1UYTFh
U0RTM2NpNUk4UGpLcjhObF1lZ2NYSmxOY3VMcDF3b1VLVURoUmtFVUVoTy8yeHl
NYz1GdX1PTVQWnZdZM08xWgt0TXVER1ZELzZXbU5Mc1JCY0xxUE1JSVNFNTRKcG
44Vk95Y28zanc1Y2xmTkh0S2dDbFBpbTZtcjRWtnVzT1U0S1hrdVJ2Zjdhb1B1S
z1VNkpJdk9sK1ZuYzIvdUJLTXRTNDdIbFh0Mk56Nwd1Vyt6ZVVQOUgrN3lPcFhZ
b1puZkp20WdYeG9MWEpKa2Rob2ltnRpbVZWVWVWw5FcnA2TWZUL01oT1BvdVE
yK2hxY3NsQzA2bWxmOCTaVnA1YTk4bElEajNuRHFoRHZ6VTIzYVhIOEg5b0Rjd0
5pZW5CTkFlak94dXBKbTNDfAREFFESzhSS2RMewdLSzVLCw02VWlqalpSa3Z2b
zd1bTA40GVIR1ZuaXR0ZVkoCkhrWnNrUnNDbFdaQUlQQWtQTHhuTFNjNitpNk1Z
Wkr1L3NRY1pEdGNQXNkSWhMZzQ4V2h1cjbMSU1RcmgzU2tTVExHSTV4c3hEbFd
QODN1NWxqV2phRvp1RFE3cUvArMYMHFFMEZuYVBUD08xNHJtdDJzS3Vidyttbk
Qrb3p1d29KMkkwOXhOSvPYTWJMWdNoS2VtTjc2NTQyUW1BOUNMZ0lvQjhXeEDT
UxhbnZkN05PVGw5UTRiT1VxZXlJbzY2TU0vSjNjNzFQZHvWnd5c3haVTNqNGkr
VmF1WjcrUVN1cWZoLzVKA0gxYmRvUmNtWjNEbjJRbHorSzRrc3B6SUMwZ3A5RG9
KdWdaMRHV1RhVUF4Y2o1WnJnRHQ5cWthandiMmcraGUxEVaanI1U1paOGpxZX
Zud1dZNHdaZ2tCNG12SVhIUkhHM2tubz1OVUhOb1o5Y1lgyNkZEUnV3Mmsza3hFR
TVQcWxKKytKK3lXUwcvOW8xcmcxUFZFUlpCMUc5d3B4cXd1Nm9CakN4NElvNWFW
SHBJNXZ5c2VFtnNmZnBoeFpaYVhZaU1hRnh1T3g1Umcyl3Fba3Q1ZytyYVI3Vis
xYnZMZ1k5dEVFbFBidmdvOVJHcXp1WTFzWUtHWk1qZU9oeDJERGdrM1hZTGJObj
NIT2pyVGtYwmFtUgt6VTdSVU9ZmVXRu5qWkV3d3hjK3dQWm5iYj1ncVZwTHd3S
jFaemlJbmdpYnJVL1lST1lXRvpWNzJ2WTh0Z1M0ekRECHBGczBnNXVxc2sxcEZs
SXFTNVpVXVEawxVMSs0dDQ1VGswOXZUQmVKt1FIaHZYwEN4cFZHS1VrbkF4ck1
VNHzgUm9sQ3BmVkv1ZGZTUHG3cUdUOWhMV3dkS3hTaGduV3NMNUhta2RETHBGRn
FchE52M280QW1kbb1KMgcYcm01Bw7nhEhTQ1Fc1l117KCU1FhDRDRFEwdkYn1d0N

1 q0L2zh200qW1K0K1N0X3am01NwZp0111qJ1 q00ZK0J1L00DF KET w0KXpH1UpH
W9XUG1ER1QybUxiaEpyVU0vRzUzaDBpSXJJNjJuVDBNVStIQtdGeGpodzgyWU0z
e1ZKd1Y0V2pMa1JKSTN6eEw0bDFIS1pZNmZ0d0xXVItBYkc4T0ovZEtxa21NL0N
oME5UWnhnR1V5NUpic2RjdzFiMEVjVFF5YkljbnNrZFM2a3N1OVBCelpzBU9qOE
I2RW5SekJkMHBMV04xRDUrN1F6Ui8xcDkzNUPwWFRnTHFzMKZ0c1BLRTZpK0tFT
lAra0loampOdjRQVDg0NE85a1Y0b0FjWk4vK1NzUERlQmZ3UzZuZGRqTF1Ma0hv
eXBydFEzQXNPVEZYQ2ptay90M2U1cw4wVjUyaGtnenI4Z0N0dTNkQ1hxN1FNaUt
MTEY1cTFNQ3VWbnBaSG5XWwMTXdjkbZ4MCsrREYxwFMzaFZQc0hqaJmWUS9kaH
lLcnd5OUM12FRJQVpHaTdQvE9uQys0VFZPRVMxQm5KckpxKzVqak5nYmc2b1JDT
mxSNjFlZGJISVoyWDJkQjJTajFnYk1UUUQzNk5IS1J0ckdvVXpBMGJaUW9ycjRh
S1RtY0crZmVxaFVPV0xGeThDdTzQd3dUeUZWTXkyZENOQkUwcDFzNGt0M0hMTHN
3UGs2UjhnRERmMGNhOHRZU0pMU11TTmd1dG9Eamc4dWEyT3U2VnB3UDFRNEVIVW
tTR1ErY2tqWURQQTRMRXRuTktwTUpSL11FOU1xK2N0dzQvNDJVMFdjaEgwTVF1b
HY4STRGSwphS1BBWFVnSmpsdVdmaTRSV3orMXFDRFBKcGdaR1lraUZGY3o5cjEy
QUZvakZ2ZE1Gc1RrN3E5b29uYkRINXVJZjFmSUpER3U1SmpKTGthSzcvWUhpYk9
HV3VGeFNzODVIVjRpS1JYWFE5Wm90bm1pT2o1cFRpeUhoQzUrRDK5NzhVVM50WE
FBQ2xqbHJWcHgzRnR6MUNPRmRxdGExM01ud1ZUWitwZwdTQTJtS1ByU2ZJRFJhR
URFQXBub0JZZE5UU1Vnd2R0WEI3UE9INGdUaUhLR0JQcmY2S3U1VThtZjVYQn1V
Tm1LaH0c2o4UDFXWmV4NjN5VkhJMXhiMwtUYkYzQU4zT1luL21wdWdoeVZpYTR
3TDBNT0ovN0R1bUUVbFJRNE9LNVVXWG91cGtxTXIzUjJ0c0I0Q3RoSFcyeUowN1
pRYUZjWEovTjgwUGJoQUdSd2h2OFdDWFZ5dk1Qaz1vUghiTV1pdm1adVh2bTliw
UxwcTVVY3YySmROOE84VDY3WdVhQkpUcw9IR05VNWJrcWxmOVVTZ3BCMWh0R2d5
Q2piejZVamUyUkNRSk1UbGtUQys3Z2RoVEduTKVUendPSWZmcDhGL3pycUVQS1Y
00E02RXVwMzBTT11MM2tsk21ncE1nMVfZWdVtSFdWd0hSZX1De1lKWGVRdjFwWk
ZsWTFzK21XTm1GczR1ZUd2MHFSVzU3RDNVZnBjMkFZTmJNbX1ERzNjK1VXNUFQW
Wd0NzfZn0ZiYkxql2iZt1dXUu1kdHJLQ1hnTXgxUzAyTgPfcFdDNG1VTnZpNm1C
cExnSW1WVFRubjJaNWh50HuTE5ZVGRhU1ZWL1UwT0NaMEFSKzZNNjY3Q3BmcDF
XL1FUWDJmK2pvs1ZNY1ljQ1g5M3pwcVkrduDFVWk2VFd0TXBSbG5keEVET0pibk
VnenRwsjRyRWRhZFA0MEp0UVFwNm5xwGRuWGNxV1U1dFRPS0ftN1Bva1E1TkNqR
3VJRTNQm1NXZ3JsNmpmac9FZHF3NFpnOG9VZ3FTZXJXRlMwMjg0ZzJ2Q2prR0Jy
M1BLSzE4QklEajZ3MD1HSHN1V0tDR3BzY0JpWk1TUDF0Yk9PVTdDMFRyQTU2aFB
6S3JCVXZiQ1lMkxpeHZSSFUwZVZ3MVhYTFZnNm5kSUXeEUIyaHkxVMENRQ09HmV
lrQldnSXpRMnYxN0RucXRqL3pTV3FnenZBc3g4bXpaUnlQNVV1b1RvZDhEYTBhU
G5VSG5VbTzHeDRjRHJzYTczV2J6L1hFN0NJamY4emhLZ0RnZfHEctkvBU5IRGo0
enVsTExB0S01SN1VqGc1QcGJzQ3E2N291SmcrZWRkbjluK0dOVG11R1JtQjFuTFc
vbnQrT05Id1dXSHRraEppOEf1RHV2d3BGNnJjNEZHVFBGZyt3M2RZTWRJUK9UMW
FGUG4xSjR1eE5iewZnWE5KQ0VVU1FkVitinW1ErOD1HZytkdXk3bko2RVdoT045e
VUVNctrS3Jqb0VowmxscE1zNEZoaStXcVRrUDYwdTg0d0srZjBJSDJuTF1qTzZh
VzhWmM83Wkv4S2JydDcyWl1JQTzNmbzdCbTY5Y3BPcHJQM3hub29FNlQ3RmRueU1
qNmZXZ0dZVEtHK2g1YjZlZHdxT0oybWQ5OU1HNXc3MUhjbk1sMU10RkxVaGc1U1
BNUUowQ2Z1VW9WM0toUFdaccjVMeXVBY1BwaDR0a2MwL1NyQXJGYVo0eWZKR1IyR
GtHZzhYMWZOZGMxbzNab2RxbjBMck1hNUFiNkdNQ3B4SjB20XRJQlhpT1FvUFBm
V1VqMXI0R3ZSfWROamgwejVKajlMeWN2bG5CdJjYmZBGe1J0dlp6Qz1xQUozTjB
PeXUxc3ZSUCsrQU1v0TVPYTVJbS8yNHBQdTJNSnNSaWkwVkfUQ0ZtZ0tSZ1Rka2
ozaDQwwWlwcH1FZ29kOWJoZFdZNFpxdDRSbnR3MTIzUzFSWEV2eUxLOUpyZEVpd
z1CYWlKZGx2cHk1R3djQWR1N2dFR1dtZ1NOUk1jVjRxcGd6M1B4K05WMFRUVm0v
dHQwVGE5cDc5UUG1Yi9Ke1JhTnFKNTE5SEZGbjRZdHFD0HVqaHJIRws4SHNkekJ
QeGRkMFVxcjVLNUNNYm1DTW44M31kNFJ0cG1zYzNQeVB0eG91dGpWawhadjF1V1
liY1J0VkyWe1hPOEJLV0J1Qko2MHFqemxjZzg10VVQcVvku21VY3V5MEV1V1I4Q
kNJVWdyVkdixXV4L1pWU2pJL1ovOWsvKy9rc290YjdMM1hHbjArRutET11QbkFM
M2ZZWEIvL1p4M09hmjd3SDAxbXV5Rm1MS9DVXF1NitmGp2RFZ3cG1EVDdJa1Z
XckozcjbVa1NVY3doRUN2TgcwVDRqdDJ20XB6K1hMMWJzcw01RmpKMDJiako00D
BqTDwveGVuYzpDaXBoZXJWYwx1ZT48L3hlbmM6Q2lwaGVyRGF0YT48L3hlbmM6R
W5jcnlwdGVkRGF0YT4

Обмен маркера доступа ADFS на маркер доступа ЦИ DSS

Завершающим шагом данного сценария является обмен маркера SAML на JWT маркер ЦИ DSS с помощью следующего запроса:

```
POST /STS/oauth/token HTTP/1.1
Host: simdss.cryptopro.ru
Content-Type: application/x-www-form-urlencoded
cache-control: no-cache

grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Atoken-exchange
&client_id=12cf4adb-ba3e-40d0-a5d9-fa507ef26932
&resource=urn%3Acryptopro%3Adss%3ASignserver%3ASignServer
&subject_token=PHh1bmM6R
&subject_token_type=urn%3Aietf%3Aparams%3Aoauth%3Atoken-type%saml1
```

Здесь в значении subject_token передаётся BASE64URL представление маркера, полученное на предыдущем шаге (в примере запроса оно сокращено).

Пример 4. WS-Trust 1.3: маркер доступа JWT

JWT маркер может быть получен по протоколу WS-Trust. Процесс авторизации состоит из следующих шагов.

1. Формирование запроса на авторизацию в ADFS.
2. Получение маркера доступа на ADFS.
3. Преобразование содержимого маркера из BASE64.
4. Обмен маркера доступа ADFS на маркер доступа ЦИ DSS.

Формирование запроса на авторизацию

Запрос в рамках WS-Trust протокола имеет следующий вид:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:a="http://www.w3.org/2005/08/addressing"
xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <s:Header>
    <a:Action s:mustUnderstand="1">http://docs.oasis-open.org/ws-sx/ws-trust/200512/RST/Issue</a:Action>
    <a:MessageID>urn:uuid:ef4d294e-b49a-44ad-8ad5-e15fe22389db</a:MessageID>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To s:mustUnderstand="1">https://simadfs.cryptopro.ru/adfs/services/trust/13/usernamemixed</a:To>
    <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
      <u:Timestamp u:Id="_0">
        <u:Created>2019-02-11T07:07:09.031Z</u:Created>
        <u:Expires>2019-02-11T07:12:09.031Z</u:Expires>
      </u:Timestamp>
      <o:UsernameToken u:Id="uuid-41f7fc65-1d07-4dc8-b72c-0b3a76fa9b74-2">
        <o:Username>test@tdss.local</o:Username>
        <o:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">1qaz@WSX</o:Password>
      </o:UsernameToken>
    </o:Security>
  </s:Header>
  <s:Body>
    <trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
      <wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
        <wsa:EndpointReference xmlns:wsa="http://www.w3.org/2005/08/addressing">
          <wsa:Address>urn:cryptopro:dss:samples</wsa:Address>
        </wsa:EndpointReference>
      </wsp:AppliesTo>
      <trust:KeyType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Bearer</trust:KeyType>
      <trust:RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue</trust:RequestType>
      <trust:TokenType>urn:ietf:params:oauth:token-type:jwt</trust:TokenType>
    </trust:RequestSecurityToken>
  </s:Body>
</s:Envelope>
```

Получение маркера доступа

В ответ сервис вернёт следующие данные:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:a="http://www.w3.org/2005/08/addressing"
xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <s:Header>
    <a:Action s:mustUnderstand="1">http://docs.oasis-open.org/ws-sx/ws-
trust/200512/RSTRC/IssueFinal</a:Action>
    <a:RelatesTo>urn:uuid:ef4d294e-b49a-44ad-8ad5-e15fe22389db</a:RelatesTo>
    <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
      <u:Timestamp u:Id="_0">
        <u:Created>2019-02-11T07:07:52.740Z</u:Created>
        <u:Expires>2019-02-11T07:12:52.740Z</u:Expires>
      </u:Timestamp>
    </o:Security>
  </s:Header>
  <s:Body>
    <trust:RequestSecurityTokenResponseCollection xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-
trust/200512">
      <trust:RequestSecurityTokenResponse>
        <trust:Lifetime>
          <wsu:Created xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">2019-02-11T07:07:52.693Z</wsu:Created>
          <wsu:Expires xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">2019-02-11T08:07:52.693Z</wsu:Expires>
        </trust:Lifetime>
        <wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
          <wsa:EndpointReference xmlns:wsa="http://www.w3.org/2005/08/addressing">
            <wsa:Address>urn:cryptopro:dss:samples</wsa:Address>
          </wsa:EndpointReference>
        </wsp:AppliesTo>
        <trust:RequestedSecurityToken>
          <wsse:BinarySecurityToken wsu:Id="_081a18a3-68d0-49e9-9515-c46d1bd48c50-
0465E3D60128657379D25AB0189F70B8" ValueType="urn:ietf:params:oauth:token-type:jwt"
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">ZXlKMGVYQWlPaUpLVjFRaUxDSmhiR2NpT2lKU1V6STFOaUlzSW5nMWRDSTZJamhPYmxWVVF5OTFRbFZ6Y1hwbVNuSlVNMWcyTTA1
M2JVaEZPQ0o5LmV5SmhkV1FpT2lKMWNtNDZZM0o1Y0hSdmNISnZPbVJ6Y3pwellXMXdiR1Z6SWl3aWFTYnpJam9pYUhsMGNEb3ZMM05wYldGa
1puTXVZM0o1Y0hSdmNISnZMbkoXTDlGa1puTXZjMlZ5ZG1salpYTXZkSEoxYzNRaUxDSnBZWFFpT2pFMU5EazROamc0TnpJc0ltVjRjQ0k2TV
RVME9UZzNNa1EzTWl3aVlVYjBhRzFsZEdodlPpSTZJbWgwZEHBNkx5OXPZMmhsYldGekxtMXBZM0p2YzI5bWRDNWpimjB2ZDNndk1qQXdPQzh
3Tmk5cFpHVnVkrR2wwZVM5aGRYUm9aVzUwYVdOaGRHbHJibTFsZEdodlPpDOXDZWE56ZDI5eVpDSXNJbUyxZEdoZmRHbHRAU0k2SWpjd01Ua3RN
REl0TVRGVU1EYzZNRGM2T1RFdU56Y3hXaUlzSW5abGNpSTZJakV1TUNKOS5hbGdsNmN6Skc3U1dTty0yb3NuElJEQm5VOVFfWVh4c0t6bjRnY
3VYdXFsV2FrRVUxUEhOTWJUSXdhRTlPb1hNU014T3hCdEl0Tud6am1US2R1N3N5YmppbXppQV9xZmdZYmw2c1NFdTZUM3plQk1nczJfUV8wUl
poRUZvV1BGt3JNzjhIQ1l1Ja3RlQk10SHlmdNDJyS3lVdmxVOGdQWY4U0J5eTVvN2NEV1A2eTVNLXlowWpBanBUX19yVEZTN0otU3NzX1ZBQWx
sM01LUjZ0QXBVWVI1TTB1Njh1ZEdxc3g0VmR1TFNtaHl0V0JDNUIXQTlTWkg3QWJ0bVBLWTU1bkFCNFVWwNFBWjZrN2xjV1Zjbm54bms5anA0
YjlkX2hZdm15N1Z2LU42U1pVeGE4ckN0cERXVG1RZWSYb01GVmQ3Ymp4Q0dnWUY2NDMzaHh4TlhwSWnc=</wsse:BinarySecurityToken>
        </trust:RequestedSecurityToken>
        <trust:TokenType>urn:ietf:params:oauth:token-type:jwt</trust:TokenType>
        <trust:RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue</trust:RequestType>
        <trust:KeyType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Bearer</trust:KeyType>
      </trust:RequestSecurityTokenResponse>
    </trust:RequestSecurityTokenResponseCollection>
  </s:Body>
</s:Envelope>
```

В узле `BinarySecurityToken` сохранился JWT маркер закодированный в BASE64.

Преобразование содержимого маркера из BASE64

Для получения JWT-маркера доступа необходимо преобразовать содержимое узла `BinarySecurityToken` из BASE64 и рассмотреть полученный массив байтов как строку в кодировке UTF-8.

Строка (переносыстрок добавлены для удобства чтения):

```
ZX1KMGVYQWlPaUpLVjFRaUXDSmhiR2NpT2lKU1V6STFOaUlzSW5nMWRDSTZJamhP
YmxwVVFsoTFRbFZ6Y1hwbVNuslVNMWcyTTA1M2JVaEZPQ0o5LmV5SmhkV1FpT2lK
MWNtNDZZM0o1Y0hSdmNISnZPbVJ6Y3pwe1lXMXdiR1Z6SWl3aWfYTnpJam9pYUHS
MGNEb3ZMM05wYldGa1puTXVZM0o1Y0hSdmNISnZMbkoXTDJGa1puTXZjMlZ5ZG1s
a1pYTXZkSEoxYzNRaUXDSnBZWFFpT2pFMU5EazROamc0TnpJc0ltVjRjQ0k2TVRV
ME9UzZnNalEzTWl3aVlYVjBhRzFsZEdodlpDSTZJbWgwZEhBNkx5OXpZMmhsYldG
ekxtMXBZM0p2YzI5bWRDNWpiMjB2ZDNndk1qQXdPQzh3Tmk5cFpHVnVkr2wwZVM5
aGRYUm9aVzUwYVd0aGRHbHZibTFsZEdodlpDOXdZWE56ZDI5eVpDSXNjbUYxZEdo
ZmRHbHRaU0k2SWpJd01Ua3RNREl0TVRGVU1EYzZNRGM2TlRFdU56Y3hXaUlzSW5a
bGNpSTZJakv1TUNKOS5hbGdsNmN6Skc3U1dTty0yb3NUelJEQm5VOVFfWVh4c0t6
bjRnY3VYdXFvS2FrRVUxUEhOTWJUSXdhRTlPb1hNU014T3hDcEl0TUd6am1US2R1
N3N5YmppbXppQV9xZmdZYmw2c1NFdTUM3plQk1nczJfUV8wU1poRUZvV1BGt3JN
ZjhIQ1lJa3R1QkJOShlmdNDjYs3lVdmxVOGdQWtY4U0J5eTVvN2NEVlA2eTVNLXlo
WwPbanBUX19yVEZTN0otU3NzX1ZBQWxsM01LUjZ0QXBVWI1TTB1Njh1ZEdxc3g0
VmR1TFNtaHloV0JDNUlXQTlTWkg3QWJOvBLWTU1bkFCNFVWnFBWjZrN2xjV1Zj
bm54bms5anA0YjlkX2hZdm1SN1Z2LU42U1pVeGE4ckN0cERXVG1RZw5Yb01GVmQ3
Ymp4Q0Q0dnWUY2NDMzaHg4TlhSWnc=
```

будет соответствовать следующему JWT-маркеру:

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6IjhbVUQ191Q1VzcXpmS
nJUM1g2M053bUhFOCJ9.eyJhdWQiOiJ1cm46Y3J5cHRvcHJvOmRzc2pzYW1wbGVzI
iwiaXNzIjoiaHR0cDovL3NpbWFKZnMuY3J5cHRvcHJvLnJ1L2FkZnMvc2Vydm1jZX
MvdHJ1c3QiLCJpcyYXQiOiJlNDk4Njg4ZnIsImV4cCI6MTU0OTg3MjQ3MiwiYXV0aG1
ldGhvZCI6Imh0dHA6Ly9zY2hlbWZlLm1pY3Jvc29mdC5jb20vd3MvMjAwOC8wNi9p
ZGVudG10eS9hdXRoZW50aWNhdGlvbm1ldGhvZC9wYXNzd29yZCI6ImF1dGhfZGltZ
SI6IjIwMTktMDItMTFUMDc6MDc6NTEuNzcxWiIsInZlciI6IjEuMCJ9.alg16czJG
7SWSO-2osTzRDBnU9Q_YXxsKzn4gcuXuqoKakEU1PHNMbTIwaE90oXMSMxOxCpINM
GzjiTKdu7sybjimziA_qfgYbl6rSEu6T3zeBMgs2_Q_0RZheFowPForMf8HBYIkte
BBtHyf42rKyUv1U8gPY68SByy5o7cDVP6y5M-yhYjAjpT__rTFS7J-Sss_VAA1l3I
KR6tApUUb5M0e68udGqsx4VduLSmhyhWBC5IWA9SZH7AbNmPKY55nAB4UpZqAZ6k7
1cWVcnnxnk9jp4b9d_hYviR7Vv-N6SZUxa8rCNPdWTiQenXoMFVd7bjxCGgYF6433
hx8NXRZw
```

Обмен маркера доступа ADFS на маркер доступа ЦИ DSS

Полученный на предыдущем шаге маркер доступа ADFS необходимо обменять на маркер ЦИ DSS с помощью запроса, указанного в начале статьи.

```
POST /STS/oauth/token HTTP/1.1
Host: simdss.cryptopro.ru
Content-Type: application/x-www-form-urlencoded
cache-control: no-cache

grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Atoken-exchange
&client_id=12cf4adb-ba3e-40d0-a5d9-fa507ef26932
&resource=urn%3Acryptopro%3Adss%3Asignserver%3ASignServer
&subject_token=eyJ0
&subject_token_type=urn%3Aietf%3Aparams%3Aoauth%3Atoken-type%3Ajwt
```

В ответ ЦИ вернёт новый маркер, который необходимо использовать для авторизации запросов к REST API Сервиса Подписи

```
{
  "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6IjhbVUQ191Q1VzcXpmS
  "expires_in": 300,
  "token_type": "Bearer"
}
```

Производительность

В данном разделе приводятся сведения о производительности сервисов СЭП КристоПро DSS.

Раздел состоит из следующих частей:

1. [Общие сведения](#)
2. [Аппаратная конфигурация](#)
3. [Пропускная способность сети](#)
4. [Пропускная способность хэширования](#)
5. [Производительность подписи](#)

Общие сведения

В данном разделе приводятся основные понятия и определения, используемые в последующих разделах.

Понятия и определения

Производительность – под производительностью понимается количество операций, выполняемых сервисом в определённый интервал времени при полной загрузке ресурсов сервиса. Иногда для этого же понятия используется термин пропускная способность (throughput).

В последующих разделах производительность измеряется в операциях в секунду (оп/с), если не оговорено иное.

Операция – неделимый набор действий, выполняемый сервисом.

Операция может состоять из вызовов нескольких методов сервиса. Например, «операция подписи» может состоять из вызова двух методов сервиса: получения маркера доступа и непосредственно подписи.

Сервис – сущность, отвечающая за обработку клиентских запросов. В общем случае может состоять из нескольких разнородных компонентов (сервер приложений IIS, веб-приложение, SQL-сервер и т.п.).

Клиент (клиентское приложение) – сущность, отвечающая за отправку запросов на выполнение операций.

Примечание

Следует особенно отметить, что все проводимые измерения относятся к производительности сервиса, а не клиента. Все полученные результаты отражают не количество операций, выполненных клиентом, а количество операций, выполненных сервисом.

Ресурсы сервиса – аппаратные ресурсы, доступные сервису для выполнения операций. К ним относятся жёсткий диск (в таблицах DISC), оперативная память (в таблицах RAM), процессор (в таблицах CPU) и сетевое оборудование (в таблицах NET).

При нехватке хотя бы одного из ресурсов дальнейший рост производительности невозможен. Все показатели, полученные в ходе измерений, соответствуют полной загрузке одного или нескольких ресурсов сервера. Если во время какого-либо измерения не удалось достичь полной загрузки ресурсов, это будет указано отдельно.

Аппаратная конфигурация

В разделе приводятся сведения об аппаратной конфигурации серверов, используемых в нагрузочном тестировании.

Нагрузочное тестирование проводилось в четырёх конфигурациях:

- 1 сервер СЭП КристоПро DSS и 1 ПАКМ КристоПро HSM.
- 1 сервер СЭП КристоПро DSS и 2 ПАКМ КристоПро HSM.
- 2 сервера СЭП КристоПро DSS и 1 ПАКМ КристоПро HSM.
- 2 сервера СЭП КристоПро DSS и 2 ПАКМ КристоПро HSM

Во всех конфигурациях SQL-сервер располагается на отдельном сервере.

Аппаратная конфигурация всех серверов одинакова и представлена в таблице.

	DSS-TEST
CPU	Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz Максимальная скорость: 2,40 ГГц Сокетов: 2 Ядра: 12 Логических процессоров: 24 Виртуализация: Включено Кэш L1: 768 КБ Кэш L2: 3,0 МБ Кэш L3: 30,0 МБ
RAM	32,0 ГБ Other Скорость: 2133 МГц Использовано гнезд: 4 из 24 Тип: DIMM Зарезервировано аппаратно: 105 МБ
DISC	TOSHIBA-HDWD105 465.76 ГБ
NET	Intel(R) Ethernet X540-T1 #2 10 Гбит/с

В качестве балансировщика используется Pfense (компонент HAProxy).

В последующих разделах при ссылке на сервер СЭП КристоПро DSS будет использоваться обозначение `DSS-TEST`.

Ссылка на конкретную конфигурацию будет указываться после обозначения сервера в квадратных скобках `[i,j]`, где `i` - количество серверов СЭП КристоПро DSS, `j` - количество ПАКМ КристоПро HSM (`i`, `j` могут принимать значения 1 или 2).

Пример

`DSS-TEST[1,2]` - конфигурация с одним сервером СЭП КристоПро DSS и двумя ПАКМ КристоПро HSM.

Пропускная способность сети

В данном разделе рассматриваются ограничения, накладываемые сетевым оборудованием на максимальное количество запросов к сервису.

В таблице приведено максимальное количество одновременно передаваемых запросов для сетей различной пропускной способности.

В таблице приведены размеры:

- исходного документа
- сформированного запроса к сервису

Размер запроса примерно на 33% больше размера исходного документа. Увеличение размера запроса в сравнении с исходным документом связано с кодированием документа в Base64, а также наличием маркера доступа (в среднем 2500 байтов) и другой служебной информации.

РАЗМЕР ДОКУМЕНТА	РАЗМЕР ЗАПРОСА (В)	СЕТЬ 1 ГБИТ/С	СЕТЬ 10 ГБИТ/С
32 В	2 577	50 862	508 622
1KB	3 902	33 591	335 910
10 KB	16 189	8 096	80 964
50 KB	70 801	1 851	18 513
100 KB	139 069	942	9 425
500 KB	685 201	191	1 913
1 MB	1 400 637	94	936
2 MB	2 798 737	47	468
3 MB	4 196 837	31	312
4 MB	5 594 941	23	234
5 MB	6 993 041	19	187
6 MB	8 391 141	16	156
10 MB	13 983 549	9	94
20 MB	27 964 561	4.69	47
50 MB	69 907 601	1.87	19
100 MB	139 812 669	0.94	9

Из представленной ниже таблицы следует, что при обработке документа размером 100 МБ производительность не может превышать 9 операций в секунду даже в высокоскоростных сетях передачи данных (10 Гбит/с).

В сети с пропускной способностью 1 Гбит/с количество операций в секунду не может превышать 191.

Следует отметить, что в приведённых в таблице данных не учитывается размер ответа сервера. Допущение справедливо для случая формирования откреплённой CAdES подписи. При создании подписи других форматов (присоединённая CAdES подпись, XMLDSig, PDF, MSOffice) указанные в таблице значения необходимо уменьшить вдвое.

Формат CAdES подписи также имеет значение, так как размер отдельной CAdES XLT1 подписи примерно в 7 раз больше отдельной CAdES BES подписи.

Производительность вычисления хеш-функции

В данном разделе приведены данные о производительности вычисления значения хеш-функции для различных алгоритмов хэширования, различных размеров входных данных в зависимости от числа потоков.

Данные собирались с сервера `DSS-TEST`. Параметры сервера приведены в разделе [Аппаратная конфигурация](#)).

На основе данных результатов можно сделать вывод о пропускной способности хэширования, выраженной в МБ/с, для одного потока.

ГОСТ Р 34.11 - 2001	ГОСТ Р 34.11 - 2012 (256)	ГОСТ Р 34.11 - 2012 (512)
60 МБ/с	100 МБ/с	100 МБ/с

Алгоритм ГОСТ Р 34.11 - 2001

В таблице приведены данные о производительности операции хэширования для алгоритма ГОСТ Р 34.11 - 2001.

РАЗМЕР ДОКУМЕНТА	1 ПОТОК	12 ПОТОКОВ	24 ПОТОКА
1 КБ	46 886	380 253	357 814
10 КБ	5 664	59 256	421 573
50 КБ	1 190	12 130	105 699
100 КБ	523	6 176	21 956
500 КБ	111	1 204	10 992
1 МБ	53	583	2 226
2 МБ	27	289	1 078
3 МБ	21	188	534
4 МБ	12	149	340
5 МБ	10,96	119	263
10 МБ	5,34	62	206
20 МБ	2,57	30	103
50 МБ	1,09	12	51
100 МБ	0,58	6	21

Алгоритм ГОСТ Р 34.11 - 2012 (256)

В таблице приведены данные о производительности операции хэширования для алгоритма ГОСТ Р 34.11 - 2012 (256).

РАЗМЕР ДОКУМЕНТА	1 ПОТОК	12 ПОТОКОВ	24 ПОТОКА
1 КБ	71 507	415 181	401 330
10 КБ	9 559	82 195	134 825
50 КБ	1 929	17 517	28 904
100 КБ	1 053	9 496	14 544
500 КБ	219	1 855	2 932
1 МБ	107	868	1 431
2 МБ	56	480	716
3 МБ	30	294	478
4 МБ	19	225	358
5 МБ	19	183	285
10 МБ	10,37	97	142
20 МБ	4,28	43	71
50 МБ	1,90	19	29
100 МБ	0,94	10	14

Алгоритм ГОСТ Р 34.11 - 2012 (512)

В таблице приведены данные о производительности операции хэширования для алгоритма ГОСТ Р 34.11 - 2012 (512).

РАЗМЕР ДОКУМЕНТА	1 ПОТОК	12 ПОТОКОВ	24 ПОТОКА
1 КБ	67 796	411 476	407 288
10 КБ	8 556	82 340	134 689
50 КБ	1 981	17 972	28 876
100 КБ	967	9 826	14 516
500 КБ	202	2 016	2 923
1 МБ	90	856	1 429
2 МБ	47	476	714
3 МБ	32	307	477

РАЗМЕР ДОКУМЕНТА	1 ПОТОК	12 ПОТОКОВ	24 ПОТОКА
4 МБ	24	245	358
5 МБ	19	177	286
10 МБ	9,66	95	143
20 МБ	4,54	44	72
50 МБ	2,06	18	29
100 МБ	0,94	10	14

Подпись документа

В данном разделе рассматривается производительность КриптоПро DSS при выполнении операции подписи. Влияние аутентификации сведено к минимуму путём получения маркера доступа перед началом выполнения теста.

Данные получены для подписи формата CAdES BES (отделённая) и алгоритма подписи ГОСТ Р 34.10 - 2012 (256).

РАЗМЕР ДОКУМЕНТА	DSS-TEST[1,1]	DSS-TEST[1,2]	DSS-TEST[2,1]	DSS-TEST[2,2]
32 B	1900	1990	2314 (60% CPU)	3163
1 KB	1892	1976	2319 (60% CPU)	3137
10 KB	1850	1868	2300 (60% CPU)	3025
50 KB	1210	1209	2087	2100
100 KB	887	890	1450	1417
500 KB	385	380	436 (70% CPU)	451 (75% CPU)
1 MB	216	219	228 (60% CPU)	220 (60% CPU)
2 MB	112	115	117 (50% CPU)	116 (50% CPU)
3 MB	70	72	77 (45% CPU)	78 (45% CPU)
4 MB	55	56	57 (30% CPU)	59 (30% CPU)
10 MB	21	22	23 (30% CPU)	25 (30% CPU)
20 MB	11	11	13 (30% CPU)	14 (30% CPU)

Документы можно разделить на три категории

- *маленькие* (до 10 KB),
- *средние* (от 10 KB до 500 KB)
- *большие* (500 KB и больше).

Операция подписи маленьких документов полностью загружает CPU одного сервера DSS в конфигурации DSS-TEST[1,1]. Конфигурация DSS-TEST[1,2] с двумя серверами HSM не приводит к увеличению производительности и показывает аналогичные результаты. Из этого следует, что операция подписи маленьких документов не загружает HSM полностью. При включении двух серверов DSS в балансировку в конфигурации DSS-TEST[2,1] один сервер HSM оказывается полностью загружен, что приводит к падению загрузки CPU на серверах DSS. При дублировании DSS и HSM в конфигурации DSS-TEST[2,2] загрузка снова смещается на серверы DSS.

Операция подписи средних документов требует больше ресурсов для десериализации содержимого запросов и хэширования документа. Увеличение размера документа приводит к постепенному снижению производительности. Сервер DSS во время выполнения данной операции загружен полностью, переход от конфигурации `DSS-TEST[1,1]` к `DSS-TEST[2,1]` приводит к существенному росту производительности, при этом добавление второго сервера HSM на производительность системы не влияет.

Операция подписи больших документов требует существенных сетевых ресурсов. Во всех конфигурациях система показала одинаковые результаты, загрузка сети при этом не поднималась выше 3 Гбит/с. Указанное поведение требует дальнейшего исследования (возможные причины: ошибки в реализации клиента, приводящие к тому, что он не может полностью использовать всю ширину канала для отправки сообщений; ограничения пропускной способности HAProxy). По мере получения новой информации данные в таблице и содержимое всего раздела будет дополняться.

Отдельно следует рассмотреть операцию подписи больших документах в сетях с небольшой пропускной способностью. Документы размером выше, чем 500 KB полностью загружают сеть пропускной способностью 1Гбит/с, что подтверждается данными раздела [Пропусная способность сети](#): максимальная производительность при данном размере документов не может превосходить 191 операцию в секунду. При этом чем больше размер документа, тем меньше нагружается CPU и тем сильнее падает производительность, так как сервис большую часть времени проводит на чтении данных из канала.

РАЗМЕР ДОКУМЕНТА	<code>DSS-TEST[1,1]</code>
500 KB	151 (80% CPU, 90% NET)
1 MB	72 (72% CPU, 90% NET)
2 MB	36 (65% CPU, 90% NET)
3 MB	24 (56% CPU, 90% NET)
4 MB	19 (46% CPU, 90% NET)

REST API Сервиса Подписи

Данный раздел содержит руководство разработчика по интеграции с Сервисом Подписи КристоПро DSS. В разделе приведено подробное описание методов и типов данных REST-интерфейса Сервиса Подписи, сценариев взаимодействия с Сервисом Подписи с примерами HTTP запросов и ответов.

Сценарии:

- [Создание запроса на сертификат](#)
- [Примеры подписи](#)
- [Подтверждение операций](#)
- [Потоковая обработка](#)
- [Асинхронная подпись](#)

Конечные точки:

- [Список конечных точек и методов](#)

Примеры:

- [Отображаемая подпись PDF-документов](#)
- [Передача параметров шаблона в метод подписи документов](#)

Типы данных:

- [Политика Сервиса Подписи \(DssPolicy\)](#)
- [Параметры УЦ \(DssCaPolicy\)](#)
- [Параметры криптопровайдеров \(DSSCSPPolicy\)](#)
- [Параметры запроса на сертификат \(CertificateRequest\)](#)
- [Запрос на сертификат \(DSSCertRequest\)](#)
- [Сертификат \(DSSCertificateEx\)](#)
- [Статус сертификата \(CertificateStatus\)](#)
- [Типы статусов сертификата \(DSSCertificateStatusEnum\)](#)
- [Информация об отзыве сертификата \(RevocationInfo\)](#)
- [Причины отзыва сертификата \(CertRevokeReasonEnum\)](#)
- [Данные о документе \(Document\)](#)
- [Данные о пакете документов \(DocumentPackage\)](#)
- [Подписанный пакет документов \(DSSSignDocumentResponse\)](#)
- [Данные транзакций \(Transaction\)](#)
- [Параметры транзакций](#)
- [Типы транзакций](#)
- [Форматы представления сертификата и запроса на сертификат \(DSSCertificateFormatEnum\)](#)
- [Тип запроса \(CARquestTypeEnum\)](#)
- [Статус запроса на сертификат \(DSSRequestStatusEnum\)](#)
- [Сведения об отзыве/приостановлении/восстановлении \(DSSRevRequest\)](#)
- [Данные для смены ПИН-кода \(RequestChangePin\)](#)
- [Данные для назначения сертификата по умолчанию \(DefaultProperty\)](#)
- [Данные для назначения дружественного имени \(FriendlyNameProperty\)](#)
- [Результат расшифрования \(DSSDecryptDocumentResponse\)](#)
- [Типы шифрования \(EncryptionType\)](#)

Создание запроса на сертификат

Параметры выпуска запроса на сертификат можно получить из Политики Сервиса Подписи (метод [/policy](#)). Политика Сервиса Подписи содержит:

- Список параметров Удостоверяющих Центров, подключенных к DSS
- Список криптопровайдеров, подключенных к DSS

Каждый элемент списка [параметров УЦ](#) содержит:

- Идентификатор Удостоверяющего Центра
- Тип Удостоверяющего Центра
- Шаблон различительного имени (Distinguished Name)
- Список шаблонов сертификатов
- Отображаемое имя

В интерфейсе интегрируемой системы должна быть возможность выбора Удостоверяющего Центра, для которого будет создан запрос на сертификат. Для каждого Удостоверяющего Центра Сервис Подписи передаёт отображаемое имя (DSSCAPolicy -> `Name`), которое может быть показано пользователю.

Для выбранного пользователем Удостоверяющего Центра в интерфейсе интегрируемой системы должна отображаться форма для заполнения Идентифицирующих данных. Форма составляется в соответствии с шаблоном имени (DSSCAPolicy -> `NamePolicy`). У каждого компонента имени в шаблоне есть отображаемое имя (`Name`), строковый идентификатор (`StringIdentifier`) и требование к заполнению (`IsRequired`).

Так же на форме создания запроса должен быть отображен список шаблонов сертификатов (`EkuTemplates`). Каждый шаблон сертификата имеет отображаемое имя.

Если Политика Сервиса Подписи содержит более одного криптопровайдера, то необходимо предоставить пользователю возможность выбора.

Данные с формы передаются в метод [/requests](#) для создания запроса на сертификат:

- Идентификатор Удостоверяющего Центра
- Различительное имя
- Шаблон сертификата
- ПИН-код на закрытый ключ (опционально)
- Идентификатор криптопровайдера (опционально)

Данные передаются в структуре [CertificateRequest](#).

Идентификатор Удостоверяющего Центра (`AuthorityId`) является константой. Он может быть получен от Администратора DSS и зафиксирован в настройках интегрируемой системы.

Примечание

Если Удостоверяющий Центр с заданным идентификатором отсутствует в Политике Сервиса Подписи, то либо он недоступен в данный момент, либо был отключен Администратором DSS. Для выяснения причин недоступности Удостоверяющего Центра следует обратиться к Администратору DSS.

Различительное имя может быть передано в двух форматах:

- Список пар oid:value (`DistinguishedName`)
- Строковое представление (`RawDistinguishedName`)

Объектные идентификаторы (**OID**) компонентов имени указаны в шаблоне имени.

Примечание

Шаблон сертификата представляет собой набор объектных идентификаторов, которые попадут в расширение Enhanced Key Usage (EKU) запроса на сертификат, или идентификатор шаблона сертификата КриптоПро УЦ 2.0, который попадёт в расширение Certificate Template (1.3.6.1.4.1.311.21.7).

Шаблон передаётся через разные поля запроса на сертификат в зависимости от типа:

- Enhanced key usage - передаётся в дополнительных параметрах запроса `Parameters` в ключе `EkuString` в формате `oid1,oid2,...,oidN`.

Примечание

Данный шаблон используется при создании запроса на сертификат к Удостоверяющему Центру типа 0 (КриптоПро УЦ 1.5) и 2 (Сторонний УЦ).

- Certificate Template - передаётся в параметре `Template` запроса на сертификат.

Примечание

Данный шаблон используется при создании запроса на сертификат к Удостоверяющему Центру типа 1 (КриптоПро УЦ 2.0) и 2 (Сторонний УЦ).

Идентификатор криптопровайдера должен быть задан, если в Политике Сервиса Подписи доступно более одного криптопровайдера. Идентификатор криптопровайдера (DSSCSPPolicy -> `GroupId`) передаётся в дополнительных параметрах запроса в ключе `GroupId`

Создание запроса на сертификат с подтверждением при помощи вторичной аутентификации

При создании запроса на сертификат с подтверждением при помощи вторичной аутентификации требуется выполнить следующую последовательность действий (шагов):

- [Создание транзакции на Сервисе Подписи](#)
- [Подтверждение транзакции на Сервисе Подтверждения Операций](#)
- [Получение результата операции на Сервисе Подписи](#)

При этом в массив параметров транзакции метода `/transactions` должны быть отображены следующие поля [запроса на сертификат](#):

CERTIFICATEREQUEST	ПАРАМЕТРЫ ТРАНЗАКЦИИ
AuthorityId	CAId
PinCode	Не используется
Template	CertTemplateOid
DistinguishedName	Не используется
RawDistinguishedName	CertSubjectName
Parameters -> EkuString	EkuString
Parameters -> GroupId	GroupId

Примечание

При создании запроса на сертификат с подтверждением с подтверждением при помощи вторичной аутентификации различительное имя может быть передано только в строковом представлении.

Примеры запросов

Пример запроса с указанием различительного имени в строковом представлении:

```
POST https://host/SignServer/rest/api/requests HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGci ... 2bgrniEg
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 153
Expect: 100-continue

{
  "AuthorityId":11,
  "PinCode":"","
  "RawDistinguishedName":"CN=dssUser,C=RU",
  "Parameters":
    {"EkuString":"1.2.643.2.2.34.2,1.2.643.2.2.34.4,1.3.6.1.5.5.7.3.2"}
}
```

Пример запроса с указанием различительного имени в виде набора компонентов:

```
POST https://host/SignServer/rest/api/requests HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJ... PhYmXscTmwGkD8b1SWy0nYQ
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 169
Expect: 100-continue

{
  "AuthorityId":11,
  "PinCode":"","
  "DistinguishedName":{"2.5.4.3":"dssUser","2.5.4.6":"RU"},
  "Parameters":
    {
      "EkuString":"1.2.643.2.2.34.2,1.2.643.2.2.34.4,1.3.6.1.5.5.7.3.2"
    }
}
```

Пример запроса с указанием шаблона сертификата:

```
POST https://host/SignServer/rest/api/requests HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV... Ysj1GpIVmR2hw
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 135
Expect: 100-continue

{
  "AuthorityId":11,
  "PinCode":"",
  "Template":"1.3.6.1.5.5.7.3.2",
  "DistinguishedName":{"2.5.4.3":"dssUser","2.5.4.6":"RU"},
  "Parameters":{}}
```

Пример ответа:

```
HTTP/1.1 200 OK
Content-Length: 723
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
Date: Tue, 04 Sep 2018 13:35:02 GMT

{
  "CertificateType": "ServerSide",
  "Base64Request": "MIIBQDCB8AIBADAFMQswCQYD ... i0ibLabDHZ2VY1G8CsaxjE",
  "CertificateAuthorityID": 11,
  "CADisplayName": null,
  "DistName": "CN=dssUser, C=RU",
  "Subject": "dssUser",
  "Status": "PENDING",
  "ID": 22,
  "CARequestID": null,
  "CertificateID": 0,
  "RequestType": "Certificate",
  "GroupID": "e8e67f9e-7eed-4116-ad98-20582e4d766e"}
```

Запрос на сертификат **с подтверждением** с подтверждением при помощи вторичной аутентификации:

```
POST https://host/SignServer/rest/api/transactions HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cGEiOiJ1bmRlciJ9.pz4erYJpgoN_RgQLA
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 200
Expect: 100-continue

{
  "OperationCode": 16,
  "Parameters": [
    { "Name": "CertSubjectName", "Value": "CN=dssUser, C=RU" },
    { "Name": "CAId", "Value": "11" },
    { "Name": "EkuString", "Value": "1.2.643.2.2.34.2, 1.2.643.2.2.34.4, 1.3.6.1.5.5.7.3.2" }
  ]
}
```

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	pending_requests_exist	У пользователя есть необработанный запрос на сертификат (статус PENDING).

Обработка ответа Сервиса Подписи

При успешном создании запроса на сертификат Сервис Подписи в ответе вернёт структуру [DSSCertRequest](#).

Дальнейшее поведение пользователя зависит от значения поля `Status` в структуре [DSSCertRequest](#) и типа УЦ, на котором создавался запрос на сертификат.

ACCEPTED - запрос на сертификат принят и обработан УЦ. В данном случае в поле `CertificateID` будет записан идентификатор выпущенного сертификата.

REGISTRATION - запрос на сертификат принят в КриптоПро УЦ 2.0 и находится на этапе регистрации пользователя УЦ. В зависимости от настроек подключения DSS к КриптоПро УЦ 2.0, необходимо:

- ожидать одобрения запроса на сертификат Администратором УЦ;
- одобрить запрос Оператором DSS.

PENDING - запрос на сертификат находится в обработке. Если запрос отправлен на КриптоПро УЦ 2.0, то в зависимости от

настроек подключения DSS к КриптоПро УЦ 2.0 необходимо:

- ожидать одобрения запроса на сертификат Администратором УЦ;
- одобрить запрос Оператором DSS.

Если запрос создавался через "Сторонний Удостоверяющий Центр", необходимо:

- скачать запрос на сертификат по идентификатору [/requests](#);
- передать запроса на сертификат в УЦ;
- выпущенный сертификат установить в DSS.

Запрос на сертификат (PKCS#10) в формате Base64 содержится в поле `Base64Request` структуры [DSSCertRequest](#).

REJECTED - запрос отклонён. Дальнейшая обработка запроса невозможна. Для выяснения причин отклонения запроса необходимо обратиться к Администратору УЦ.

Подтверждение операций

Данный раздел содержит руководство разработчика по подтверждению операций на Сервисе Подписи. В разделе приведен общий подход при подтверждении операций.

В подтверждении операций задействованы следующие сервисы DSS:

КОНЕЧНАЯ ТОЧКА	СЕРВИС	ОПИСАНИЕ
https://<host>/<StsAppName>/oauth	Сервис Аутентификации.	Аутентификация пользователей для возможности обращений к Сервису Подписи
https://<host>/<SignServerAppName>/rest/api	Сервис Подписи	Создание транзакций и получение результатов, подтвержденной операции
https://<host>/<StsAppName>/confirmation	Сервис Подтверждения Операций	Подтверждение транзакций

Подтверждены вторым фактором аутентификации могут быть операции требующие доступа к закрытому ключу. Список подтверждаемых операций - [Коды операций](#)

Последовательность шагов:

- Аутентификация пользователя
- [Получение политики Сервиса Подписи \(опционально\)](#)
- [Создание транзакции на Сервисе Подписи](#)
- [Подтверждение транзакции на Сервисе Подтверждения Операций](#)
- [Получение результата операции на Сервисе Подписи](#)

Получение Политики Сервиса Подписи

Для получения Политики Сервиса Подписи необходимо обратиться на конечную точку [/policy](#).

Из Политики Сервиса Подписи можно получить:

- Список операций, требующих подтверждения
- Параметры создания запроса на сертификат

Список операций доступных пользователю содержится в структуре [ActionPolicy](#). Элементами списка являются объекты вида:

ПОЛЕ	ТИП	ОПИСАНИЕ
DisplayName	string	Отображаемое имя операции
Uri	string	Идентификатор операции
Action	string	Имя операции
MfaRequired	bool	Требование подтверждения операции

Список имён операций приведён в разделе [Коды операций](#)

Примечание

Любую операцию на Сервисе Подписи можно выполнить с подтверждением вторым фактором аутентификации, даже если для данной операции не выставлено требование подтверждения.

Пример ActionPolicy


```

"ActionPolicy": [{
  "DisplayName": "Выпуск маркера (вход в ЦИ)",
  "Uri": "http://dss.cryptopro.ru/identity/claims/action/issue",
  "Action": "Issue",
  "MfaRequired": false
}, {
  "DisplayName": "Подпись документа",
  "Uri": "http://dss.cryptopro.ru/identity/claims/action/signdocument",
  "Action": "SignDocument",
  "MfaRequired": false
}, {
  "DisplayName": "Подпись пакета документов",
  "Uri": "http://dss.cryptopro.ru/identity/claims/action/signdocuments",
  "Action": "SignDocuments",
  "MfaRequired": false
}, {
  "DisplayName": "Расшифрование документа",
  "Uri": "http://dss.cryptopro.ru/identity/claims/action/decryptdocument",
  "Action": "DecryptDocument",
  "MfaRequired": false
}, {
  "DisplayName": "Создание запроса на сертификат",
  "Uri": "http://dss.cryptopro.ru/identity/claims/action/createrequest",
  "Action": "CreateRequest",
  "MfaRequired": false
}, {
  "DisplayName": "Смена пин-кода закрытого ключа",
  "Uri": "http://dss.cryptopro.ru/identity/claims/action/changepin",
  "Action": "ChangePin",
  "MfaRequired": false
}, {
  "DisplayName": "Обновление сертификата",
  "Uri": "http://dss.cryptopro.ru/identity/claims/action/renewcertificate",
  "Action": "RenewCertificate",
  "MfaRequired": false
}, {
  "DisplayName": "Отзыв сертификата",
  "Uri": "http://dss.cryptopro.ru/identity/claims/action/revokecertificate",
  "Action": "RevokeCertificate",
  "MfaRequired": false
}, {
  "DisplayName": "Приостановление действия сертификата",
  "Uri": "http://dss.cryptopro.ru/identity/claims/action/holdcertificate",
  "Action": "HoldCertificate",
  "MfaRequired": false
}, {
  "DisplayName": "Возобновление действия сертификата",
  "Uri": "http://dss.cryptopro.ru/identity/claims/action/unholdcertificate",
  "Action": "UnholdCertificate",
  "MfaRequired": false
}, {
  "DisplayName": "Удаление сертификата",
  "Uri": "http://dss.cryptopro.ru/identity/claims/action/deletecertificate",
  "Action": "DeleteCertificate",
  "MfaRequired": false
}, {
  "DisplayName": "Доступ к закрытому ключу",
  "Uri": "http://dss.cryptopro.ru/identity/claims/action/privatekeyaccess",
  "Action": "PrivateKeyAccess",
  "MfaRequired": false
}
]

```


ПОЛЕ	ОПИСАНИЕ
Error	Ошибка обработки запроса. Заполняется при <code>IsFinal</code> - false
ErrorDescription	Подробное описание ошибки обработки запроса

Поле `Challenge` содержит:

ПОЛЕ	ОПИСАНИЕ
Title	Текст, который вызывающая система может отобразить пользователю в своём интерфейсе
TextChallenge	Дополнительные данные для подтверждения операции

Содержимое поля `TextChallenge` зависит от метода подтверждения (OTP-via-SMS, myDSS, и т.д.)

Примечание

`RefId` - Идентификатор транзакции, созданной на Сервисе Подтверждения Операций. Идентификатор необходимо будет использовать при последующих обращениях на конечную точку `/confirmation`.

Примечание

При обработке ответа Сервиса Подтверждения Операций вызывающее приложение должно смотреть на значение двух флагов: `IsFinal` и `IsError`.

Если получен ответ с `IsError` - true, то дальнейшее подтверждение транзакции не возможно.

Если получен ответ с `IsFinal` - false, то подтверждение транзакции ещё не завершено.

Дальнейшие действия пользователя зависят от метода подтверждения (OTP-via-SMS, myDSS, и т.д.)

Пример подтверждения операции через мобильное приложение myDSS смотреть [здесь](#)

Результатом подтверждения транзакции на Сервисе Подтверждения Операций является `AccessToken`, содержащий идентификатор подтверждённой транзакции.

Получение результата

Для получения результата операции необходимо вызвать метод соответствующий типу транзакции (см. таблицу ниже).

В заголовке `Authorization` HTTP-запроса клиент должен указать токен полученный на шаге "Подтверждение транзакции"
`Authorization: Bearer <access_token2>`.

Примеры запроса

ТИП ТРАНЗАКЦИИ	КОД ТРАНЗАКЦИИ	МЕТОД
SignDocument	2	/documents
SignDocuments	4	/documents/packagesignature
DecryptDocument	8	
CreateRequest	16	/request

ТИП ТРАНЗАКЦИИ	КОД ТРАНЗАКЦИИ	МЕТОД
ChangePin	32	
RenewCertificate	64	
RevokeCertificate	128	
HoldCertificate	256	
UnholdCertificate	512	
DeleteCertificate	1024	

Примечание

Методы вызываются без параметров. Исключение составляют методы требующие ввода ПИН-код: подпись документа, подпись пакета документов.

Потоковая обработка

Для обработки документов большого размера требуются значительные ресурсы со стороны сервера, так как весь документ на время выполнения криптографической операции хранится в памяти, для обхода этого ограничения следует применять режим потоковой передачи данных.

В потоковом режиме обработки запроса сервер начинает выполнение операции сразу после получения первых байтов документа, из входного потока данные считываются небольшими блоками, без буферизации всего содержимого в памяти.

Сервис Подписи поддерживает выполнение следующих операций в потоковом режиме:

ОПЕРАЦИЯ	МЕТОД
Подпись	/documents
Хэширование	/documents/hash

Примечание

Выполнение операций требует аутентификации пользователя. Поточное хэширование документа без аутентификации возможно на сервисе DSS Lite - конечная точка `/hash`.

Формат запроса

Для того чтобы передать запрос на выполнение операции в потоковом режиме клиентское приложение должно сформировать следующий HTTP запрос:

```
POST /SignServer/rest/api/documents HTTP/1.1
CPDSS-OPERATION-PARAMS: eyJTa...
Content-Type: application/octet-string
Authorization: Bearer eyJ0eXA...
Host: dss.cryptopro.ru
```

В отличие от обычного режима передачи данных, в потоковом режиме содержимое обрабатываемого документа передаётся в теле запроса, а параметры операции - в заголовке `CPDSS-OPERATION-PARAMS`. Тип содержимого при этом должен быть `application/octet-string` (можно не указывать).

Формирование заголовка `CPDSS-OPERATION-PARAMS`

Заголовок `CPDSS-OPERATION-PARAMS` в качестве значения содержит параметры операции (то есть запрос в обычном режиме, из которого убран параметр `Content`).

```
CPDSS-OPERATION-PARAMS: BASE64URL(UTF8Bytes(json-request-wo-content))
```

Пример запроса на подпись

Рассмотрим запрос на подпись документа в стандартном режиме (переносы строк добавлены для удобства чтения):

```
{
  "Content": "AX/dfse...",
  "Signature": {
    "Type": "CADES",
    "CertificateId": 0,
    "Parameters": {
      "CADESType": "XLT1",
      "IsDetached": "true",
      "TSPAddress": "http://testca2.cryptopro.ru/TSP/tsp.srf"
    }
  }
}
```

В потоковом режиме в заголовок помещается значение

```
eyJTawduYXR1cmUiIDogeyAivHlwZSI6IkNBZEVTIiwgIkNlcnRpZmljYXRlSWQiOjAsICJQYXJhbWV0ZXJzIjp7IkNBREVTVHlwZSI6IlhMV
DEiLCAiSXNEZXRhY2hlZCI6InRydWUiLCAiVFNQWRkcmVzcyI6Imh0dHA6Ly90ZXN0Y2EyLmNyeXB0b3Byby5ydS9UU1AvdHNwLnNyZiJ9fX
0
```

которое представляет собой значение `BASE64URL(UTF8Bytes(M))`, где `M`:

```
{"Signature" : { "Type": "CADES", "CertificateId": 0, "Parameters": {"CADESType": "XLT1", "IsDetached": "true",
" TSPAddress": "http://testca2.cryptopro.ru/TSP/tsp.srf" }}}
```

Пример запроса на хэширование

```
{
  "Hash": {
    "Parameters": {
      "HashAlgorithm": "GR 34.11-2012 256"
    }
  }
}
```

В потоковом режиме в заголовок помещается значение

```
eyJIYXNoIjpw7IlBhcmFtZXR1cnMiOnsiSGFzaEFsZ29yaXRobSI6IkdsIDM0LjExLTlwMTIgMjU2In19fQ
```

Ограничения

Потоковая подпись может применяться только для создания отделённой подписи форматов CADES (BES, T, XLT1).

Примеры запросов на создание подписи средствами REST API

Раздел содержит примеры HTTP-запросов к REST-интерфейсу Сервиса Подписи DSS для создания электронной подписи.

Пользовательский сертификат

При выполнении любой операции подписи Сервис должен располагать информацией о сертификате ключа подписи. Это можно сделать двумя способами:

- передача идентификатора сертификата в запросе на формирование подписи
- использование сертификата по умолчанию

Для получения идентификатора сертификата можно воспользоваться методом [GetCertificates](#).

Если необходимо использовать сертификат по умолчанию, то в качестве идентификатора сертификата в запросе следует передавать значение 0, предварительно [назначив](#) один из сертификатов пользователя сертификатом по умолчанию.

Форматы подписи

REST-интерфейс Сервиса Подписи DSS позволяет подписывать документы, используя следующие форматы электронной подписи:

- Необработанная подпись ГОСТ Р 34.10-2001/ГОСТ Р 34.10-2012 (Raw)
 - [Необработанная подпись документа](#)
 - [Необработанная подпись хэш-значения документа](#)
- Усовершенствованная подпись (CMS Advanced Electronic Signature)
 - [CMS \(CAAdES Basic Electronic Signature, CAAdES-BES\)](#)
 - [CAAdES-T \(CAAdES Timestamp\)](#)
 - [CAAdES XLT1 \(CAAdES X-Long Type 1\)](#)
- [Подпись XML-документов \(XML Digital Signature, XMLDSig\)](#)
- [Подпись документов Microsoft Office](#)
- [Подпись PDF-документов](#)

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_signature_params	Неверно указаны параметры подписи
400	invalid_certificate	Указан неверный идентификатор сертификата
400	invalid_pin	Указан неверный ПИН-код
400	content_required	Не передан документ для подписания
500	An error has occurred	Внутренняя ошибка сервера

Примечание

Одной из причин ошибки HTTP 500 может быть несоответствие формата документа и формата подписи. Например, для подписи формата PDF отправлен текстовый документ.

Сервис Подписи DSS позволяет формировать необработанную подпись двумя способами:

- [Необработанная подпись документа](#)
- [Необработанная подпись хэш-значения документа](#)

Необработанная подпись документа

Пример запроса

В примере создается простая подпись по стандарту ГОСТ Р 34.10-2001/ГОСТ Р 34.10-2012.

```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Expect: 100-continue
```

```
{
  "Content": "T2gsIGhlbGxvIHRobXJlIQ==",
  "Signature": {
    "Type": "GOST3410",
    "Parameters": {
      "Hash": "False"
    },
    "CertificateId": 1,
    "PinCode": ""
  }
}
```

Пример ответа

Сервис Подписи вернет закодированную в Base64 электронную подпись.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
"pq3fJ6NP1LWNW6yYgx0zOvdMSJaZBZ58L99IXvpVj/T/1+d0cbXqJHX6u7M7CGFuvyvb6mA6 ..."
```

Необработанная подпись хэш-значения

Пример запроса

В примере создается простая подпись хэш-значения по стандарту ГОСТ Р 34.10-2001/ГОСТ Р 34.10-2012.


```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Expect: 100-continue

{
  "Content": "mM40LMXbEv6DbpJe+ov8eCLRFHjglAZ22b3YebKmyU0=",
  "Signature":
    {
      "Type": "GOST3410",
      "Parameters":
        {
          "Hash": "True"
        },
      "CertificateId": 1,
      "PinCode": ""
    }
}
```

Примечание

При создании подписи хэш-значения в поле `Content` необходимо передавать хэш-значение подписываемого документа.

Пример ответа

Сервис Подписи вернет закодированную в Base64 электронную подпись.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
"pq3fJ6NP1LWNW6yYgx0zOvdMSJaZBZ58L99IXvpVj/T/1+d0cbXqJHX6u7M7CGFuvyvb6mA6 ..."
```

Подпись в формате CMS

Сервис Подписи DSS позволяет создавать следующие виды CMS-подписи:

- [Отделенная подпись](#)
- [Присоединенная подпись](#)
- [Отделенная подпись хэш-значения](#)
- [Соподпись документа](#)
- [Соподпись хэш-значения документа](#)
- [Пакетная соподпись документов](#)
- [Пакетная соподпись хэш-значений документов](#)
- [Заверяющая подпись](#)

Отделенная подпись

Пример запроса

В примере создается открепленная CAdES-BES подпись.

```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Expect: 100-continue
```

```
{
  "Content": "T2gsIGhlbGxvIHRoZXJlIQ==",
  "Signature": {
    "Type": "CADES",
    "Parameters": {
      "Hash": "False",
      "CADESType": "BES",
      "IsDetached": "True"
    },
    "CertificateId": 1,
    "PinCode": ""
  }
}
```

Пример ответа

Сервис Подписи вернет закодированное в Base64 содержимое файла с электронной подписью.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
"MIIFtWYJKoZIhvcNAQcCoIIFqDCCBaQCAQExDjgUAMAsGCSqGSIb3DQ..."
```

Присоединенная подпись

Пример запроса

В примере создается присоединенная CADES-BES подпись.

```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Expect: 100-continue
```

```
{
  "Content": "T2gsIGhlbGxvIHRoZXJlIQ==",
  "Signature": {
    "Type": "CADES",
    "Parameters": {
      "Hash": "False",
      "CADESType": "BES",
      "IsDetached": "False"
    },
    "CertificateId": 1,
    "PinCode": ""
  }
}
```

Пример ответа

Сервис Подписи вернет закодированное в Base64 содержимое файла с электронной подписью.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
"MIIFtWYJKoZIhvcNAQcCoIIFqDCCBaQCAQEUAAsGCSqGSIB3DQ ..."
```

Отделенная подпись хэш-значения

Пример запроса

В примере создается отделенная подпись хэш-значения документа в формате CAdES-BES.

```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh ... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Expect: 100-continue

{
  "Content": "j/MkCnzPzzIVmj42zxIRuu42akUzSVEq1fQTIuk1lPk=",
  "Signature":
    {
      "Type": "CAdES",
      "Parameters":
        {
          "Hash": "True",
          "CADESType": "BES",
          "IsDetached": "True"
        },
      "CertificateId": 1,
      "PinCode": ""
    }
}
```

Примечание

При создании подписи хэш-значения в поле **Content** необходимо передавать хэш-значение подписываемого документа.

Пример ответа

Сервис Подписи вернет закодированное в Base64 содержимое файла с электронной подписью.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
"MIIFtWYJKoZIhvcNAQcCoIIFqDCCBaQCAQExDjAMBggMAAsGCSqGSIB3DQ ..."
```

Соподпись документа

Пример запроса

В примере формируется соподпись документа с использованием второго сертификата ключа проверки электронной подписи.

```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Expect: 100-continue

{
  "Content": "MIIIpwYJKoZIhvcNAQcCoIIImDCCKiG9w0BBwGxAJOVQQJ ...",
  "Signature":
    {
      "Type": "CADES",
      "Parameters":
        {
          "Hash": "False",
          "CADESType": "BES",
          "IsDetached": "True",
          "OriginalDocument": "T2gsIGhlbGxvIHRoZXJlIQ==",
          "CmsSignatureType": "Cosign"
        },
      "CertificateId": 2,
      "PinCode": ""
    }
}
```

Примечание

При создании соподписи в поле `Content` необходимо передавать подписанное сообщение, в которое следует добавить соподпись.

В поле `OriginalDocument` необходимо передавать содержимое исходного документа.

В запросе должен присутствовать параметр `CmsSignatureType`, имеющий значение `cosign`.

Пример ответа

Сервис Подписи вернет закодированное в Base64 содержимое файла с электронной подписью.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
"MIIFtwYJKoZIhvcNAQcCoIIFqDCCBaQCAQEhDjAMBggMASGCSqGSib3DQ ..."
```

Соподпись хэш-значения документа

Пример запроса

В примере формируется соподпись хэш-значения документа с использованием второго сертификата ключа проверки электронной подписи.

```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Expect: 100-continue
```

```
{
  "Content": "MIIIPwYJKoZIhvcNAQcCoIIImDCCCJQCAQExDDAKBgYqh ...",
  "Signature": {
    "Type": "CADES",
    "Parameters": {
      "Hash": "True",
      "CADESType": "BES",
      "IsDetached": "True",
      "OriginalDocument": "mM40LMXbEv6DbpJe+ov8eCLRFHjglAZ22b3YebKmyU0=",
      "CmsSignatureType": "Cosign"
    },
    "CertificateId": 2,
    "PinCode": ""
  }
}
```

Примечание

При создании соподписи в поле `Content` необходимо передавать подписанное сообщение, в которое следует добавить соподпись.

В поле `OriginalDocument` необходимо передавать хэш-значение исходного документа.

В запросе должен присутствовать параметр `CmsSignatureType`, имеющий значение `cosign`.

Пример ответа

Сервис Подписи вернет закодированное в Base64 содержимое файла с электронной подписью.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
"MIIFtwYJKoZIhvcNAQcCoIIFqDCCBaQCAQExDjAMBggMASGCSqGSib3DQ ..."
```

Пакетная соподпись документа

Пример запроса

В примере формируется соподпись документов с использованием второго сертификата ключа проверки электронной подписи.

Содержимое исходных документов в случае пакетной отделённой соподписи передаётся в структуре `Documents` в поле `OriginalContent`. Параметр подписи `OriginalDocument` не используется.

```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
```

```
{
  "Documents" : [
    {
      "Content" : "MIIMngYJKoZIhvcNAQcCoIIMjzCCDIs...",
      "Name": "testdoc",
      "OriginalContent" : "0KLQtdGB0YLQvtCy0YvQtSDQtNCw0L3QvdGL0LUG0LTQu9GPINC/0L7QtNC/0LjRgdC4IDE="
    },
    {
      "Content" : "MIIMngYJKoZIhvcNAQcCoIIMjzCCDIs...",
      "Name": "testdoc1",
      "OriginalContent" : "0KLQtdGB0YLQvtCy0YvQtSDQtNCw0L3QvdGL0LUG0LTQu9GPINC/0L7QtNC/0LjRgdC4IDI="
    }
  ],
  "Signature":
  {
    "Type": "CADES",
    "Parameters":
    {
      "Hash": "False",
      "CADESType": "BES",
      "IsDetached": "True",
      "CmsSignatureType": "Cosign"
    },
    "CertificateId": 2,
    "PinCode": ""
  }
}
```

Примечание

При создании соподписи в поле **Content** необходимо передавать подписанное сообщение, в которое следует добавить соподпись.

В поле **OriginalContent** необходимо передавать содержимое исходного документа.

В запросе должен присутствовать параметр **CmsSignatureType**, имеющий значение **Cosign**.

Пример ответа

Сервис Подписи вернет закодированное в Base64 содержимое файла с электронной подписью.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
"MIIFtwYJKoZIhvcNAQcCoIIFqDCCBaQCAQExDjAMBggMASGCSqGSib3DQ ..."
```

Пакетная соподпись хэш-значения документа

Пример запроса

В примере формируется соподпись хэш-значения документов с использованием второго сертификата ключа проверки электронной подписи.

Хэш-значения документов в случае пакетной отведённой соподписи передаётся в структуре **Documents** в поле **OriginalContent**. Параметр подписи **OriginalDocument** не используется.

```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Expect: 100-continue
```

```
{
  "Documents" : [
    {
      "Content" : "MIIMngYJKoZIhvcNAQcCoIIMjzCCDIs...",
      "Name": "testdoc",
      "OriginalContent" : "mM40LMXbEv6DbpJe+ov8eCLRFHjglAZ22b3YebKmyU0="
    },
    {
      "Content" : "MIIMngYJKoZIhvcNAQcCoIIMjzCCDIs...",
      "Name": "testdoc1",
      "OriginalContent" : "mM40LMXbEv6DbpJe+ov8eCLRFHjglAZ22b3YebKmyU2="
    }
  ],
  "Signature":
  {
    "Type": "CADES",
    "Parameters":
    {
      "Hash": "True",
      "CADESType": "BES",
      "IsDetached": "True",
      "CmsSignatureType": "Cosign"
    },
    "CertificateId": 2,
    "PinCode": ""
  }
}
```

Примечание

При создании соподписи в поле **Content** необходимо передавать подписанное сообщение, в которое следует добавить соподпись.

В поле **OriginalContent** необходимо передавать хэш-значение исходного документа.

В запросе должен присутствовать параметр **CmsSignatureType**, имеющий значение **cosign**.

Пример ответа

Сервис Подписи вернет закодированное в Base64 содержимое файла с электронной подписью.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
"MIIFtWYJKoZIhvcNAQcCoIIFqDCCBaQCAQExDjAMBggMAsgCSqGSib3DQ ..."
```

Заверяющая подпись

Пример запроса

В примере формируется заверяющая подпись документа.

```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Expect: 100-continue

{
  "Content": "MIIIpwYJKoZIhvcNAQcCoIIImDCCCJQCAQExDDAKBgYqhQMCagkF ...",
  "Signature": {
    "Type": "CADES",
    "Parameters": {
      "Hash": "False",
      "CADESType": "BES",
      "IsDetached": "True",
      "OriginalDocument": "mM40LMXbEv6DbpJe+ov8eCLRFHjglAZ22b3YebKmyU0=",
      "CmsSignatureType": "Countersign"
    },
    "CertificateId": 2,
    "PinCode": ""
  }
}
```

Примечание

При создании заверяющей подписи в поле **Content** необходимо передавать подписанный документ, для которого требуется заверяющая подпись.

В поле **OriginalDocument** необходимо передавать исходный документ.

В запросе должен присутствовать параметр **CmsSignatureType**, имеющий значение **countersign**.

В запросе может присутствовать параметр **SignatureIndex**, указывающий индекс заверяемой подписи. В случае, если данный параметр отсутствует в запросе, Сервис Подписи будет заверять первую подпись в документе.

Пример ответа

Сервис Подписи вернет закодированное в Base64 содержимое файла с электронной подписью.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
"MIIFtWYJKoZIhvcNAQcCoIIFqDCCBaQCAQExDjAMBggMASGCSqGSib3DQ ..."
```

Подпись в формате CAdES-XLT1/CAdES-T

Сервис Подписи DSS позволяет создавать следующие виды подписи в формате CAdES-XLT1/CAdES-T:

- Отделенная подпись
- Присоединенная подпись
- Отделенная подпись хэш-значения
- Соподпись документа
- Соподпись хэш-значения документа
- Заверяющая подпись

Примечание

Подпись в формате CAdES-XLT1/CAdES-T требует обязательной передачи адреса TSP-службы посредством передачи параметра **TSPAddress** в запросе.

В случае, если в конфигурации Сервиса Подписи параметр `AllowThirdPartyTsp` имеет значение `false`, передаваемый адрес службы должен находиться в списке доверенных служб TSP, зарегистрированных на Сервисе Подписи.

Отделенная подпись

Пример запроса

В примере создается открепленная CAdES-XLT1 подпись.

```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Expect: 100-continue

{
  "Content": "mM40LMXbEv6DbpJe+ov8eCLRFHjglAZ22b3YebKmyU0=",
  "Signature": {
    "Type": "CAdES",
    "Parameters": {
      "Hash": "False",
      "CADESType": "XLT1",
      "IsDetached": "True",
      "TSPAddress": "https://dss-ca20-w12r2/tsp/tsp.srf"
    },
    "CertificateId": 1,
    "PinCode": ""
  }
}
```

Пример ответа

Сервис Подписи вернет закодированное в Base64 содержимое файла с электронной подписью.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
"MIIFtwYJKoZIhvcNAQcCoIIFqDCCBaQCAQEhDjgUAMAsGCSqGSIB3DQ ..."
```

Присоединенная подпись

Пример запроса

В примере создается присоединенная CAdES-XLT1 подпись.

```

POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Expect: 100-continue

{
  "Content": "mM40LMXbEv6DbpJe+ov8eCLRFHjglAZ22b3YebKmyU0=",
  "Signature": {
    "Type": "CADES",
    "Parameters": {
      "Hash": "False",
      "CADESType": "XLT1",
      "IsDetached": "False",
      "TSPAddress": "https://dss-ca20-w12r2/tsp/tsp.srf"
    },
    "CertificateId": 1,
    "PinCode": ""
  }
}

```

Пример ответа

Сервис Подписи вернет закодированное в Base64 содержимое файла с электронной подписью.

```

HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
"MIIFtWYJKoZIhvcNAQcCoIIFqDCCBaQCAQEUAAMsGCSqGSIb3DQ... "

```

Отделенная подпись хэш-значения

Пример запроса

В примере создается отделенная CADES-XLT1 подпись хэш-значения документа.

```

POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Expect: 100-continue

{
  "Content": "mM40LMXbEv6DbpJe+ov8eCLRFHjglAZ22b3YebKmyU0=",
  "Signature": {
    "Type": "CADES",
    "Parameters": {
      "Hash": "True",
      "CADESType": "XLT1",
      "IsDetached": "True",
      "TSPAddress": "https://dss-ca20-w12r2/tsp/tsp.srf"
    },
    "CertificateId": 1,
    "PinCode": ""
  }
}

```

Примечание

При создании подписи хэш-значения в поле **Content** необходимо передавать хэш-значение подписываемого документа.

Пример ответа

Сервис Подписи вернет закодированное в Base64 содержимое файла с электронной подписью.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
"MIIFtwYJKoZIhvcNAQcCoIIFqDCCBaQCAQEhDjAMBggMASGCSqGSIb3DQ ..."
```

Соподпись документа

Пример запроса

В примере осуществляется соподпись документа с использованием второго сертификата ключа проверки электронной подписи.

```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh ... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Expect: 100-continue

{
  "Content": "MIIPwYJKoZIhvcNAQcCoIIImDCCKiG9w0BBwGxAJ0VQQJ ...",
  "Signature":
  {
    "Type": "CADES",
    "Parameters":
    {
      "Hash": "False",
      "CADESType": "XLT1",
      "IsDetached": "True",
      "OriginalDocument": "T2gsIGh1bGxvIHRoZXJlIQ==",
      "CmsSignatureType": "Cosign",
      "TSPAddress": "https://dss-ca20-w12r2/tsp/tsp.srf"
    },
    "CertificateId": 2,
    "PinCode": ""
  }
}
```

Примечание

При создании соподписи в поле **Content** необходимо передавать подписанный документ, для которого требуется соподпись.

В поле **OriginalDocument** необходимо передавать содержимое исходного документа.

В запросе должен присутствовать параметр **CmsSignatureType**, имеющий значение **cosign**.

Пример ответа

Сервис Подписи вернет закодированное в Base64 содержимое файла с электронной подписью.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
"MIIFtwYJKoZIhvcNAQcCoIIFqDCCBaQCAQEhDjAMBggMASGCSqGSIb3DQ ..."
```

Соподпись хэш-значения документа

Пример запроса

В примере осуществляется соподпись хэш-значения документа с использованием второго сертификата ключа проверки электронной подписи.

```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Expect: 100-continue
```

```
{
  "Content": "MIIPwYJKoZIhvcNAQcCoIIImDCCCJQCAQExDDAKBgYqh ...",
  "Signature": {
    "Type": "CADES",
    "Parameters": {
      "Hash": "True",
      "CADESType": "XLT1",
      "IsDetached": "True",
      "OriginalDocument": "mM40LMXbEv6DbpJe+ov8eCLRFHjglAZ22b3YebKmyU0=",
      "CmsSignatureType": "Cosign",
      "TSPAddress": "https://dss-ca20-w12r2/tsp/tsp.srf"
    },
    "CertificateId": 2,
    "PinCode": ""
  }
}
```

Примечание

При создании соподписи в поле `Content` необходимо передавать подписанный документ, для которого требуется соподпись.

В поле `OriginalDocument` необходимо передавать хэш-значение исходного документа.

В запросе должен присутствовать параметр `CmsSignatureType`, имеющий значение `cosign`.

Пример ответа

Сервис Подписи вернет закодированное в Base64 содержимое файла с электронной подписью.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
"MIIFtwYJKoZIhvcNAQcCoIIFqDCCBaQCAQExDjAMBggMASGCSqGSib3DQ ..."
```

Заверяющая подпись

Пример запроса

В примере формируется заверяющая подпись документа.

```

POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Expect: 100-continue

{
  "Content": "MIIIpwYJKoZIhvcNAQcCoIIImDCCCJQCAQExDDAKBgYqhQMCagkF ...",
  "Signature":
  {
    "Type": "CADES",
    "Parameters":
    {
      "Hash": "False",
      "CADESType": "XLT1",
      "CmsSignatureType": "Countersign",
      "TSPAddress": "https://dss-ca20-w12r2/tsp/tsp.srf"
    },
    "CertificateId": 2,
    "PinCode": ""
  }
}

```

Примечание

При создании заверяющей подписи в поле **Content** необходимо передавать подписанное сообщение, содержащее подпись, которую требуется заверить.

В запросе должен присутствовать параметр **CmsSignatureType**, имеющий значение **Countersign**. > В запросе может присутствовать параметр **SignatureIndex**, указывающий индекс заверяемой подписи. В случае, если данный параметр отсутствует в запросе, Сервис Подписи будет заверять первую подпись в документе.

Пример ответа

Сервис Подписи вернет закодированное в Base64 содержимое файла с электронной подписью.

```

HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
"MIIFtWYJKoZIhvcNAQcCoIIFqDCCBaQCAQExDjAMBggMAsgCSqGSib3DQ ..."

```

Подпись в формате XMLDSig

REST-интерфейс Сервиса Подписи позволяет формировать XML-подпись следующими способами:

- [Enveloping](#)
- [Enveloped](#)
- [Подпись по шаблону \(Template\)](#)

Enveloping XML-подпись

Результатом выполнения операции создания Enveloping-подписи является XML-документ, который состоит из электронной подписи, содержащей исходный документ в элементе **Object**

Пример запроса

В примере создается Enveloping XML-подпись.

```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Expect: 100-continue

{
  "Content": "PHJvb3Q+PHZhbHVlPnNhbXBsZTwvdmFsdWU+PC9yb290Pg==",
  "Signature": {
    "Type": "XMLDSig",
    "Parameters": {
      "XMLDSigType": "XMLEnveloping"
    },
    "CertificateId": 1,
    "PinCode": ""
  }
}
```

Пример ответа

Сервис Подписи вернет закодированное в Base64 содержимое файла с электронной подписью.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
"PFNpZ25hdHVyZSB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMj..."
```

Пример результирующего файла с электронной подписью: [enveloping.xml](#).

Enveloped XML-подпись

Результатом выполнения операции создания Enveloped-подписи является XML-документ, который представляет из себя подписываемый документ с включенным в него узлом электронной подписи `Signature`.

Пример запроса

В примере создается Enveloped XML-подпись.

```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Expect: 100-continue

{
  "Content": "PHJvb3Q+PHZhbHVlPnNhbXBsZTwvdmFsdWU+PC9yb290Pg==",
  "Signature": {
    "Type": "XMLDSig",
    "Parameters": {
      "XMLDSigType": "XMLEnveloped"
    },
    "CertificateId": 1,
    "PinCode": ""
  }
}
```

Пример ответа

Сервис Подписи вернет закодированное в Base64 содержимое файла с электронной подписью.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
"PFNPZ25hdHVyZSB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMj ..."
```

Пример результирующего файла с электронной подписью: [enveloped.xml](#).

XML-подпись по шаблону

XML-подпись по шаблону применяется в том случае, когда необходимо сформировать подпись не всего документа, а одного или нескольких отдельных узлов в документе.

При формировании XML-подписи по шаблону исходный документ уже включает в себя узлы, содержащие информацию об электронной подписи. В процессе формирования подписи Сервис вычисляет ее значение и подставляет его в исходный документ.

Данная статья содержит примеры следующих шаблонов XML-подписи:

- XML-подпись по стандарту **ГОСТ Р 34.10-2001 + ГОСТ Р 34.11-94**
- XML-подпись по стандарту **ГОСТ Р 34.10-2012 (256) + ГОСТ Р 34.11-2012 (256)**
- XML-подпись по стандарту **ГОСТ Р 34.10-2012 (512) + ГОСТ Р 34.11-2012 (512)**

XML-подпись по стандарту ГОСТ Р 34.10-2001 + ГОСТ Р 34.11-94

Здесь приводится пример шаблона для формирования подписи по шаблону для алгоритма подписи ГОСТ Р 34.10-2001

```
<root>
  <data id="dtbs">2BSigned</data>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="sigID1">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      <ds:SignatureMethod Algorithm="urn:ietf:params:xml:ns:cpxmlsec:algorithms:gostr34102001-
gostr3411" />
      <ds:Reference URI="#dtbs">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
          <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#gostr34116" />
        <ds:DigestValue/>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue/>
    <ds:KeyInfo/>
  </ds:Signature>
</root>
```

Примечание

Идентификаторы узлов, подлежащих подписи, необходимо указывать посредством атрибута `URI` узла `Reference`

Примечание

Идентификатор алгоритма подписи:

```
urn:ietf:params:xml:ns:cpxmlsec:algorithms:gostr34102001-gostr3411
```

Идентификатор алгоритма хэширования: `http://www.w3.org/2001/04/xmldsig-more#gostr3411`

Пример запроса

В примере создается XML-подпись по шаблону с использованием алгоритма ГОСТ Р 34.10-2001.

```

POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Expect: 100-continue

{
  "Content": "PHJvb3Q+PHZhbHVlPnNhbXBsZTwvdmFsdWU+PC9yb290Pg==",
  "Signature": {
    "Type": "XMLDSig",
    "Parameters": {
      "XMLDsigType": "XMLTemplate"
    },
    "CertificateId": 1,
    "PinCode": ""
  }
}

```

Пример ответа

Сервис Подписи вернет закодированное в Base64 содержимое файла с электронной подписью.

```

HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
"PFNpZ25hdHVyZSB4bWuc20iaHR0cDovL3d3dy53My5vcmcvMj... "

```

Пример результирующего файла с электронной подписью: [template34102001.xml](#).

XML-подпись по стандарту ГОСТ Р 34.10-2012 (256) + ГОСТ Р 34.11-2012 (256)

Здесь приводится пример шаблона для формирования подписи по шаблону для алгоритма подписи ГОСТ Р 34.10-2012 (256)

```

<root>
  <data id="dtbs">2BSigned</data>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="sigID1">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      <ds:SignatureMethod Algorithm="urn:ietf:params:xml:ns:cpxmlsec:algorithms:gostr34102012-
gostr34112012-256" />
      <ds:Reference URI="#dtbs">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
          <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </ds:Transforms>
        <ds:DigestMethod Algorithm="urn:ietf:params:xml:ns:cpxmlsec:algorithms:gostr34112012-256" />
        <ds:DigestValue/>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue/>
    <ds:KeyInfo/>
  </ds:Signature>
</root>

```

Примечание

Идентификаторы узлов, подлежащих подписи, необходимо указывать посредством атрибута `URI` узла `Reference`

Примечание

Идентификатор алгоритма подписи:

Пример запроса

В примере создается XML-подпись по шаблону с использованием алгоритма ГОСТ Р 34.10-2012 (256).

```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh...8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Expect: 100-continue

{
  "Content": "PHJvb3Q+PHZhbHVlPnNhbXBsZTwvdmFsdWU+PC9yb290Pg==",
  "Signature": {
    "Type": "XMLDSig",
    "Parameters": {
      "XMLDSigType": "XMLTemplate"
    },
    "CertificateId": 1,
    "PinCode": ""
  }
}
```

Пример ответа

Сервис Подписи вернет закодированное в Base64 содержимое файла с электронной подписью.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
"PFNpZ25hdHVyZSB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMj..."
```

Пример результирующего файла с электронной подписью: [template34102012-256.xml](#).

XML-подпись по стандарту ГОСТ Р 34.10-2012 (512) + ГОСТ Р 34.11-2012 (512)

Здесь приводится пример шаблона для формирования подписи по шаблону для алгоритма подписи ГОСТ Р 34.10-2012 (512)

```
<root>
  <data id="dtbs">2BSigned</data>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="sigID1">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      <ds:SignatureMethod Algorithm="urn:ietf:params:xml:ns:cpxmlsec:algorithms:gostr34102012-gostr34112012-512" />
      <ds:Reference URI="#dtbs">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
          <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </ds:Transforms>
        <ds:DigestMethod Algorithm="urn:ietf:params:xml:ns:cpxmlsec:algorithms:gostr34112012-512" />
        <ds:DigestValue/>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue/>
    <ds:KeyInfo/>
  </ds:Signature>
</root>
```

Примечание

Идентификаторы узлов, подлежащих подписи, необходимо указывать посредством атрибута `URI` узла `Reference`

Примечание

Идентификатор алгоритма подписи:

```
urn:ietf:params:xml:ns:cpxmlsec:algorithms:gostr34102012-gostr34112012-512
```

Идентификатор алгоритма хэширования: `urn:ietf:params:xml:ns:cpxmlsec:algorithms:gostr34112012-512`

Пример запроса

В примере создается XML-подпись по шаблону с использованием алгоритма ГОСТ Р 34.10-2012 (512).

```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Expect: 100-continue

{
  "Content": "PHJvb3Q+PHZhbHVlPnNhbXBsZTwvdmFsdWU+PC9yb290Pg==",
  "Signature": {
    "Type": "XMLDSig",
    "Parameters": {
      "XMLDSigType": "XMLTemplate"
    },
    "CertificateId": 1,
    "PinCode": ""
  }
}
```

Пример ответа

Сервис Подписи вернет закодированное в Base64 содержимое файла с электронной подписью.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
"PFNpZ25hdHVyZSB4bWxuc20iaHR0cDovL3d3dy53My5vcmcvMj..."
```

Пример результирующего файла с электронной подписью: [template34102012-512.xml](#).

Подпись документов MSOffice

Сервис Подписи позволяет формировать электронную подпись документов, созданных в офисном пакете Microsoft Office.

Для того, чтобы Сервис Подписи осуществил формирование подписи документа MS Office, в параметре `SignatureType` необходимо передавать значение `MSOffice`.

Пример запроса к Сервису Подписи на формирование подписи документа представлен далее:

Пример запроса

В примере формируется подпись Word-документа, переданного в параметре `Content`.

Примечание

В поле `Content` необходимо передавать закодированное в BASE64 бинарное представление документа

```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Expect: 100-continue
```

```
{
  "Content": "UESDBBQAB...",
  "Signature": {
    "Type": "MSOffice",
    "Parameters": {},
    "CertificateId": 1006,
    "PinCode": ""
  }
}
```

Пример ответа

Сервис Подписи вернет закодированные в Base64 двоичные данные подписанного документа.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
"PFNpZ25hdHVyZSB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMj..."
```

Подпись PDF-документов

Сервис Подписи позволяет формировать электронную подпись PDF-документов.

Для того, чтобы Сервис Подписи осуществил формирование подписи PDF-документа, в параметре `SignatureType` необходимо передавать значение `PDF`.

Пример запроса к Сервису Подписи на формирование подписи PDF-документа представлен далее:

Пример запроса

В примере формируется подпись PDF-документа, переданного в параметре `Content`.

Примечание

В поле `Content` необходимо передавать закодированное в BASE64 бинарное представление документа

```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Expect: 100-continue
```

```
{
  {
    "Content": "JVBERi0xLjUN...",
    "Signature": {
      "Type": "PDF",
      "Parameters": {
        "PDFFormat": "CMS"
      },
      "CertificateId": 1006,
      "PinCode": ""
    }
  }
}
```

Пример ответа

Сервис Подписи вернет закодированные в Base64 двоичные данные подписанного PDF-документа.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
"PFNpZ25hdHVyZSB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMj ..."
```

Отображаемая подпись PDF-документов

СЭП КриптоПро DSS позволяет добавлять в PDF-документы видимую (отображаемую) подпись. Видимая подпись может служить как подтверждение факта подписания электронного документа при его печати.

Шаблоны подписи

Для того чтобы поставить видимую подпись в электронный PDF-документ необходимо передать описание представления подписи в запрос на формирование подписи. Представление подписи (в терминологии PDF «signature appearance») можно описать шаблоном подписи, содержащим параметры данного представления.

СЭП КриптоПро DSS позволяет добавлять три вида шаблонов видимой подписи:

- [Простой текстовый шаблон](#)
- [Шаблон с логотипом и текстом](#)
- [Шаблон в виде изображения](#)

Общие параметры шаблонов

Размеры всех элементов в представлении подписи и в шаблоне указываются в единицах измерения – типографских пунктах Adobe (points). 1 пункт = 1/72 дюйма = 0,3528 мм.

Цвет элементов указывается в системе RGB (red, green, blue; красный, зелёный, синий).

Положение представления подписи на странице задаётся в системе координат PDF документа. Точка с координатами (0, 0) соответствует левому нижнему углу страницы.

ColorDescription

Описание цвета элемента. Описание полей приведено в таблице ниже.

поле	тип	описание
Red	int	Значение красной компоненты цвета
Green	int	Значение зелёной компоненты цвета
Blue	int	Значение синей компоненты цвета

SignatureRect

Описание прямоугольника подписи. Описание полей приведено в таблице ниже.

поле	тип	описание
LowerLeftX	int	X координата левого нижнего угла прямоугольника
LowerLeftY	int	Y координата левого нижнего угла прямоугольника
UpperRightX	int	X координата правого верхнего угла прямоугольника
UpperRightY	int	Y координата правого верхнего угла прямоугольника
BorderRadius	int	Радиус скругления углов прямоугольника
BorderWeight	int	Толщина линии границы прямоугольника. 0 – отсутствие границы (значение по умолчанию)

ПОЛЕ	ТИП	ОПИСАНИЕ
BorderColor	ColorDescription	Цвет границы прямоугольника. По умолчанию (0, 0, 0)
BackgroundColor	ColorDescription	Цвет фона прямоугольника. По умолчанию (255, 255, 255)
ContentMargin	int	Отступ от границы прямоугольника до содержимого представления подписи

FontDescription

Описание шрифта. Описание полей класса приведено ниже.

ПОЛЕ	ТИП	ОПИСАНИЕ
FontSize	int	Размер шрифта
FontFamily	int	Название шрифта. Допустимые значения: times, arial По умолчанию times
FontColor	ColorDescription	Цвет шрифта

Textblock

Описание блока текста. Блок текста – это набор символов, заканчивающийся переводом строки (абзац). Описание полей класса приведено ниже.

ПОЛЕ	ТИП	ОПИСАНИЕ
Text	string	Содержимое блока
Font	FontDescription	Описание шрифта
Margin	int	Отступ от границы прямоугольника до текста.

ImageBlock

Описание изображения – логотипа или фона прямоугольника подписи. Для задания изображения в качестве фона необходимо указать только поле Image.

ПОЛЕ	ТИП	ОПИСАНИЕ
Image	string	Байты изображения, закодированное в Base64. Поддерживаемые форматы изображений: JPEG, JPEG2000, GIF, PNG, BMP, WMF, TIFF, CCITT, JBIG2.
LowerLeftX	int	X координата левого нижнего угла прямоугольника. Только для описания логотипа.
LowerLeftY	int	Y координата левого нижнего угла прямоугольника. Только для описания логотипа.
Scale	Int	Масштаб изображения. Только для описания логотипа.

Простой текстовый шаблон

Данный шаблон является наиболее простым в конфигурации и используется в случае, когда отображаемая подпись включает в себя только текстовые данные.

Конфигурация шаблона включает в себя следующие поля:

ПОЛЕ	ТИП	ОПИСАНИЕ
Content	TextBlock[]	Содержимое представления в виде массива текстовых блоков
Page	int	Номер страницы (первая страница имеет номер 1), на которой следует расположить представление. Отрицательные значения соответствуют отсчету от последней страницы. -1 указывает на последнюю страницу
Rect	SignatureRect	Описание прямоугольника подписи.
TemplateId	int	Идентификатор шаблона. Должен иметь значение 1.

Шаблон с логотипом и текстом

Данный шаблон может включать в себя графическое изображение (например, логотип компании) вдобавок к текстовой информации, описывающей подписываемые данные.

Шаблон включает в себя следующие поля:

ПОЛЕ	ТИП	ОПИСАНИЕ
Content	TextBlock[]	Содержимое представления в виде массива текстовых блоков
Page	int	Номер страницы (первая страница имеет номер 1), на которой следует расположить представление. Отрицательные значения соответствуют отсчету от последней страницы. -1 указывает на последнюю страницу
Icon	ImageBlock	Описание параметров логотипа.
Rect	SignatureRect	Описание прямоугольника подписи.
TemplateId	int	Идентификатор шаблона. Должен иметь значение 2.

Шаблон в виде изображения

Данный шаблон позволяет использовать цельное изображение в качестве отображаемой подписи

ПОЛЕ	ТИП	ОПИСАНИЕ
Content	TextBlock[]	Содержимое представления в виде массива текстовых блоков
Page	int	Номер страницы (первая страница имеет номер 1), на которой следует расположить представление. Отрицательные значения соответствуют отсчету от последней страницы. -1 указывает на последнюю страницу
Background	ImageBlock	Описание параметров изображения.
Rect	SignatureRect	Описание прямоугольника подписи.
TemplateId	int	Идентификатор шаблона. Должен иметь значение 3.

Передача параметров шаблона в метод подписи документов

Если в документ необходимо поставить видимую подпись, то в дополнительные параметры следует положить два значения: первое – идентификатор шаблона в качестве значения параметра PdfSignatureTemplateId, второе – параметры шаблона в качестве значения параметра PdfSignatureAppearance.

Примечание

Возможные значения параметра PdfSignatureTemplateId зависят от вида шаблона отображаемой подписи. Поле TemplateId при использовании каждого из шаблонов должно иметь значение 1, 2 или 3 соответственно.

Примечание

Параметр шаблона отображаемой подписи Page может иметь как положительные, так и отрицательные значения. Отрицательные значения показывают, что количество страниц нужно отсчитывать от последней страницы документа в сторону начала. При этом значение -1 соответствует последней странице документа.

Содержимое шаблона должно быть представлено как json-объект, сериализованный в UTF-8 строку и закодированный в BASE64.

Пример запроса на подпись PDF-документа, включающий в себя отображаемую подпись в виде набора текстовых данных:

```
POST http://<hostname>/SignServer/rest/api/documents HTTP/1.1
Content-Type: application/json
Authorization: Bearer
User-Agent: PostmanRuntime/7.15.2
Accept: */*
Cache-Control: no-cache
Host: grand-pc
Cookie: ASP.NET_SessionId=maaczeeyn5tx02wjtwy41cfc;
__RequestVerificationToken_L1NUUw2=mfs0a7NJ4_N7PxmvjMjdv3B36CRqSzKaW00-
UV3o74Is1n2lJC06KtInVogpMvWiS6lkp5YI5jypYkgzn4g1YwEMX-mwwtNI4ZtYteIkoE1
Accept-Encoding: gzip, deflate
Content-Length: 74210
Connection: keep-alive

{
  "Content": "JVBERi0xL...",
  "Signature": {
    "Type": "PDF",
    "Parameters": {
      "PDFFormat": "CMS",
      "PdfSignatureAppearance": "eyJDb2...",
      "PdfSignatureTemplateId": "1"
    },
    "CertificateId": 1006,
    "PinCode": ""
  }
}
```

Примеры содержимого параметра PdfSignatureAppearance для различных шаблонов.

Простой текстовый шаблон (PdfSignatureTemplateId = 1)

```
{
  "Content": [{
    "Text": "Подлинник электронного документа, подписанного ЭП, хранится в системе элеткронного документооборота Минкомсвзязи России.",
    "Font": {
      "Name": "Arial",
      "Size": 12,
      "Style": "Normal"
    }
  }]
}
```



```

        "FontSize": 4,
        "FontFamily": "arial",
        "FontStyle": 0,
        "FontColor": {
            "Red": 0,
            "Green": 0,
            "Blue": 0
        }
    },
    {
        "Text": "СВЕДЕНИЯ О СЕРТИФИКАТЕ ЭП:",
        "Font": {
            "FontSize": 8,
            "FontFamily": "times",
            "FontStyle": 0,
            "FontColor": {
                "Red": 0,
                "Green": 0,
                "Blue": 0
            }
        }
    },
    {
        "Text": "Кому выдан: Иванов Иван Иванович",
        "Font": {
            "FontSize": 8,
            "FontFamily": "times",
            "FontStyle": 0,
            "FontColor": {
                "Red": 0,
                "Green": 0,
                "Blue": 0
            }
        }
    },
    {
        "Text": "Кем выдан: УЦ Минкомсвязи",
        "Font": {
            "FontSize": 8,
            "FontFamily": "times",
            "FontStyle": 0,
            "FontColor": {
                "Red": 0,
                "Green": 0,
                "Blue": 0
            }
        }
    },
    {
        "Text": "Действителен: с 12.12.2015 по 12.12.2016",
        "Font": {
            "FontSize": 8,
            "FontFamily": "times",
            "FontStyle": 0,
            "FontColor": {
                "Red": 0,
                "Green": 0,
                "Blue": 0
            }
        }
    }
}],
"TemplateId": 1,
"Rect": {
    "LowerLeftX": 215,
    "LowerLeftY": 10.

```

```

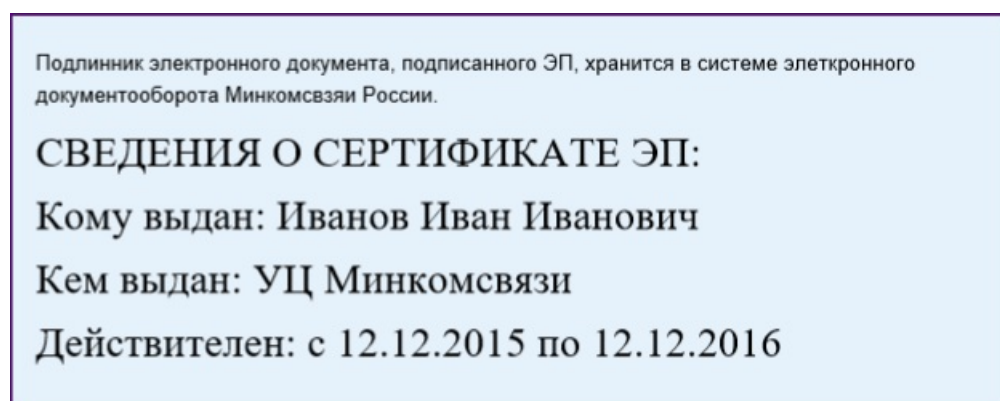
    "UpperRightX": 405,
    "UpperRightY": 85,
    "BorderRadius": 0,
    "BorderWeight": 1,
    "BorderColor": {
        "Red": 75,
        "Green": 13,
        "Blue": 100
    },
    "BackgroundColor": null,
    "ContentMargin": 5
},
"Page": 1
}

```

Base64-encoded содержимое шаблона может быть загружено здесь: [Base64-encoded](#)

Как видно из примера, шаблон состоит из прямоугольника подписи `Rect`, описывающего позицию и размеры штампа, а также набора текстовых блоков в поле `Content`, которые в этот штамп входят.

Результатом использования приведенного выше шаблона при формировании подписи является следующий штамп:



Шаблон с логотипом и текстом (PdfSignatureTemplateId = 2)

```

{
  "Content": [
    {
      "Text": "Подлинник электронного документа, подписанного ЭП, хранится в системе электронного документооборота Минкомсвязи России.",
      "Margin": 50,
      "Font": {
        "FontSize": 4,
        "FontFamily": "arial",
        "FontStyle": 0,
        "FontColor": {
          "Red": 0,
          "Green": 0,
          "Blue": 0
        }
      }
    },
    {
      "Text": "СВЕДЕНИЯ О СЕРТИФИКАТЕ ЭП",
      "Margin": 50,
      "Font": {
        "FontSize": 8,
        "FontFamily": "times",
        "FontStyle": 0,
        "FontColor": {
          "Red": 0,
          "Green": 0,
          "Blue": 0
        }
      }
    }
  ]
}

```

```

        "Green":0,
        "Blue":0
    }
}
},
{
    "Text":"Кому выдан: Иванов Иван Иванович",
    "Font":{
        "FontSize":8,
        "FontFamily":"times",
        "FontStyle":0,
        "FontColor":{
            "Red":0,
            "Green":0,
            "Blue":0
        }
    }
},
{
    "Text":"Кем выдан: УЦ Минкомсвязи",
    "Font":{
        "FontSize":8,
        "FontFamily":"times",
        "FontStyle":0,
        "FontColor":{
            "Red":0,
            "Green":0,
            "Blue":0
        }
    }
},
{
    "Text":"Действителен: с 12.12.2015 по 12.12.2016",
    "Font":{
        "FontSize":8,
        "FontFamily":"times",
        "FontStyle":0,
        "FontColor":{
            "Red":0,
            "Green":0,
            "Blue":0
        }
    }
}
],
"TemplateId":2,
"Icon":{
    "Image":"ivBORw0KGgo...",
    "LowerLeftX":null,
    "LowerLeftY":50,
    "Scale":40
},
"Rect":{
    "LowerLeftX":215,
    "LowerLeftY":10,
    "UpperRightX":405,
    "UpperRightY":85,
    "BorderRadius":0,
    "BorderWeight":1,
    "BorderColor":{
        "Red":75,
        "Green":13,
        "Blue":100
    },
    "BackgroundColor":null,
    "ContentMargin":5
}

```

```

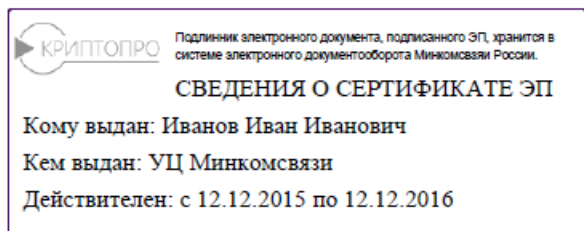
    },
    "Page":1
}

```

Base64-encoded содержимое шаблона может быть загружено здесь: [Base64-encoded](#)

Данный шаблон помимо текстовых блоков, которые были приведены в предыдущем примере, включает в себя блок изображения `Icon`, описывающий логотип, который включается в штамп подписи.

Результатом использования приведенного выше шаблона при формировании подписи является следующий штамп:



Шаблон в виде изображения (PdfSignatureTemplateId = 3)

```

{
  "Background":{
    "Image":"iVBORw0KGg..."
  },
  "TemplateId":3,
  "Rect":{
    "LowerLeftX":10,
    "LowerLeftY":10,
    "UpperRightX":194,
    "UpperRightY":86
  },
  "Page":1
}

```

Base64-encoded содержимое шаблона может быть загружено здесь: [Base64-encoded](#)

В данном примере в качестве штампа используется логотип компании, использованный в прошлом примере. Изображение представляет из себя `ImageBlock` в поле `Background`, которое будет размещено на всей площади штампа, размеры которого описываются полем `Rect`.

Результатом использования приведенного выше шаблона при формировании подписи является следующий штамп:



Асинхронная подпись

REST API v2 Сервиса подписи поддерживает режим "асинхронной подписи".

Примечание

"Асинхронная подпись" поддерживается только для операции подписи, выполняемой с подтверждением (myDSS, DSS SDK, OTP-via-SMS, OTP-via-Email и т.п).

В режиме "асинхронной подписи" отправит Callback в вызывающую систему после того как пользователь подтвердит подписание, документы будут подписаны и загружены в хранилище документов.

После получения Callback вызывающая система может выгрузить подписанные документы из хранилища документов.

Настройка асинхронной подписи

Включение режима асинхронной подписи

По умолчанию режим асинхронной подписи отключен на сервисе подписи. Для его включения необходимо выполнить команду в консоли PowerShell

```
Enable-DssAsyncOperationSettings
```

Настройка модуля для отправки Callback

Для регистрации модуля необходимо выполнить следующий скрипт:

```
#Настройка транспортного плагина
$plugin = Add-DssSignServerPlugin -PluginTypeName
"CryptoPro.DSS.Identity.Authentication.Notification.HttpTransportPlugin,CryptoPro.DSS.Identity.Authentication
.Notification" -PluginType AuthenticationResult -Settings @{}

#Добавление модуля оповещения
Add-DssSignServerNotifier -TransportPluginID $plugin.ID -NotifierType AuthenticationResultCallback -Settings
@{} -Type
"CryptoPro.DSS.Identity.Authentication.Notification.AuthenticationResultNotifier,CryptoPro.DSS.Identity.Authe
ntication.Notification"
```

При успешной регистрации в списке зарегистрированных модулей оповещения Сервиса Подписи будет модуль с типом `AuthenticationResultCallback`:

```
Get-DssSignServerNotifier

ID           : 6
Type         : AuthenticationResultCallback
Settings     : {[ThreadCount, 1]}
IsEnabled    : True
TransportPlugin : CryptoPro.DSS.PowerShell.Common.Objects.Plugin
MessagePlugin  :
```

Создание подписи в асинхронном режиме

Последовательность шагов при подтверждении операции подписи в асинхронном режиме:

1. [Аутентификация пользователя на Центре Идентификации](#)
2. [Загрузка документов для подписи](#)
3. [Создание операции подписи на Сервисе Подписи](#)
4. [Отправка запроса на подтверждение операции подписи](#)
5. Ожидание подтверждения операции в DSS SDK и подписи документов
6. [Выгрузка подписанного документа](#)

На шаге 3 для создания подписи с подтверждением в асинхронном режиме необходимо в вызове [Signature](#) передать дополнительные параметры:

- `IsAsync` - true
- `Callback` - URL-адрес для оповещения о завершении операции подписи

Пример запроса

```
{
  "BinaryData": [{
    "RefId" : "37bcc23f-73d8-4cb9-8b54-9b47ad03fd52"
  }],
  "Signature" :
  {
    "CertificateId": "13",
    "ProcessingTemplateId": "1"
  },
  "IsAsync": "true",
  "Callback": "http://hostname/signresult"
}
```

На шаге 5 Сервис Подписи отправит Callback следующего формата:

```
{
  "Operation": {
    "Id": "6ec9d54a-7f4d-4579-abe9-660ce241e185",
    "Result": {
      "ProcessedDocuments": [
        {
          "RefId": "f7dbd740-9fad-4aea-88a7-39dfcc81c0ab",
          "OriginalRefId": "37bcc23f-73d8-4cb9-8b54-9b47ad03fd52",
          "Content": null,
          "Status": "Completed",
          "Error": null,
          "ErrorDescription": null
        }
      ]
    },
    "Status": "Completed",
    "Error": null,
    "ErrorDescription": null,
    "ExpirationDate": 1582121878
  }
}
```

Список методов Сервиса Подписи

Базовый адрес REST Сервиса Подписи:

```
https://<hostname>/<AppName>/rest/api
```

где

<hostname> - DNS-имя сервера DSS

<AppName> - имя Веб-приложения Сервиса подписи. По умолчанию - SignServer

Примечание
Аутентификация на Сервисе Подписи **только** по OAuth-токену, выпущенному на Центре Идентификации.

Примечание
В примерах ниже будут использованы следующие значения
<hostname> - dss.cryptopro.ru
<AppName> - SignServer

Полное описание всех методов конечных точек по категориям:

- [Policy](#)
- [Requests](#)
- [Certificates](#)
- [Documents](#)
- [Transaction](#)
- [Operations](#)
- [Signature](#)
- [Keys](#)

Краткое описание всех методов конечных точек приводится ниже.

Разное:

МЕТОД	ОПИСАНИЕ
/policy	Получение настроек Сервиса Подписи (Политики Сервиса Подписи)
/transactions	Создание транзакций (выполнение операций на Сервисе Подписи с подтверждением вторым фактором аутентификации)

Запросы на сертификаты:

ТИП	ОПИСАНИЕ
/requests	Создание запроса на сертификат
/requests	Получение списка запросов на сертификаты
/requests/{key}	Получение запроса на сертификат по идентификатору
/requests/{key}	Удаление запроса на сертификат
/requests/{key}/content	Получение содержимого запроса на сертификат

тип	описание
/requests/revokerequests/{key}/content	Получение содержимого запроса на отзыв/приостановление/восстановление
/requests/{key}/status	Одобрение и отклонение запроса на сертификат
/requests/revokerequests/{key}/status	Одобрение и отклонение запроса на отзыв/приостановление/восстановление

Сертификаты:

тип	описание
/certificates	Получение списка сертификатов
/certificates/{cert_id}	Получение сертификата по идентификатору
/certificates/{cert_id}	Удаление сертификата по идентификатору
/certificates/{cert_id}/content	Получение содержимого сертификата
/certificates/{cert_id}/revokerequests	Получение списка запросов на отзыв/приостановление/восстановление
/certificates/{cert_id}/pin	Смена ПИН-кода
/certificates/{cert_id}/default	Назначение сертификата по умолчанию
/certificates/{cert_id}/friendlyname	Назначение дружественного имени
/certificates/{cert_id}/status	Отзыв
/certificates/{cert_id}/status	Приостановление
/certificates/{cert_id}/status	Восстановление
/certificates	Установка сертификата

Подпись:

тип	описание
/documents	Отправка документа на подпись, зашифрование, расшифрование
/documents/packagesignature	Отправка пакета документов на подпись
/documents/encrypt	Зашифрование документа
/documents/decrypt	Расшифрование документа
/documents/decrypt/parse	Разбор зашифрованного документа
/documents/enhancesignature	Усовершенствование подписи

Получение настроек Сервиса Подписи (Политики Сервиса Подписи)

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://dss.cryptopro.ru/SignServer/rest/api/policy
Параметры	Метод не имеет параметров
Возвращаемое значение	DSSPolicy - Политика Сервиса Подписи

Описание

Конечная точка предназначена для получения настроек Сервиса Подписи:

- Список подключенных Удостоверяющих Центров
- Список подключенных криптопровайдеров
- Список разрешённых форматов подписи
- Требования к ПИН-коду
- Требование к подтверждению операций
- Список TSP-служб

Примечание

Рекомендуется вызывать метод /policy один раз при начале взаимодействия с Сервисом Подписи. Данные Политики Сервиса Подписи позволят настроить интерфейс и логику работы интегрируемой с DSS системы.

Списки подключенных Удостоверяющих Центров и криптопровайдеров определяют параметры создания запроса на сертификат (/requests). Подробнее о выпуске сертификата смотреть [здесь](#).

Также список подключенных Удостоверяющих Центров позволит определить тип и параметры УЦ на котором был выпущен сертификат пользователя, обработан запрос на сертификат пользователя. Тип Удостоверяющего Центра определяет набор действий который можно выполнить с запросом на сертификат и сертификатом пользователя.

Требования к подтверждению операций определяют список операций Сервиса Подписи требующих двух факторной аутентификации.

Список разрешённых форматов подписи определяет типы подписи, которые можно создавать на Сервисе Подписи.

В списке TSP-служб перечислены разрешённые адреса TSP служб, которые используются при создании подписи формата CAdES-T, CAdES-X Long Type 1 или при усовершенствовании подписи до форматов CAdES-T, CAdES-X Long Type 1.

Требования к ПИН-коду определяют необходимость задания ПИН-кода на закрытый ключ пользователя. Политикой Сервиса Подписи не задаются требования к сложности ПИН-кода.

Внимание!

Если Политика Сервиса Подписи содержит пустой список Удостоверяющих Центров или криптопровайдеров, то необходимо обратиться к Администратору DSS. Это свидетельствует о том что на сервере DSS нет связи с Удостоверяющим Центром и/или с криптопровайдером (КриптоПро HSM). Создание запросов на сертификата и/или создание подписи будет невозможно.

Конечная точка Requests

Конечная точка requests предоставляет следующие методы:

- [Создание запроса на сертификат](#)
- [Получение списка запросов на сертификат](#)
- [Получение запроса на сертификат по идентификатору](#)
- [Удаление запроса на сертификат по идентификатору](#)
- [Получение содержимого запроса на сертификат](#)
- [Одобрение и отклонение запроса на сертификат](#)
- [Одобрение и отклонение запроса на отзыв/приостановление/восстановление](#)

Создание запроса на сертификат

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://dss.cryptopro.ru/SignServer/rest/api/requests
Параметры	CertificateRequest - Запрос на создание сертификата
Возвращаемое значение	DSSCertRequest - Созданный запрос на сертификат

Подробнее о выпуске запроса на сертификат смотреть раздел: [Выпуск сертификата](#)

Получение списка запросов на сертификат

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://dss.cryptopro.ru/SignServer/rest/api/requests
Параметры	Метод не имеет параметров
Возвращаемое значение	List< DSSCertRequest > - Список созданных запросов на сертификат

Примечание

Для получения списка запросов на отзыв/приостановление/восстановление используйте метод [/certificates/{key}/revokerequests](#). Список запросов на отзыв, приостановление и восстановление сертификата может быть получен только для сертификата выпущенного через модуль интеграции с КриптоПро УЦ 2.0.

Пример запроса

```
GET https://host/SignServer/rest/api/requests HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV... eFfe5CkkuKMQ
```

Пример ответа

HTTP/1.1 200 OK
Content-Length: 1379
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5

```
[{
  "CertificateType": "ServerSide",
  "Base64Request": "MIIBKTCB2QIBADARMQ8wDQ ... h7a6eLJ9axIe0vqaqhtWw==",
  "CertificateAuthorityID": 11,
  "CADisplayName": null,
  "DistName": "CN=idonly",
  "Subject": "idonly",
  "Status": "ACCEPTED",
  "ID": 23,
  "CARequestID": null,
  "CertificateID": 14,
  "RequestType": "Certificate",
  "GroupID": "e8e67f9e-7eed-4116-ad98-20582e4d766e"
}, {
  "CertificateType": "ServerSide",
  "Base64Request": "MIIBKTCB2QIBADARMQ8wDQ ... BthH+U20bfd9sbjGtE47YDeW+Ng==",
  "CertificateAuthorityID": 11,
  "CADisplayName": null,
  "DistName": "CN=idonly",
  "Subject": "idonly",
  "Status": "PENDING",
  "ID": 24,
  "CARequestID": null,
  "CertificateID": 0,
  "RequestType": "Certificate",
  "GroupID": "e8e67f9e-7eed-4116-ad98-20582e4d766e"
}]
```

Получение запроса на сертификат по идентификатору

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://dss.cryptopro.ru/SignServer/rest/api/requests/{req_id}
Параметры	Идентификатор запроса на сертификат
Возвращаемое значение	DSSCertRequest - запрос на сертификат

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404		Запрос на сертификат с указанным ID не найден

Пример запроса

```
GET https://host/SignServer/rest/api/requests/24 HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJK ... ZIF89F4cmHc6kFM_QQXQ
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Length: 687
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5

{
  "CertificateType": "ServerSide",
  "Base64Request": "MIIBKTCB2QIBADARMQ8wDQ ... Tx1BthH+U20bfd9sbjGtE47YDeW+Ng==",
  "CertificateAuthorityID": 11,
  "CADisplayName": null,
  "DistName": "CN=idonly",
  "Subject": "idonly",
  "Status": "PENDING",
  "ID": 24,
  "CARequestID": null,
  "CertificateID": 0,
  "RequestType": "Certificate",
  "GroupID": "e8e67f9e-7eed-4116-ad98-20582e4d766e"
}
```

Удаление запроса на сертификат по идентификатору

ПАРАМЕТР		ЗНАЧЕНИЕ
HTTP-метод		DELETE
Путь		https://dss.cryptopro.ru/SignServer/rest/api/requests/{req_id}
Параметры		Идентификатор запроса на сертификат
Возвращаемое значение		Метод не имеет возвращаемого значения
HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404		Запрос на сертификат с указанным ID не найден

Пример запроса

```
DELETE https://host/SignServer/rest/api/requests/27 HTTP/1.1
Authorization: Bearer eyJ0eXAi ... RwkYRac7AJuuX2aeY1H7fjpQ
Host: host
Content-Length: 0
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Length: 0
Server: Microsoft-IIS/7.5
```

Получение содержимого запроса на сертификат

Метод предназначен для получения содержимого запроса на сертификат в различных представлениях:

- PKCS#10
- XML
- Печатное представление (HTML)

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://dss.cryptopro.ru/SignServer/rest/api/requests/{req_id}/content?format={format}&authorityId={ca_id}
Параметры	req_id - Идентификатор запроса на сертификат format - Формат представления содержимого запроса. ca_id - Идентификатор Удостоверяющего Центра
Возвращаемое значение	Содержимое запроса на сертификат

Описание форматов представления запроса на сертификат - [DSSCertificateFormatEnum](#)

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404		Запрос на сертификат с указанным ID не найден

Пример запроса

```
GET https://host/SignServer/rest/api/requests/24/content?format=4&authorityId=0 HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cGE6YWV0IjoiZjBkZDZlOUkZOG6yZj1CNQ
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Length: 4535
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5

"<?xml version='1.0' encoding='utf-16'>
<html xmlns:pki='http://www.cryptopro.ru/2001/Schema/WD1-PKI'>
  <head>
    <title>Запрос на сертификат X.509</title>
    <style>
      <!--\r\nH1\r\n{ \r\n\tfont-size: 10pt; \r\n\tfont-family: Verdana; \r\n\ttext-align: Center;
\r\n\tmargin: 0px \r\n}\r\nH2\r\n{ \r\n\tfont-size: 8pt; \r\n\tfont-family: Verdana; \r\n\ttext-align: Center;
\r\n\tmargin: 0px \r\n}\r\n.nfoot \r\n{ \r\n\tfont-size: 8pt; \r\n\tfont-family: Verdana; \r\n\ttext-align: Right \r\n}\r\n.note \r\n{ \r\n\tfont-size: 7pt; \r\n\tfont-family: Verdana; \r\n\ttext-decoration: underline; \r\n\tmargin: 0px \r\n}\r\nP\r\n{ \r\n  FONT-SIZE: 7pt;\r\n  MARGIN: 0px;\r\n  FONT-FAMILY: Verdana\r\n}\r\nPRE\r\n{ \r\n  FONT-SIZE: 7pt;\r\n  MARGIN: 0px;\r\n  FONT-FAMILY: Verdana\r\n}\r\n-->
    </style>
  </head>
  <body>
    <h1>
      <b>Наименование организации-Удостоверяющего Центра</b>
    </h1>
    <h2>
      <b>Запрос на сертификат ключа проверки электронной подписи</b>
    </h2>
    <hr />
    <div style='margin-left:1em'>
      <p>
        <b>Сведения о запросе на сертификат:</b>
      </p>
      <div style='margin-left:1em'>
```

```

        <p>
            <b>Кем выпущен:</b>
        </p>
        <div style="margin-left:1em">
            <p>idonly</p>
        </div>
    </div>
</div>
<div style="margin-left:1em">
    <p>
        <b>Версия:</b> 1\r\n      (0x0)\r\n    </p><p>
        <b>Субъект запроса на сертификат:</b>CN\r\n  =\r\n  idonly</p><p>
        <b>Ключ проверки электронной подписи:</b>
    </p>
    <div style="margin-left:1em">
        <p>Алгоритм ключа проверки электронной подписи:</p>
        <div style="margin-left:1em">
            <p>Название: \r\n      ГОСТ Р 34.10-2001</p>
            <div style="margin-left:1em">
                <p>Параметры:\r\n      30 12 06 07 2a 85 03 02 02 24 00 06 07 2a 85 03 02 02 1e
01</p>
            </div>
        </div><p>Значение: \r\n      04 40 16 5b c3 2c c2 81 2d 76 d7 16 cf 05 71 62 ef cf 2b de d2 bb
a4 8c 76 31 03 e9 3b c2 32 ca 7c 7c 61 ca 7f 88 b4 58 e1 81 f0 7a 74 33 59 15 67 10 4c 1d 57 88 7d 66 d9 d4
dc ea 5b 3b 97 09 e9 0a</p>
    </div><p>
        <b>Атрибуты запроса на сертификат X.509</b>
    </p>
    <div style="margin-left:1em">
        <p>Название: \r\n      Расширения сертификатов</p><p>
            <b>Расширения сертификата X.509</b>
        </p>
        <div style="margin-left:1em">
            <p>1.\r\n      Расширение </p>
            <div style="margin-left:1em">
                <p>Название: \r\n      Использование ключа</p><p>Значение: \r\n      Цифровая подпись,
Неотрекаемость, Шифрование ключей, Шифрование данных (f0)</p>
            </div><p>2.\r\n      Расширение </p>
            <div style="margin-left:1em">
                <p>Название: \r\n      Идентификатор ключа субъекта</p><p>Значение: \r\n      77 32 f2 d3
22 77 0d b6 72 06 d7 18 23 ad 2a 69 78 48 4c 78</p>
            </div><p>3.\r\n      Расширение </p>
            <div style="margin-left:1em">
                <p>Название: \r\n      Улучшенный ключ</p><p>Значение: \r\n      Пользователь Центра
Регистрации, HTTP, TLS клиент (1.2.643.2.2.34.6), Проверка подлинности клиента (1.3.6.1.5.5.7.3.2)</p>
            </div>
        </div><p>
            <b>Подпись запроса:</b>
        </p>
        <div style="margin-left:1em">
            <p>Алгоритм подписи:</p>
            <div style="margin-left:1em">
                <p>Название: \r\n      ГОСТ Р 34.11/34.10-2001</p>
                </div><p>Значение: \r\n      36 BE E5 0D D8 8E 13 AD 31 6E 6C DF 7D 9B 63 53 FE 11 B6 41 19
4F FF DA 31 4A 0C 27 EE 4A D2 81 EE B1 42 82 55 EF 5E DA 00 08 26 DC 88 35 7E AD 98 13 EC B0 CA 5D D9 1C 33
C1 B3 89 3F C8 07 DD</p>
            </div>
        </div>
    </div>
</div>
<hr />
<br />
<p class="foot">Подпись владельца запроса на сертификат: _____/_____</p>
<br />
<p class="foot">\ " ____ \ " _____ 20__ г.</p>
<br />

```

```
<p class=\ "note\ ">М. П.</p><p>средство электронной подписи \ криптопро CSP\</p>
<br />
<p class=\ "note\ ">Подписанный запрос на сертификат ключа проверки электронной подписи следует
переслать по адресу:</p><p class=\ "note\ ">111111, Москва, ул. XXXXXXXX, д.ХХ. XXXXXXXXXXXXXXXXXXXX</p><p
class=\ "note\ ">Администратору информационной безопасности.</p>
</body>
</html>"
```

Получение содержимого запроса на отзыв/приостановление/восстановление

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://dss.cryptopro.ru/SignServer/rest/api/requests/revokerequests/{req_id}/content?format={format}&authorityId={ca_id}
Параметры	req_id - Идентификатор запроса на сертификат format - Тип предоставления содержимого запроса. ca_id - Идентификатор Удостоверяющего Центра
Возвращаемое значение	Содержимое запроса

Описание форматов представления запроса на сертификат - [DSSCertificateFormatEnum](#)

Примечание

Список запрос на отзыв/приостановление/восстановление может быть получен только для сертификатов выпущенных через модуль интеграции с КриптоПро УЦ 2.0.

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404		Запрос на сертификат с указанным ID не найден

Одобрение и отклонение запроса на сертификат

Принятие/отклонение запроса на сертификат:

ПАРАМЕТР		ЗНАЧЕНИЕ
HTTP-метод		PATCH
Путь		https://dss.cryptopro.ru/SignServer/rest/api/requests/{req_id}/status
Параметры		req_id - Идентификатор запроса на сертификат RequestStatus - Новый статус запроса на сертификат.
Возвращаемое значение		Метод не имеет возвращаемого значения
HTTP-КОД	ОШИБКА	ОПИСАНИЕ
401	Authorization has been denied for this request	Пользователь не имеет разрешения на выполнение запроса

Примечание

Метод может быть вызван только пользователем с ролью Оператор DSS. Оператор DSS может принять или отклонить запросы на сертификаты созданные через модуль интеграции с КриптоПро УЦ 2.0. Запросы на сертификат созданные через Offline УЦ могут быть только отклонены.

При отклонении/одобрении запроса на сертификат необходимо заполнить только поля `Type` и `Value`:

- Поле `Type` имеет фиксированное значение **0** ([Certificate](#)).
- Поле `Value` может принимать значения 4 ([ACCEPTED](#)) или 8 ([REJECTED](#))

Пример запроса

```
POST https://host/SignServer/rest/api/requests/26/status HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiOiJY3m5uZc59QwnD0BA
Content-Type: application/json; charset=utf-8

Content-Length: 39
Expect: 100-continue

{"Value":8,"ID":"-1","caID":0,"Type":0}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Length: 0
Server: Microsoft-IIS/7.5
```

Принятие/отклонение запроса на отзыв/приостановление/восстановление:

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	PATCH
Путь	https://dss.cryptopro.ru/SignServer/rest/api/requests/revokerequests/{revreq_id}/status
Параметры	revreq_id - Идентификатор запроса на отзыв/приостановление/восстановление в УЦ RequestStatus - Новый статус запроса на сертификат.
Возвращаемое значение	Метод не имеет возвращаемого значения

В структуре [RequestStatus](#) необходимо заполнить поля:

- `Value` - может принимать значения **4** ([ACCEPTED](#)) или **8** ([REJECTED](#))
- `caID` - идентификатор Удостоверяющего Центра
- `Type` имеет фиксированное значение **0** ([Certificate](#))
- `Comment` (опциональное) - комментарий Оператора DSS. Например, причина отклонения запроса.

Значение `req_id` необходимо взять из результата работы метода [/certificates/{cert_id}/revokerequests](#) - поле `ID` в структуре [DSSRevRequest](#).

Последовательность действий Оператора DSS при одобрении/отклонении запросов на отзыв/приостановление/восстановление:

- получить список сертификатов пользователя [/certificates/](#)
- для каждого из сертификатов пользователя проверить наличие запросов на отзыв/приостановление/восстановление [/certificates/{cert_id}/revokerequests](#)
- если запрос [DSSRevRequest](#) имеет поле `Status` [PENDING](#), то одобрить или отклонить его

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
401	Authorization has been denied for this request	Пользователь не имеет разрешения на выполнение запроса
400	invalid_enroll	Неверно указан идентификатор УЦ
400	invalid_rev_request	Неверно указан идентификатор запроса

Примечание

Метод может быть вызван только пользователем с ролью Оператор DSS. Оператор DSS может принять или отклонить запросы на отзыв/приостановление/восстановление созданные через модуль интеграции с КриптоПро УЦ 2.0.

Пример запроса

```
POST https://host/SignServer/rest/api/requests/revokerequests/6a5c8b4a-52c6-e811-80db-00155d454d12/status
HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIHMD ... o9ju8CQuZneFapoL2q4CjUJAw
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 39
Expect: 100-continue

{"Value":4,"ID":"-1","caID":6,"Type":0}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Length: 0
Server: Microsoft-IIS/7.5
```

Конечная точка Certificates

Конечная точка certificates предоставляет следующие методы:

- [Получение списка сертификатов](#)
- [Получение сертификата по идентификатору](#)
- [Удаление сертификата по идентификатору](#)
- [Получение содержимого сертификата](#)
- [Получение списка запросов на отзыв/приостановление/восстановление](#)
- [Установка сертификата](#)

Изменение свойств сертификата:

- [Смена ПИН-кода](#)
- [Валидация ПИН-кода](#)
- [Назначение сертификата по умолчанию](#)
- [Назначение дружественного имени](#)

Изменение статуса сертификата:

- [Отзыв](#)
- [Приостановление](#)
- [Восстановление](#)

Получение списка сертификатов

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://dss.cryptopro.ru/SignServer/rest/api/certificates
Параметры	Метод не имеет параметров
Возвращаемое значение	List< DSSCertificateEx > - Список сертификатов

Пример запроса

```
GET https://host/SignServer/rest/api/certificates HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV... eFfe5CkkuKMQ
```

Пример ответа

HTTP/1.1 200 OK
Content-Length: 4272
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5

```
[{
  "CertificateType": "ServerSide",
  "ID": 14,
  "DName": "CN=idonly",
  "CertificateBase64": "MIIDCDCCA ... ugFV8td4DaneG2/gno7T6Alohp6CF/yOu",
  "Status": {
    "Value": "ACTIVE",
    "RevocationInfo": null,
    "PinCode": null,
    "ActiveCertId": 0
  },
  "IsDefault": false,
  "CertificateAuthorityID": 11,
  "CspID": "e8e67f9e-7eed-4116-ad98-20582e4d766e",
  "HashAlgorithms": ["GOST R 34.11-94"],
  "ProviderName": null,
  "ProviderType": 0,
  "PrivateKeyNotBefore": null,
  "PrivateKeyNotAfter": null,
  "HasPin": false,
  "FriendlyName": ""
}, {
  "CertificateType": "ServerSide",
  "ID": 15,
  "DName": "CN=idonly, C=RU",
  "CertificateBase64": "MIIG+TCCBq ... dJkhC/rkJrBYhT574WAMgGdxGQb1lQ==",
  "Status": {
    "Value": "ACTIVE",
    "RevocationInfo": null,
    "PinCode": null,
    "ActiveCertId": 0
  },
  "IsDefault": false,
  "CertificateAuthorityID": 6,
  "CspID": "e8e67f9e-7eed-4116-ad98-20582e4d766e",
  "HashAlgorithms": ["GOST R 34.11-94"],
  "ProviderName": null,
  "ProviderType": 0,
  "PrivateKeyNotBefore": null,
  "PrivateKeyNotAfter": null,
  "HasPin": false,
  "FriendlyName": ""
}
]
```

Получение сертификата по идентификатору

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://dss.cryptopro.ru/SignServer/rest/api/certificates/{cert_id}
Параметры	cert_id - идентификатор сертификата
Возвращаемое значение	DSSCertificateEx - Сертификат

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404		Сертификат с указанным ID не найден

Пример запроса

```
GET https://host/SignServer/rest/api/certificates/14 HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV... eFfe5CkkuKMQ
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Length: 1458
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5

{
  "CertificateType": "ServerSide",
  "ID": 14,
  "DName": "CN=idonly",
  "CertificateBase64": "MIIDCDCCAregAwIBAgITEgAtlI ... hYHzugFV8td4DaneG2/gno7T6Alohp6CF/yOu",
  "Status": {
    "Value": "ACTIVE",
    "RevocationInfo": null,
    "PinCode": null,
    "ActiveCertId": 0
  },
  "IsDefault": false,
  "CertificateAuthorityID": 11,
  "CspID": "e8e67f9e-7eed-4116-ad98-20582e4d766e",
  "HashAlgorithms": ["GOST R 34.11-94"],
  "ProviderName": null,
  "ProviderType": 0,
  "PrivateKeyNotBefore": null,
  "PrivateKeyNotAfter": null,
  "HasPin": false,
  "FriendlyName": ""
}
```

Удаление сертификата по идентификатору

ПАРАМЕТР		ЗНАЧЕНИЕ
HTTP-метод		DELETE
Путь		https://dss.cryptopro.ru/SignServer/rest/api/certificates/{cert_id}
Параметры		cert_id - Идентификатор сертификата
Возвращаемое значение		Метод не имеет возвращаемого значения
HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404		Сертификат с указанным ID не найден

Пример запроса

DELETE https://host/SignServer/rest/api/certificates/14 HTTP/1.1
Authorization: Bearer eyJ0eXAi ... RvwkYRac7AJuuX2aeY1H7fjpQ
Host: host
Content-Length: 0

Получение содержимого сертификата

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://dss.cryptopro.ru/SignServer/rest/api/certificates/{cert_id}/content?format={format}
Параметры	req_id - Идентификатор сертификата format - Формат представления содержимого сертификата.
Возвращаемое значение	Содержимое сертификата

Описание форматов представления сертификата - [DSSCertificateFormatEnum](#)

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404		Сертификат с указанным ID не найден

Пример запроса

GET https://host/SignServer/rest/api/certificates/14/content?format=2 HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLC ... LgtvjC0-2GA0Kth2ZnorIoiA
Host: host

Пример ответа

HTTP/1.1 200 OK
Content-Length: 5571
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5

```
<?xml version="1.0" encoding="utf-16"?>
<pki:certificate xmlns:pki="http://www.cryptopro.ru/2001/Schema/WD1-PKI">
  <pki:version pki:value="2">3</pki:version>
  <pki:serial-number>12002D94874E4BA91F129E408D0000002D9487</pki:serial-number>
  <pki:issuer pki:count="5">
    <pki:rdn pki:order="5" pki:count="1">
      <pki:rdn-attr pki:order="1" pki:id="2.5.4.3">
        <pki:name xml:lang="ru">CN</pki:name>
        <pki:value pki:lang="ru">CRYPTO-PRO Test Center 2</pki:value>
      </pki:rdn-attr>
    </pki:rdn>
    <pki:rdn pki:order="4" pki:count="1">
      <pki:rdn-attr pki:order="1" pki:id="2.5.4.10">
        <pki:name xml:lang="ru">O</pki:name>
        <pki:value pki:lang="ru">CRYPTO-PRO LLC</pki:value>
      </pki:rdn-attr>
    </pki:rdn>
    <pki:rdn pki:order="3" pki:count="1">
      <pki:rdn-attr pki:order="1" pki:id="2.5.4.7">
        <pki:name xml:lang="ru">L</pki:name>
        <pki:value pki:lang="ru">Moscow</pki:value>
      </pki:rdn-attr>
    </pki:rdn>
  </pki:issuer>
</pki:certificate>
```

```

    </pki:rdn-attr>
  </pki:rdn>
  <pki:rdn pki:order="2" pki:count="1">
    <pki:rdn-attr pki:order="1" pki:id="2.5.4.6">
      <pki:name xml:lang="ru">C</pki:name>
      <pki:value pki:lang="ru">RU</pki:value>
    </pki:rdn-attr>
  </pki:rdn>
  <pki:rdn pki:order="1" pki:count="1">
    <pki:rdn-attr pki:order="1" pki:id="1.2.840.113549.1.9.1">
      <pki:name xml:lang="ru">E</pki:name>
      <pki:value pki:lang="ru">support@cryptopro.ru</pki:value>
    </pki:rdn-attr>
  </pki:rdn>
</pki:issuer>
<pki:validity>
  <pki:notBefore>27.09.2018 19:08:26</pki:notBefore>
  <pki:notAfter>27.12.2018 19:18:26</pki:notAfter>
</pki:validity>
<pki:subject pki:count="1">
  <pki:rdn pki:order="1" pki:count="1">
    <pki:rdn-attr pki:order="1" pki:id="2.5.4.3">
      <pki:name xml:lang="ru">CN</pki:name>
      <pki:value pki:lang="ru">idonly</pki:value>
    </pki:rdn-attr>
  </pki:rdn>
</pki:subject>
<pki:subject-public-key-info>
  <pki:public-key-algorithm>
    <pki:algorithm pki:id="1.2.643.2.2.19">
      <pki:name xml:lang="ru">ГОСТ Р 34.10-2001</pki:name>
    </pki:algorithm>
    <pki:parameters>
      <![CDATA[30 12 06 07 2a 85 03 02 02 24 00 06 07 2a 85 03 02 02 1e 01]]>
    </pki:parameters>
  </pki:public-key-algorithm>
  <pki:subject-public-key>
    <pki:value>
      <![CDATA[04 40 d6 c4 15 66 c2 a5 22 53 38 65 5c 0d 20 60 81 a8 84 fb 21 b0 ea 13 23 e6 cf 6f
de 7c 58 64 12 de 72 34 c6 44 4c cd bd a0 d6 d4 f6 63 11 ba b0 f7 5d db 57 7b 58 af 6f 99 4b 57 da de 88 99
88 40]]>
    </pki:value>
  </pki:subject-public-key>
</pki:subject-public-key-info>
<pki:extensions pki:count="6">
  <pki:extension pki:order="1" pki:critical="no" id="2.5.29.15">
    <pki:name xml:lang="ru">Использование ключа</pki:name>
    <pki:value xml:lang="ru">
      <![CDATA[Цифровая подпись, Неотрекаемость, Шифрование ключей, Шифрование данных (f0)]]>
    </pki:value>
  </pki:extension>
  <pki:extension pki:order="2" pki:critical="no" id="2.5.29.14">
    <pki:name xml:lang="ru">Идентификатор ключа субъекта</pki:name>
    <pki:value xml:lang="ru">
      <![CDATA[fb e3 98 a8 fd 82 3b ea c7 90 df d9 67 11 7a 64 a5 80 c7 b6]]>
    </pki:value>
  </pki:extension>
  <pki:extension pki:order="3" pki:critical="no" id="2.5.29.37">
    <pki:name xml:lang="ru">Улучшенный ключ</pki:name>
    <pki:value xml:lang="ru">
      <![CDATA[Пользователь Центра Регистрации, HTTP, TLS клиент (1.2.643.2.2.34.6), Проверка
подлинности клиента (1.3.6.1.5.5.7.3.2)]]>
    </pki:value>
  </pki:extension>
  <pki:extension pki:order="4" pki:critical="no" id="2.5.29.35">
    <pki:name xml:lang="ru">Идентификатор ключа центра сертификатов</pki:name>

```

```

    <pki:name xml:lang="ru">Идентификатор ключа центра сертификатов</pki:name>
    <pki:value xml:lang="ru">
      <![CDATA[Идентификатор ключа=15 31 7c b0 8d 1a de 66 d7 15 9c 49 52 97 17 24 b9 01 7a 83]]>
    </pki:value>
  </pki:extension>
  <pki:extension pki:order="5" pki:critical="no" id="2.5.29.31">
    <pki:name xml:lang="ru">Точки распространения списков отзыва (CRL)</pki:name>
    <pki:value xml:lang="ru">
      <![CDATA[[1]Точка распределения списка отзыва (CRL): Имя точки распространения:Полное
имя:URL=http://testca.cryptopro.ru/CertEnroll/CRYPTO-PRO%20Test%20Center%202.cr1]]>
    </pki:value>
  </pki:extension>
  <pki:extension pki:order="6" pki:critical="no" id="1.3.6.1.5.5.7.1.1">
    <pki:name xml:lang="ru">Доступ к информации о центрах сертификации</pki:name>
    <pki:value xml:lang="ru">
      <![CDATA[[1]Доступ к сведениям центра сертификации: метод доступа=Поставщик центра
сертификации (1.3.6.1.5.5.7.48.2), дополнительное имя=URL=http://testca.cryptopro.ru/CertEnroll/test-ca-
2014_CRYPTOPRO-PRO%20Test%20Center%202.crt, [2]Доступ к сведениям центра сертификации: метод доступа=Протокол
определения состояния сертификата через сеть (1.3.6.1.5.5.7.48.1), дополнительное
имя=URL=http://testca.cryptopro.ru/ocsp/ocsp.srf]]>
    </pki:value>
  </pki:extension>
</pki:extensions>
<pki:signed-content>
  <pki:signature-algorithm>
    <pki:algorithm pki:id="1.2.643.2.2.3">
      <pki:name xml:lang="ru">ГОСТ Р 34.11/34.10-2001</pki:name>
    </pki:algorithm>
  </pki:signature-algorithm>
  <pki:signature-value>
    <pki:value pki:unused-bits="0">
      <![CDATA[AE 23 FF 85 A0 A7 21 5A 02 FA B4 A3 27 F8 DB 86 77 6A 03 DE B5 7C 55 80 EE 7C 60 E1
60 57 FC 15 BB 11 22 F9 E6 B9 C9 F0 45 32 33 70 FA F7 3D B2 2E 11 5C 92 92 B6 0E CB B0 B3 C0 21 27 06 58
F5]]>
    </pki:value>
  </pki:signature-value>
</pki:signed-content>
</pki:certificate>

```

Получение списка запросов на отзыв/приостановление/восстановление

ПАРАМЕТР		ЗНАЧЕНИЕ
HTTP-метод		GET
Путь		https://dss.cryptopro.ru/SignServer/rest/api/certificates/{cert_id}/revokerequests
Параметры		cert_id - идентификатор сертификата
Возвращаемое значение		List<DSSRevRequest> - Список сертификатов
HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404		Сертификат с указанным ID не найден

Пример запроса

```

GET https://host/SignServer/rest/api/certificates/15/revokerequests HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1Qi ... FQKc8WFHkugEg
Host: host

```

Пример ответа

```
HTTP/1.1 200 OK
Content-Length: 4263
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5

[{
  "Base64Request": "MIILTAYJKoZIHvcNAQcCoIILPTC ... SPQVB97aK36gqswNYKW3X8NGRK01i+Q==",
  "ID": "6a5c8b4a-52c6-e811-80db-00155d454d12",
  "CertificateAuthorityID": 6,
  "Status": "PENDING",
  "RevInfo": {
    "RevocationReason": "CRL_REASON_CERTIFICATE_HOLD",
    "RevocationDate": "0001-01-01T00:00:00",
    "RevocationComments": "",
    "UnholdDate": "2018-10-25T14:48:00",
    "DistName": null,
    "RequestDate": "2018-10-02T14:48:59",
    "ApproveDate": "0001-01-01T00:00:00",
    "UnholdAction": "U",
    "SignedRequest": null
  }
}]
```

Установка сертификата

ПАРАМЕТР		ЗНАЧЕНИЕ
HTTP-метод		POST
Путь		https://dss.cryptopro.ru/SignServer/rest/api/certificates
Параметры		certificate - BASE64-содержимое сертификата
Возвращаемое значение		DSSCertificate - Сертификат пользователя
HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	request_not_found	Не найден соответствующий сертификату запрос
400	invalid_certificate_format	Некорректный формат входных данных
400	appropriate_enroll_not_found	Не найден соответствующий обработчик УЦ
400	invalid_request_status	Соответствующий запрос не находится в состоянии "Обрабатывается"

Пример запроса

```
POST https://host/SignServer/rest/api/certificates/ HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh ... 8TElfEktxVFCuR27tUg
Content-Type: application/json; charset=utf-8
Host: host
Expect: 100-continue

{"certificate":"MIIC/jCCAq2gAwIBAgITEgAv ..."}

```

Пример ответа


```
HTTP/1.1 200 OK
Content-Length: 4263
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5

[{
  "CertificateType": "ServerSide",
  "ID": 2,
  "DName": "CN=Test1",
  "CertificateBase64":
"MIIC/jCCAq2gAwIBAgITEgAvWSQcF1NFyPSX5AAAAC9ZJDAIBgYqhQMCAGMwfzEjMCEGCSqGSIB3DQEJARYUc3VwcG9ydEBjcmlwdG9wcm8u
cnUxCzAJBgNVBAYTA1JVMQ8wDQYDVQQHEWZnbn3Njb3cxZzAvZAVBgNVBAoTDkNSWVBUTy1QUk8gTEExDMSEwHwYDVQQDEExhDUIlQVE8tUFJPIFRlc
3QgQ2VudGVyIDIwHhcNMTgxMTE0NjAyWWhcNMTkwMjE5MTE1NjAyWjAQMjQ4wDAYDVQQDEwVUZXR0MTBjMBwGBiqFAwICEzASBgqhQMCAi
QABgcqhQMCAh4BA0MABEB0PKbSC4qOptWM6iNj+1RSP2KKQEPgJlL+AmbAS01b8+19ztSwPzLZeWcxssHPU34Bkt5JMDbS71yn/1pgAtiTo4I
BbTCCAkwCwYDVR0PBAQDAgTwMB0GA1UdDgQWBBTKDKIrDj/kAdg4a5d6mQBadkDXkTATBgNVHSUEDDAKBggqhQMCAi4ACDAfBgNVHSMEGDAW
gBQVMXywjRreZtcVnElSlxckuQF6gzBZBgNVHR8EUjBQME6gTKBKhhodHRwOi8vdGVzdGNhLmNyeXB0b3Byby5ydS9DZXJ0RW5yb2xsL0NSW
VBUTy1QUk8lMjBUZXN0JTJwQ2VudGVyJTJwMi5jcmwwgagGCCsGAQUFBwEBB1GcMIGZMGECCsGAQUFBzACHlVodHRwOi8vdGVzdGNhLmNyeX
B0b3Byby5ydS9DZXJ0RW5yb2xsL3Rlc3QtY2EtMjAxNF9DUl1QVE8tUFJPIjJwVGVzdCUyMENlbmRlciUyMDIuY3J0MDQGCCsGAQUFBzABhih
odHRwOi8vdGVzdGNhLmNyeXB0b3Byby5ydS9vY3NwL29jc3Auc3JmMagGBiqFAwICAwnBAFPsyyX3s39AvO786Fuc/jYmaFG7y/RUEPygviWt
ygbbl9Ix9h3qo5Sh7XOT/PIyEVC13/9DnP4wtBffEh2tHvg=",
  "Status": {
    "Value": "ACTIVE",
    "RevocationInfo": null,
    "PinCode": null,
    "ActiveCertId": 0
  },
  "IsDefault": false,
  "CertificateAuthorityID": 11,
  "CspID": "332101a9-4aca-4f89-94fc-9eb771b2588c",
  "HashAlgorithms": [
    "GOST R 34.11-94"
  ],
  "ProviderName": null,
  "ProviderType": 0,
  "PrivateKeyNotBefore": null,
  "PrivateKeyNotAfter": null,
  "HasPin": false,
  "FriendlyName": ""
}]
```

Смена ПИН-кода

ПАРАМЕТР		ЗНАЧЕНИЕ
HTTP-метод		POST
Путь		https://dss.cryptopro.ru/SignServer/rest/api/certificates/{cert_id}/pin
Параметры		cert_id - идентификатор сертификата PinInfo - данные для смены ПИН-кода
Возвращаемое значение		Метод не имеет возвращаемого значения
HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404		Сертификат с указанным ID не найден
400	invalid_pin	Неверно введен текущий ПИН-код

Пример запроса

```
POST https://host/SignServer/rest/api/certificates/15/pin HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLC0ubG9ja3Raz4KhIP6FrowzA
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 66
Expect: 100-continue

{"OldPin":"","NewPin":"1234"}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Length: 0
Server: Microsoft-IIS/7.5
```

Валидация ПИН-кода

ПАРАМЕТР	ЗНАЧЕНИЕ	
HTTP-метод	POST	
Путь	https://dss.cryptopro.ru/SignServer/rest/api/certificates/{cert_id}/pin/check	
Параметры	cert_id - идентификатор сертификата ValidatePinRequest - Запрос на валидацию ПИН-кода	
Возвращаемое значение	В случае, если ValidatePinRequest содержит CheckAttempts со значением <code>true</code> , возвращает объект ValidatePinResult В противном случае возвращается bool значение, сообщающее о том, завершилась ли валидация успешно.	
HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404		Сертификат с указанным ID не найден

Пример запроса

```
POST /SignServer/rest/api/certificates/5/pin/check HTTP/1.1
Content-Type: application/json; charset=utf-8
Authorization: Bearer eyJ0eXA...
cache-control: no-cache

{"CheckAttempts":true,"pin":"1234"}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Date: Thu, 14 Mar 2019 13:18:44 GMT
Content-Length: 64

{"IsValid":false,"ErrorDescription":"key_blocked","Remaining":0}
```

Назначение сертификата по умолчанию

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST

ПАРАМЕТР		ЗНАЧЕНИЕ
Путь		https://dss.cryptopro.ru/SignServer/rest/api/certificates/{cert_id}/default
Параметры		cert_id - идентификатор сертификата DefaultProperty - Данные для назначения сертификата по умолчанию
Возвращаемое значение		Метод не имеет возвращаемого значения
HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404		Сертификат с указанным ID не найден

Пример запроса

```
POST https://host/SignServer/rest/api/certificates/15/default HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh... 8TElfEktxVFCuR27tUg
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 16
Expect: 100-continue

{"Default":true}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Length: 0
Server: Microsoft-IIS/7.5
```

Назначение дружественного имени

ПАРАМЕТР		ЗНАЧЕНИЕ
HTTP-метод		POST
Путь		https://dss.cryptopro.ru/SignServer/rest/api/certificates/{cert_id}/friendlyname
Параметры		cert_id - идентификатор сертификата FriendlyNameProperty - Данные для назначения дружественного имени
Возвращаемое значение		Метод не имеет возвращаемого значения

Примечание

Дружественное имя должно быть уникально среди всех сертификатов пользователя.

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404		Сертификат с указанным ID не найден
400	duplicate_friendly_name	Указанное дружественно имя уже назначено
400	invalid_friendly_name_length	Длина дружественного имени больше 255 символов

Пример запроса

```
POST https://host/SignServer/rest/api/certificates/15/friendlyname HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJK...d0KhU_-Qz8jgDFlcFU_RMA
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 32
Expect: 100-continue

{"FriendlyName":"Friendly Name"}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Length: 0
Server: Microsoft-IIS/7.5
```

Отзыв

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://dss.cryptopro.ru/SignServer/rest/api/certificates/{cert_id}/status
Параметры	cert_id - идентификатор сертификата CertificateStatus - параметры отзыва сертификата
Возвращаемое значение	Метод не имеет возвращаемого значения

При отзыве сертификата в поле `Value` должно быть указано значение 2 ([REVOKED](#))

В структуре `RevocationInfo` необходимо заполнить:

- `RevocationReason` - причину отзыва сертификата [CertRevokeReasonEnum](#)
- `RevocationDate` (опционально) - дату отзыва, если требуется отозвать сертификат в будущем
- `RevocationComments` (опционально) - комментарий

Формат даты `RevocationDate` - yyyy-MM-dd hh:mm:ss (0001-01-01 00:00:00) или yyyy-MM-ddThh:mm:ss (0001-01-01T00:00:00).

Запрос на отзыв может быть подписан действующим сертификатом пользователя. В этом случае необходимо заполнить поля `ActiveCertId` и `PinCode`.

Примечание

Подпись запросов на отзыв/приостановление/восстановление возможна только если в настройках модуля интеграции с УЦ 2.0 выставлен флаг `AllowUserMode`. Настройку должен выполнить Администратор DSS.

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404		Сертификат с указанным ID не найден
500		Сертификат уже был отозван. Внутренняя ошибка сервиса
400	invalid_parameters	Невалидные параметры запроса

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_certificate_status	Повторный отзыв сертификата

Пример запроса

```
POST https://host/SignServer/rest/api/certificates/16/status HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJ...VCgDbAc5XLyjt0pwkET7I2fg
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 228
Expect: 100-continue

{
  "Value": 2,
  "RevocationInfo": {
    "RevocationReason": 1,
    "RevocationComments": "Comment"
  }
}
```

Запрос с указанием даты отзыва:

```
POST https://host/SignServer/rest/api/certificates/16/status HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJ...VCgDbAc5XLyjt0pwkET7I2fg
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 228
Expect: 100-continue

{
  "Value": 2,
  "RevocationInfo": {
    "RevocationReason": 1,
    "RevocationComments": "Comment",
    "RevocationDate": "2018-10-05T12:58"
  }
}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Length: 0
Server: Microsoft-IIS/7.5
```

Приостановление

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://dss.cryptopro.ru/SignServer/rest/api/certificates/{cert_id}/status
Параметры	cert_id - идентификатор сертификата CertificateStatus - параметры отзыва сертификата
Возвращаемое значение	Метод не имеет возвращаемого значения

При отзыве сертификата в поле `Value` должно быть указано значение 3 ([HOLD](#))

В запросе необходимо указать:

- `UnholdDate` - дату восстановления сертификата
 - `UnholdAction` - причину отзыва, которая будет указана в CRL, если сертификат не будет восстановлен до даты `UnholdDate` (`CertRevokeReasonEnum`).
 - `RevocationComments` (опционально) - комментарий
- Если требуется приостановить сертификат в будущем, то необходимо указать значение поля `RevocationDate`.
- Формат даты `RevocationDate` - уууу-MM-dd hh:mm:ss (0001-01-01 00:00:00) или уууу-MM-ddThh:mm:ss (0001-01-01T00:00:00).
- Запрос на отзыв может быть подписан действующим сертификатом пользователя. В этом случае необходимо заполнить поля `ActiveCertId` и `PinCode`.

Примечание

Подпись запросов на отзыв/приостановление/восстановление возможна только если в настройках модуля интеграции с УЦ 2.0 выставлен флаг `AllowUserMode`. Настройку должен выполнить Администратор DSS.

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404		Сертификат с указанным ID не найден
400	invalid_parameters	Невалидные параметры запроса
400	invalid_certificate_status	Повторный отзыв сертификата

Пример запроса

```
POST https://host/SignServer/rest/api/certificates/19/status HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLC...vZdqvGkG431UxNH0EA
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 278
Expect: 100-continue

{
  "Value": 3,
  "RevocationInfo": {
    "RevocationComments": "Comment",
    "UnholdDate": "2018-10-12T12:13:58",
    "UnholdAction": "0"
  },
  "ActiveCertId": 0
}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Length: 0
Server: Microsoft-IIS/7.5
```

Восстановление

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST

ПАРАМЕТР	ЗНАЧЕНИЕ
Путь	https://dss.cryptopro.ru/SignServer/rest/api/certificates/{cert_id}/status
Параметры	cert_id - идентификатор сертификата CertificateStatus - параметры отзыва сертификата
Возвращаемое значение	Метод не имеет возвращаемого значения

При отзыве сертификата в поле Value должно быть указано значение 1 (ACTIVE)

Запрос на отзыв может быть подписан действующим сертификатом пользователя. В этом случае необходимо заполнить поля ActiveCertId и PinCode.

Примечание

Подпись запросов на отзыв/приостановление/восстановление возможна только если в настройках модуля интеграции с УЦ 2.0 выставлен флаг AllowUserMode. Настройку должен выполнить Администратор DSS.

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404		Сертификат с указанным ID не найден

Пример запроса

```
POST https://host/SignServer/rest/api/certificates/19/status HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1Qi ... ZkmjxmKuKej6f2Cyhg
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 245
Expect: 100-continue

{
  "Value": 1,
  "RevocationInfo": {
    "RevocationComments": "Comment"
  }
}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Length: 0
Server: Microsoft-IIS/7.5
```

Конечная точка Documents

Конечная точка /documents предоставляет следующие методы:

- Подпись документа
- Подпись пакета документов
- Зашифрование документа
- Расшифрование документа
- Разбор зашифрованного документа
- Усовершенствование подписи
- хэширование документа

Примеры подписи приведены в разделе [Примеры запросов на создание подписи](#)

Подпись одного документа

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://dss.cryptopro.ru/SignServer/rest/api/documents
Параметры	Document - Информация о документе
Возвращаемое значение	string - Подписанный документ в кодировке Base64

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_signature_params	Неверно указаны параметры подписи
400	invalid_certificate	Указан неверный идентификатор сертификата
400	invalid_pin	Указан неверный ПИН-код
400	content_required	Не передан документ для подписания
500	An error has occurred	Внутренняя ошибка сервера

Примечание

Одной из причин ошибки HTTP 500 может быть несоответствие формата документа и формата подписи. Например, для подписи формата PDF отправлен текстовый документ.

Примеры запросов

Запрос


```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGc ... Y9AEWNCMQxHiug
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 143
Expect: 100-continue
```

```
{
  "Content": "AQ==",
  "Signature": {
    "Type": 2,
    "Parameters": {
      "CADESType": "BES",
      "IsDetached": "true"
    },
    "CertificateId": 14,
    "PinCode": ""
  }
}
```

Ответ

```
HTTP/1.1 200 OK
Content-Length: 1922
Content-Type: application/json; charset=utf-8

"MIIFmgYJKoZIhvcNA ... YYxySjbyJJg=="
```

Подпись пакета документов

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://dss.cryptopro.ru/SignServer/rest/api/documents/packagesignature
Параметры	DocumentPackage - Пакет документов
Возвращаемое значение	DSSSignDocumentResponse - Результат подписание пакета документов

Результатом выполнения метода является структура [DSSSignDocumentResponse](#), содержащая список подписанных документов. Порядок подписанных документов соответствует порядку документов, переданных в запросе.

Если при подписании одного или нескольких документов возникли ошибки, то в ответе сервиса в поле `Results` на соответствующей позиции будет выставлено значение null, а в поле `Error` на соответствующей позиции будет сообщение об ошибке.

При создании пакетной отделённой соподписи (как самого документа, так и его хэш-значения) содержимое исходного документа (или его хэш-значение) передаётся через структуру `DocumentContent` (в поле `OriginalContent`), параметр `OriginalDocument` не используется.

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_signature_params	Неверно указаны параметры подписи
400	invalid_certificate	Указан неверный идентификатор сертификата

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_pin	Указан неверный ПИН-код
400	content_required	Не передан документ для подписания
500	An error has occurred	Внутренняя ошибка сервера

Примеры запросов

Запрос

```
POST https://host/SignServer/rest/api/documents/packagesignature HTTP/1.1
Authorization: Bearer eyJ0eXAiOi ... P9RY0lwn-r5HGc7ZyTQ
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 217
Expect: 100-continue

{
  "Documents": [{
    "Content": "AQ=="
  }, {
    "Content": "Ag=="
  }, {
    "Content": "Aw=="
  }
],
  "Signature": {
    "Type": 2,
    "Parameters": {
      "CADESType": "BES",
      "IsDetached": "true"
    },
    "CertificateId": 14,
    "PinCode": ""
  }
}
```

Ответ

```
HTTP/1.1 200 OK
Content-Length: 5772
Content-Type: application/json; charset=utf-8

{
  "Results": ["JCBsXwo7P7ZZ5Qt568GTmnLZXdoI8xICKGrM6le9WzIC+v3JGGyewvQM3uXhEAUgd0LhBf9GjZyocK488POYrw==",
    "G2JTqRXlouM/tDxCN2Ja8ozwRdYp10eVY5cyfrAdRT27vGeJlagJCO0DgTbBeWsVFTctufkbpqfIeSsmNiNiQ==",
    "4csaHBAmFecNCTAlg5beoGj465zjlGZkG4QkeM1D+zcs1TgqBm38uJlTOcfcRyziw1RKqelCwH6EidpDqIzsmg==" ],
  "Errors": [null, null, null]
}
```

Зашифрование документа

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://dss.cryptopro.ru/SignServer/rest/api/documents/encrypt

ПАРАМЕТР	ЗНАЧЕНИЕ
Параметры	Document - Информация о документе
Возвращаемое значение	string - Зашифрованный документ в кодировке Base64

Примечание

В структуре [Document](#) должно быть заполнено поле `Encryption`

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	parameters_required	Не заполнено поле Encryption
400	content_required	Не передан документ для зашифрования
400	invalid_content	Неправильный формат документа (например, в случае XML шифрования, необходимо передать корректный xml-документ)
400	invalid_recipients	Сертификаты получателей не заданы или невалидны
500	An error has occurred	Внутренняя ошибка сервера

Примеры запросов

Запрос

```
POST https://dss.cryptopro.ru/SignServer/rest/api/documents/encrypt HTTP/1.1
Authorization: Bearer eyJ0eXAiOi ... P9RY0lwn-r5HGc7ZyTQ
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 1827

{
  "Content": "PHJvb3Q+P ... ZT48L3Jvb3Q+",
  "Encryption": {
    "Type": 1,
    "Parameters": {},
    "Certificates": [ "MIIE7DC ... 4oXwmndTQ==" ]
  }
}
```

Ответ

```
HTTP/1.1 200 OK
Content-Length: 3543
Content-Type: application/json; charset=utf-8

"PEVuY3J5 ... J5cHRlZERhdGE+"
```

[Расшифрование документа](#)

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://dss.cryptopro.ru/SignServer/rest/api/documents/decrypt
Параметры	Document - Информация о документе
Возвращаемое значение	DSSDecryptDocumentResponse - Расшифрованный документ в кодировке Base64

Примечание

В структуре [Document](#) должно быть заполнено поле `Decryption`

Метод /decrypt/parse должен быть использован, если пользователь не знает заранее идентификатор сертификата для расшифрования. Результатом работы метода будет список идентификаторов сертификатов пользователя.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://dss.cryptopro.ru/SignServer/rest/api/documents/decrypt/parse
Параметры	Document - Информация о документе
Возвращаемое значение	List - Список идентификаторов сертификатов пользователя

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	content_required	Не передан документ для расшифрования
400	invalid_content	Неправильный формат документа (например, в случае XML шифрования, необходимо передать корректный xml-документ)
400	certificate_not_found	Не найден сертификат для расшифрования
400	invalid_certificate	Указан неверный идентификатор сертификата
400	invalid_pin	Указан неверный ПИН-код
500	An error has occurred	Внутренняя ошибка сервера

Примеры запросов

Запрос на получение идентификаторов сертификатов для расшифрования

```
POST https://dss.cryptopro.ru/SignServer/rest/api/documents/decrypt/parse HTTP/1.1
Authorization: Bearer eyJ0eXAiOi ... P9RY0lwn-r5HGc7ZyTQ
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 3597
```

```
{
  "Content": "PEVuY3J5 ... J5cHRlZERhdGE+",
  "Decryption": {
    "Type": 1,
    "CertificateId": 0
  }
}
```

Ответ

```
HTTP/1.1 200 OK
Content-Length: 5
Content-Type: application/json; charset=utf-8

[680]
```

Запрос на расшифрование

```
POST https://dss.cryptopro.ru/SignServer/rest/api/documents/decrypt HTTP/1.1
Authorization: Bearer eyJ0eXAiOi ... P9RY0lwn-r5HGc7ZyTQ
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 3599
```

```
{
  "Content": "PEVuY3J5 ... J5ZERhdGE+",
  "Decryption": {
    "Type": 1,
    "CertificateId": 680
  }
}
```

Ответ

```
HTTP/1.1 200 OK
Content-Length: 67
Content-Type: application/json; charset=utf-8

"PHJvb3Q+P ... ZT48L3Jvb3Q+"
```

Усовершенствование подписи

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://dss.cryptopro.ru/SignServer/rest/api/documents/enhancesignature
Параметры	Document - Информация о документе
Возвращаемое значение	string - Подписанный документ в кодировке Base64

Примечание

В структуре [Document](#) должно быть заполнено поле [Signature](#)

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_signature_params	Неверно указаны параметры подписи
400	content_required	Не передан документ для подписания
500	An error has occurred	Внутренняя ошибка сервера

Примеры запросов

Запрос

Ответ

Хэширование документа

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://dss.cryptopro.ru/SignServer/rest/api/documents/hash
Параметры	Document - Информация о документе
Возвращаемое значение	string - Хэш-значение документа в кодировке Base64

Примечание

Хэширование документа работает только в поточном режиме. Подробнее о поточной обработке данных см. [Потоковая обработка](#)

Создание транзакций - выполнение операций на Сервисе Подписи с подтверждением вторым фактором аутентификации.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://dss.cryptopro.ru/SignServer/rest/api/transactions
Параметры	Transaction - Параметр транзакции
Возвращаемое значение	string - Идентификатор транзакции

Типовые ошибки

Примеры запросов

Запрос на создание сертификата

```
POST https://host/SignServer/rest/api/transactions HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGci ... pz4erYJpgoN_RgQLA
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 200
Expect: 100-continue

{
  "OperationCode":16,
  "Parameters":
  [
    {"Name":"CertSubjectName","Value":"CN=dssUser,C=RU"},
    {"Name":"CAId","Value":"11"},
    {"Name":"EkuString","Value":"1.2.643.2.2.34.2,1.2.643.2.2.34.4,1.3.6.1.5.5.7.3.2"}
  ]
}
```

Запрос на подпись документа

```
POST https://host/SignServer/rest/api/transactions HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh ... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 355049
Expect: 100-continue

{
  "OperationCode":2,
  "Parameters":
  [
    {"Name":"SignatureType","Value":"CMS"},
    {"Name":"CertificateID","Value":"13"},
    {"Name":"DocumentInfo","Value":"testPdf.pdf"},
    {"Name":"DocumentType","Value":"pdf"},
    {"Name":"IsDetached","Value":"false"},
    {"Name":"CADESType","Value":"BES"}
  ],
  "Document":"JVBERi0xLjUNCiW1tbW14Kfu ...."
}
```

Конечная точка Operations

Конечная точка

Operations

 предназначена для создания различных криптографических операций.

Методы данной конечной точки позволяют выполнять операции над множеством документов, содержимое которых загружается из удалённого хранилища документов.

Получение сведений об операции

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	/operations/{opid}
Параметры	Идентификатор операции.
Возвращаемое значение	Объект OperationInfo , содержащий сведения о созданной операции.

Отправка запроса на выполнения операции

Метод отправляет запрос на выполнение операции после её подтверждения в ЦИ.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	/operations/{opid}/start
Параметры	Идентификатор операции.
Возвращаемое значение	Метод не имеет возвращаемого значения.

Конечная точка Signature

Конечная точка

Signature

 предназначена для создания операций подписи документов.

Методы данной конечной точки позволяют выполнять операции над множеством документов, содержимое которых загружается из удалённого хранилища.

Создание операции подписи

Метод предназначен для создания операции подписи документов.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	/v2/signature
Параметры	Объект SignatureOperation , содержащий сведения об операции подписи.
Возвращаемое значение	Объект OperationInfo , содержащий сведения о созданной операции.

Если операция подписи требует подтверждения, то поле

Status

 будет содержать значение

Created

, иначе

Completed

.

Пример запроса

```
POST /SignServer/rest/api/v2/signature HTTP/1.1
Content-Type: application/json
Authorization: Bearer eyJ...

{
  "BinaryData": [{"RefId": "3a6629fa-5759-4d4a-8602-0df7ac8420aa"}],
  "Signature" :
  {
    "CertificateId": "0",
    "ProcessingTemplateId": "1"
  }
}
```

Примеры ответов

Операция требует подтверждения.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
  "Operation": {
    "Id": "ea2eb4e5-dc7c-4c33-9b72-5efba3fd0f75", ,
    "Status": "Created"
  }
}
```

Операция не требует подтверждения, результат подписи возвращён в ответе.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
  "Operation": {
    "Id": "ea2eb4e5-dc7c-4c33-9b72-5efba3fd0f75",
    "Result": {
      "ProcessedDocuments": [
        {
          "RefId": "257c1cbb-6494-45ee-9a00-d55a63e5b703",
          "OriginalRefId": "3a6629fa-5759-4d4a-8602-0df7ac8420aa",
          "Status": "Completed"
        }
      ]
    },
    "Status": "Completed"
  }
}
```

Выполнение операции подписи

Метод предназначен для выполнения операции подписи документов после подтверждения.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	/v2/signature
Параметры	Объект SignatureOperation , содержащий сведения об операции подписи.
Возвращаемое значение	Объект OperationInfo , содержащий сведения о созданной операции.

Для выполнения подтверждённой операции запрос отправляется на ту же конечную точку, что и при создании операции.

В теле запроса в структуре `SignatureOperation` может быть заполнено только одно поле `PinCode`, если ПИН-код для доступа к ключу не требуется, то весь запрос представляет из себя пустой JSON-объект.

Примеры запросов

Без передачи ПИН-кода для доступа к закрытому ключу сертификата.

```
POST /SignServer/rest/api/v2/signature HTTP/1.1
Content-Type: application/json
Authorization: Bearer eyJ...

{}
```

С передачей ПИН-кода для доступа к закрытому ключу сертификата.

```
POST /SignServer/rest/api/v2/signature HTTP/1.1
Content-Type: application/json
Authorization: Bearer eyJ...

{
  "Signature" :
  {
    "PinCode" : "1234567890"
  }
}
```

Пример ответа

HTTP/1.1 200 OK

Content-Type: application/json; charset=utf-8

```
{
  "Operation": {
    "Id": "ea2eb4e5-dc7c-4c33-9b72-5efba3fd0f75",
    "Result": {
      "ProcessedDocuments": [
        {
          "RefId": "257c1cbb-6494-45ee-9a00-d55a63e5b703",
          "OriginalRefId": "3a6629fa-5759-4d4a-8602-0df7ac8420aa",
          "Status": "Completed"
        }
      ]
    },
    "Status": "Completed"
  }
}
```

Конечная точка Keys

Конечная точка keys предоставляет следующие методы:

- Генерация закрытого ключа
- Назначение параметров закрытого ключа
- Получение параметров закрытого ключа

Генерация закрытого ключа

ПАРАМЕТР		ЗНАЧЕНИЕ
HTTP-метод		POST
Путь		https://dss.cryptopro.ru/Signserver/rest/api/keys/generate
Параметры		AlgId - идентификатор алгоритма закрытого ключа, ParametersOid - идентификатор набора параметров эллиптической кривой, PinCode - пин-код на ключевой контейнер.
Возвращаемое значение		GenKeyOutput - Результат выполнения операции

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_pin	Некорректный пин-код
400	algid_is_not_supported	Запрошенный идентификатор алгоритма не поддерживается
400	param_set_is_not_supported	Набор параметров кривой не поддерживается
400	prorvider_not_found	Не найден запрошенный криптографический провайдер
400	enroll_is_missing	Не найден подходящий модуль УЦ

Пример запроса

```
POST https://host/signserver/rest/api/keys/generate HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLC08bnN9Raz4KhIP6FrowzA
Content-Type: application/json; charset=utf-8
Host: host
{
  "AlgId": "11849",
  "ParametersOid": "1.2.643.2.2.36.0",
  "PinCode": ""
}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  "IsError": false,
  "ErrorDescription": null,
  "CertId": 1011,
  "Base64Certificate": "MIIBJDCB0..."
}
```

Назначение параметров закрытого ключа

ПАРАМЕТР		ЗНАЧЕНИЕ
HTTP-метод		POST
Путь		https://dss.cryptopro.ru/SignServer/rest/api/keys/{key}/params
Параметры		Key - идентификатор сертификата, связанного с закрытым ключом, Param - имя назначаемого параметра, Data - значение назначаемого параметра.
Возвращаемое значение		Отсутствует

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	cert_not_found	Не найден сертификат, связанный с закрытым ключом
400	key_not_found	Не найден закрытый ключ
400	param_is_null	Пустое имя параметра
400	value_is_null	Пустое значение параметра

Пример запроса

```
POST https://host/SignServer/rest/api/keys/11/params HTTP/1.1

Content-Type: application/json
Authorization: Bearer eyJ0eXAiO....
Host: host
{
  "Param": "testParam",
  "Data": "testValue2"
}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
  "IsError": false,
  "ErrorDescription": null,
}
```

Получение параметров закрытого ключа

ПАРАМЕТР		ЗНАЧЕНИЕ
HTTP-метод		Get
Путь		https://dss.cryptopro.ru/SignServer/rest/api/keys/{key}/params?param={param}
Параметры		Key - идентификатор сертификата, связанного с закрытым ключом, Param - имя требуемого параметра.

ПАРАМЕТР	ЗНАЧЕНИЕ	
Возвращаемое значение	string - значение запрошенного параметра	
HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404		Не найден закрытый ключ

Пример запроса

```
GET https://host/SignServer/rest/api/keys/11/params?param=testParam HTTP/1.1
Host: host
Content-Type: application/json
Authorization: Bearer eyJ0eXAiOiJKV1QiLC...
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
  "Value": "testValue2",
  "IsError": false,
  "ErrorDescription": null
}
```

Типы данных:

- Политика Сервиса Подписи (DssPolicy)
- Параметры УЦ (DssCaPolicy)
- Параметры криптопровайдеров (DSSCSPPolicy)
- Параметры запроса на сертификат (CertificateRequest)
- Запрос на сертификат (DSSCertRequest)
- Сертификат (DSSCertificateEx)
- Статус сертификата (CertificateStatus)
- Типы статусов сертификата (DSSCertificateStatusEnum)
- Информация об отзыве сертификата (RevocationInfo)
- Причины отзыва сертификата (CertRevokeReasonEnum)
- Данные о документе (Document)
- Данные о пакете документов (DocumentPackage)
- Подписанный пакет документов (DSSSignDocumentResponse)
- Данные транзакций (Transaction)
- Параметры транзакций
- Типы транзакций
- Форматы представления сертификата и запроса на сертификат (DSSCertificateFormatEnum)
- Тип запроса (CARequestTypeEnum)
- Статус запроса на сертификат (DSSRequestStatusEnum)
- Сведения об отзыве/приостановлении/восстановлении (DSSRevRequest)
- Данные для смены ПИН-кода (RequestChangePin)
- Данные для назначения сертификата по умолчанию (DefaultProperty)
- Данные для назначения дружественного имени (FriendlyNameProperty)
- Результат расшифрования (DSSDecryptDocumentResponse)
- Типы шифрования (EncryptionType)

Политика Сервиса Подписи (DSSPolicy)

Политику Сервиса Подписи можно получить вызвав метод [/policy](#).

Политика Сервиса Подписи определяют основные настройки Сервиса Подписи:

- Список подключенных Удостоверяющих Центров
- Список подключенных криптопровайдеров
- Список разрешённых форматов подписи
- Требования к ПИН-коду
- Требование к подтверждению операций.
- Список TSP-служб

Описание полей DSSPolicy:

ПОЛЕ	ТИП	ОПИСАНИЕ
CAPolicy	List< DSSCAPolicy >	Список удостоверяющих центров, настроенных на Сервере Подписи.
CSPsPolicy	List< DSSCSPPolicy >	Список криптопровайдеров настроенных на Сервисе Подписи.
PinCodeMode	PinCodeMode	Режим установки пин-кода на контейнер закрытого ключа.
TspServices	List< TspService >	Список TSP служб, настроенных на Сервере Подписи.
TransactionConfirmation	bool	Устаревшее. Использовать или нет подтверждение транзакций при подписи.
AllowedSignatureType	List< SignatureType >	Список типов подписи, зарегистрированных на Сервере Подписи.
ActionPolicy [устаревший]	List< DSSAction >	Политика подтверждения действий сервера подписи.

Описание полей DSSAction:

ПОЛЕ	ТИП	ОПИСАНИЕ
Action	DSSActions	Тип действия
DisplayName	string	Отображаемое имя действия
Uri	string	Идентификатор действия (является идентификатором соответствующего утверждения)
MfaRequired	Bool	Требование двухфакторной аутентификации. True – требуется, False – не требуется.

Типы подписи SignatureType

Перечисление содержит поддерживаемые Сервером Подписи форматы подписи.

ИМЯ	ОПИСАНИЕ
XMLDSig	Подпись документа в формате XMLDSig
GOST3410	Электронная подпись по ГОСТ Р 34.10 - 2001

ИМЯ	ОПИСАНИЕ
CAdES	Подпись формата CAdES-BES, CAdES-T, CAdES-X Long Type 1
PDF	Подпись PDF документов
MSOffice	Подпись документов MS Word и Excel
CMS	Подпись формата CAdES-BES

Режимы установки Пин-кода PinCodeMode

Перечисление содержит политику работы с пин-кодом.

ИМЯ	ОПИСАНИЕ
Required	Установка пин-кода обязательна.
Forbid	Установка пин-кода запрещена.
Allow	Установка пин-кода опциональна.

Описание полей TspService:

ПОЛЕ	ТИП	ОПИСАНИЕ
Name	string	Идентификатор службы
Title	string	Отображаемое имя службы
Url	string	Адрес службы

Пример Политики Сервиса Подписи

```
{
  "CAPolicy": [{
    "ID": 11,
    "Name": "Out of Band",
    "Active": true,
    "AllowUserMode": false,
    "SNChangesEnable": true,
    "NamePolicy": [{
      "IsRequired": false,
      "Order": 2,
      "OID": "1.2.840.113549.1.9.1",
      "Name": "E-Mail",
      "Value": null,
      "StringIdentifier": "E"
    }, {
      "IsRequired": false,
      "Order": 8,
      "OID": "1.2.643.3.131.1.1",
      "Name": "ИНН",
      "Value": null,
      "StringIdentifier": "INN"
    }, {
      "IsRequired": false,
```

```

        "Order": 4,
        "OID": "2.5.4.7",
        "Name": "Населенный пункт",
        "Value": null,
        "StringIdentifier": "L"
    }, {
        "IsRequired": false,
        "Order": 7,
        "OID": "1.2.643.100.1",
        "Name": "ОГРН",
        "Value": null,
        "StringIdentifier": "OGRN"
    }, {
        "IsRequired": false,
        "Order": 5,
        "OID": "2.5.4.10",
        "Name": "Организация",
        "Value": null,
        "StringIdentifier": "O"
    }, {
        "IsRequired": false,
        "Order": 6,
        "OID": "2.5.4.11",
        "Name": "Подразделение",
        "Value": null,
        "StringIdentifier": "OU"
    }, {
        "IsRequired": false,
        "Order": 3,
        "OID": "2.5.4.8",
        "Name": "Регион",
        "Value": null,
        "StringIdentifier": "S"
    }, {
        "IsRequired": false,
        "Order": 9,
        "OID": "2.5.4.6",
        "Name": "Страна",
        "Value": null,
        "StringIdentifier": "C"
    }, {
        "IsRequired": true,
        "Order": 1,
        "OID": "2.5.4.3",
        "Name": "Общее имя",
        "Value": null,
        "StringIdentifier": "CN"
    }
    ],
    "EKUTemplates": {
        "Временный сертификат администратора УЦ": ["1.2.643.2.2.34.2", "1.2.643.2.2.34.4",
"1.3.6.1.5.5.7.3.2"],
        "Временный сертификат оператора УЦ": ["1.2.643.2.2.34.2", "1.2.643.2.2.34.5",
"1.3.6.1.5.5.7.3.2"],
        "Временный сертификат пользователя УЦ": ["1.2.643.2.2.34.2 ", "1.3.6.1.5.5.7.3.2"],
        "Временный сертификат пользователя УЦ1": ["1.2.643.2.2.34.6", "1.3.6.1.5.5.7.3.2"],
        "Сертификат пользователя УЦ": ["1.2.643.2.2.34.6", "1.3.6.1.5.5.7.3.2"]
    },
    "CAType": "DSSOutOfBandEnroll",
    "ValidationMode": "CertificateAuthority"
}
],
"CSPsPolicy": [{
    "ID": "67e6f39d-c6c5-4ce6-9535-4e22bae84786",
    "GroupID": "e8e67f9e-7eed-4116-ad98-20582e4d766e",
    "Type": "DSS"
}]

```

```

        typeID : 2,
        "ProviderName": "Crypto-Pro GOST R 34.10-2001 Cryptographic Service Provider",
        "ProviderType": 75,
        "KeyLength": 512,
        "HashAlgorithms": ["GOST R 34.11-94"],
        "Description": "GOST 2001"
    }, {
        "ID": "60e9912c-68a8-4608-b1c2-0c6a074456e8",
        "GroupID": "648092d3-46a9-422a-9240-d32c58cc498b",
        "TypeID": 2,
        "ProviderName": "Crypto-Pro GOST R 34.10-2012 Cryptographic Service Provider",
        "ProviderType": 80,
        "KeyLength": 512,
        "HashAlgorithms": ["GR 34.11-2012 256"],
        "Description": "GOST 2012"
    }
],
"ActionPolicy": [{
    "DisplayName": "Выпуск маркера (вход в ЦИ)",
    "Uri": "http://dss.cryptopro.ru/identity/claims/action/issue",
    "Action": "Issue",
    "MfaRequired": false
}, {
    "DisplayName": "Подпись документа",
    "Uri": "http://dss.cryptopro.ru/identity/claims/action/signdocument",
    "Action": "SignDocument",
    "MfaRequired": false
}, {
    "DisplayName": "Подпись пакета документов",
    "Uri": "http://dss.cryptopro.ru/identity/claims/action/signdocuments",
    "Action": "SignDocuments",
    "MfaRequired": false
}, {
    "DisplayName": "Расшифрование документа",
    "Uri": "http://dss.cryptopro.ru/identity/claims/action/decryptdocument",
    "Action": "DecryptDocument",
    "MfaRequired": false
}, {
    "DisplayName": "Создание запроса на сертификат",
    "Uri": "http://dss.cryptopro.ru/identity/claims/action/createrequest",
    "Action": "CreateRequest",
    "MfaRequired": false
}, {
    "DisplayName": "Смена пин-кода закрытого ключа",
    "Uri": "http://dss.cryptopro.ru/identity/claims/action/changepin",
    "Action": "ChangePin",
    "MfaRequired": false
}, {
    "DisplayName": "Обновление сертификата",
    "Uri": "http://dss.cryptopro.ru/identity/claims/action/renewcertificate",
    "Action": "RenewCertificate",
    "MfaRequired": false
}, {
    "DisplayName": "Отзыв сертификата",
    "Uri": "http://dss.cryptopro.ru/identity/claims/action/revokecertificate",
    "Action": "RevokeCertificate",
    "MfaRequired": false
}, {
    "DisplayName": "Приостановление действия сертификата",
    "Uri": "http://dss.cryptopro.ru/identity/claims/action/holdcertificate",
    "Action": "HoldCertificate",
    "MfaRequired": false
}, {
    "DisplayName": "Возобновление действия сертификата",
    "Uri": "http://dss.cryptopro.ru/identity/claims/action/unholdcertificate",
    "Action": "UnholdCertificate",

```

```

        "MfaRequired": false
    }, {
        "DisplayName": "Удаление сертификата",
        "Uri": "http://dss.cryptopro.ru/identity/claims/action/deletecertificate",
        "Action": "DeleteCertificate",
        "MfaRequired": false
    }, {
        "DisplayName": "Доступ к закрытому ключу",
        "Uri": "http://dss.cryptopro.ru/identity/claims/action/privatekeyaccess",
        "Action": "PrivateKeyAccess",
        "MfaRequired": false
    }
],
"PinCodeMode": "Allow",
"TspServices": [{
    "Name": "TestTSP",
    "Title": "TestTSP",
    "Url": "http://TEST-DSS-W8R2/TSP/tsp.srf"
}],
"TransactionConfirmation": "NotSet",
"AllowedSignatureTypes": ["GOST3410", "CMS", "CAAdES", "XMLDSig", "MSOffice", "PDF"]
}

```

Параметры Удостоверяющего Центра (DSSCAPolicy)

Параметры Удостоверяющего Центра являются частью политики Сервиса Подписи. Политику Сервиса Подписи можно получить вызывая метод [/policy](#).

Параметры Удостоверяющего Центра определяют требования к создаваемому запросу на сертификат.

Описание полей DSSCAPolicy:

ПОЛЕ	ТИП	ОПИСАНИЕ
ID	int	Идентификатор Удостоверяющего Центра
Name	string	Отображаемое имя Удостоверяющего Центра
Active	bool	Доступен ли УЦ для создания запросов (фактически всегда равно true, так как сервер подписи отдаёт список только активных УЦ)
SNChangesEnable	bool	Разрешено ли редактировать шаблон запроса на сертификат (если false, то данные берутся из первого запроса на сертификат)
NamePolicy	List< SubjectNameComponent >	Шаблон имени. Содержит список компонентов различительного имени, их порядок и обязательность
EkuTemplates	Dictionary<string, List>	Словарь шаблонов сертификатов в формате {"Наименование шаблона", {Список OID}}
AllowUserMode	bool	Режим работы обработчика Удостоверяющего Центра.
CAType	DSSCAType	Тип модуля интеграции с Удостоверяющим Центром.

Описание полей SubjectNameComponent:

ПОЛЕ	ТИП	ОПИСАНИЕ
IsRequired	bool	Требовать обязательного заполнения компоненты имени.
Order	int	Порядковый номер в списке компонент.
OID	string	Объектный идентификатор компоненты имени.
Name	string	Понятное имя компоненты.
Value	string	Значение.
StringIdentifier	string	Строковый идентификатор компоненты

Типы подключенных Удостоверяющих Центров

НАИМЕНОВАНИЕ	КОД	ОПИСАНИЕ
CryptoProCA15Enroll	0	Модуль интеграции с КриптоПро УЦ 1.5

НАИМЕНОВАНИЕ	КОД	ОПИСАНИЕ
CryptoProCA20Enroll	1	Модуль интеграции с КриптоПро УЦ 2.0
DSSOutOfBandEnroll	2	Модуль интеграции со сторонним Удостоверяющим Центром (Offline).

Пример политики УЦ, возвращаемой Сервисом Подписи (метод [/policy](#))

```
"CAPolicy":
[
  {
    "ID": 11,
    "Name": "Out of Band",
    "Active": true,
    "AllowUserMode": false,
    "SNChangesEnable": true,
    "NamePolicy":
    [
      {
        "IsRequired": false,
        "Order": 2,
        "OID": "1.2.840.113549.1.9.1",
        "Name": "E-Mail",
        "Value": null,
        "StringIdentifier": "E"
      },
      {
        "IsRequired": false,
        "Order": 8,
        "OID": "1.2.643.3.131.1.1",
        "Name": "ИНН",
        "Value": null,
        "StringIdentifier": "INN"
      },
      {
        "IsRequired": false,
        "Order": 4,
        "OID": "2.5.4.7",
        "Name": "Населенный пункт",
        "Value": null,
        "StringIdentifier": "L"
      },
      {
        "IsRequired": false,
        "Order": 7,
        "OID": "1.2.643.100.1",
        "Name": "ОГРН",
        "Value": null,
        "StringIdentifier": "OGRN"
      },
      {
        "IsRequired": false,
        "Order": 5,
        "OID": "2.5.4.10",
        "Name": "Организация",
        "Value": null,
        "StringIdentifier": "O"
      },
      {
        "IsRequired": false,
        "Order": 6,
        "OID": "2.5.4.11",
        "Name": "Подразделение",
        "Value": null,
        "StringIdentifier": "OU"
      },
      {
        "IsRequired": false,
        "Order": 3,
        "OID": "2.5.4.8",
        "Name": "Регион",

```

```

        "Value": null,
        "StringIdentifier": "S"
    }, {
        "IsRequired": false,
        "Order": 9,
        "OID": "2.5.4.6",
        "Name": "Страна",
        "Value": null,
        "StringIdentifier": "C"
    }, {
        "IsRequired": true,
        "Order": 1,
        "OID": "2.5.4.3",
        "Name": "Общее имя",
        "Value": null,
        "StringIdentifier": "CN"
    }
],
"EKUTemplates": {
    "Временный сертификат администратора УЦ": ["1.2.643.2.2.34.2", "1.2.643.2.2.34.4",
"1.3.6.1.5.5.7.3.2"],
    "Временный сертификат оператора УЦ": ["1.2.643.2.2.34.2", "1.2.643.2.2.34.5",
"1.3.6.1.5.5.7.3.2"],
    "Временный сертификат пользователя УЦ": ["1.2.643.2.2.34.2 ", "1.3.6.1.5.5.7.3.2"],
    "Сертификат пользователя УЦ": ["1.2.643.2.2.34.6", "1.3.6.1.5.5.7.3.2"]
},
"CAType": "DSSOutOfBandEnroll",
"ValidationMode": "CertificateAuthority"
}
]

```

Параметры криптопровайдеров (DSSCSPPolicy)

Параметры криптопровайдеров являются частью политики Сервиса Подписи. Политику Сервиса Подписи можно получить вызывав метод `/policy`.

Параметры криптопровайдеров определяют алгоритм закрытого ключа сертификата пользователя.

Описание полей DSSCSPPolicy:

ПОЛЕ	ТИП	ОПИСАНИЕ
ID	guid	ID криптопровайдера
GroupID	guid	ID группы криптопровайдера в БД
TypeID	int	Метод хранения закрытых ключей пользователей
ProviderName	string	Название криптопровайдера (Crypto API)
ProviderType	int	Тип криптопровайдера (Crypto API)
KeyLength	int	Длина ключа
HashAlgorithms	List<string>	Хэш алгоритмы, поддерживаемые криптопровайдером
Description	string	Отображаемое имя криптопровайдера

Методы хранения ключей пользователей (TypeID)

ТИП	ОПИСАНИЕ
1	Хранение ключей пользователей в HSM
2	Хранение ключей пользователей в базе данных, защищёнными на мастер ключе HSM
3	Храенение ключей на стороне пользователя

Типы криптопровайдеров (ProviderType)

ТИП	ОПИСАНИЕ
75	ГОСТ Р 34.10-2001
80	ГОСТ Р 34.10-2012 (256)
81	ГОСТ Р 34.10-2012 (512)

Пример политики криптопровайдеров, возвращаемой Сервисом Подписи (метод `/policy`)


```
CSPsPolicy":  
[  
  {  
    "ID": "67e6f39d-c6c5-4ce6-9535-4e22bae84786",  
    "GroupID": "e8e67f9e-7eed-4116-ad98-20582e4d766e",  
    "TypeID": 2,  
    "ProviderName": "Crypto-Pro GOST R 34.10-2001 Cryptographic Service Provider",  
    "ProviderType": 75,  
    "KeyLength": 512,  
    "HashAlgorithms": ["GOST R 34.11-94"],  
    "Description": "GOST 2001"  
  },  
  {  
    "ID": "60e9912c-68a8-4608-b1c2-0c6a074456e8",  
    "GroupID": "648092d3-46a9-422a-9240-d32c58cc498b",  
    "TypeID": 2,  
    "ProviderName": "Crypto-Pro GOST R 34.10-2012 Cryptographic Service Provider",  
    "ProviderType": 80,  
    "KeyLength": 512,  
    "HashAlgorithms": ["GR 34.11-2012 256"],  
    "Description": "GOST 2012"  
  }  
]
```

Параметры подписи документа

ИМЯ	ЗНАЧЕНИЕ	ОПИСАНИЕ
SignatureType (r)	CAdES	Подпись формата CAdES-BES, CAdES-T, CAdES-X Long Type 1
	CMS	Подпись формата CAdES-BES
	PDF	Подпись PDF документов
	MSOffice	Подпись документов MS Word и Excel
	XMLDSig	Подпись документа в формате XMLDSig
	GOST3410	Электронная подпись по ГОСТ Р 34.10 - 2001 или ГОСТ Р 34.10 - 2012
CertificateId (r)		Идентификатор сертификата подписи
DocumentInfo (o)		Сведения о документе. Например, имя файла
DocumentType (o)		Тип документа. Параметр используется для визуализации документа

Типы подписи

ИМЯ	КОД	ЗНАЧЕНИЕ
XMLDSig	0	Подпись документа в формате XMLDSig
GOST3410	1	Электронная подпись по ГОСТ Р 34.10 - 2001 или ГОСТ Р 34.10 - 2012
CAdES	2	Подпись формата CAdES-BES, CAdES-T, CAdES-X Long Type 1
PDF	3	Подпись PDF документов
MSOffice	4	Подпись документов MS Word и Excel
CMS	5	Подпись формата CAdES-BES

Обычно в параметре **DocumentType** указывают расширение подписываемого файла. Допустим также любой другой строковый идентификатор. С данным идентификатором должен быть связан плагин для отображения документов на сервере DSS.

В зависимости от выбранного формата подписи необходимо указать сопутствующие параметры. При подписи без подтверждения вторым фактором аутентификации параметры передаются в поле `Parameters` структуры [Document](#).

При подписи с подтверждением вторым фактором аутентификации параметры передаются в поле `Parameters` структуры [Transaction](#).

Внимание!

Имена параметров, которые передаются в поле `Parameters` регистрозависимы.

Подпись формата CAdES или CMS

ИМЯ	ЗНАЧЕНИЕ	ОПИСАНИЕ
CADESType	BES	Подпись в формате CAdES-BES
	T	Подпись в формате CAdES-T
	XLT1	Подпись в формате CAdES-X Long Type 1
IsDetached (o)	true/false	Отделённая/присоединённая подпись. По умолчанию false
TSPAddress (r)		Адрес TSP службы (используется только для формата T, XLT1)
Hash (o)	true/false	Подпись значения хэш-функции ГОСТ Р 34.11 - 94 В поле Content должно быть передано значение хэш-функции от документа.
CmsSignatureType (o)	sign	Подпись документа
	cosign	Соподпись документа
	counterSign	Заверяющая подпись документа
ContentEncoding (o)	base64	Содержимое документа закодировано в Base64
	binary	Содержимое документа в двоичном представлении
OriginalDocument (o)		Содержимое исходного документа (используется при соподписи подписи). При создании пакетной отделённой соподписи (как самого документа, так и его хэш-значения) содержимое исходного документа (или его хэш-значение) передаётся через структуру DocumentContent (в поле OriginalContent), параметр OriginalDocument не используется.
SignatureIndex (o)		Индекс подписи, для которой создается заверяющая подпись

Подпись формата PDF

ИМЯ	ЗНАЧЕНИЕ	ОПИСАНИЕ
PDFFormat	CMS	Подпись PDF документов с использованием формата PKCS7
	CAdEST	Подпись PDF документов с использованием формата CAdES-T
	CAdES	Подпись PDF документов с использованием формата CAdES-X Long Type 1
PDFReason (r)		Цель подписания документа
PDFLocation (r)		Место подписания документа
PdfSignatureAppearance (o)		Строковое представление шаблона видимой (отображаемой) PDF-подписи
	1	Идентификатор шаблона видимой (отображаемой) PDF-подписи Простой текстовый шаблон

Имя	ЗНАЧЕНИЕ	ОПИСАНИЕ
PdfSignatureTemplateId(o)	2	Идентификатор шаблона видимой (отображаемой) PDF-подписи Шаблон с логотипом и текстом
	3	Идентификатор шаблона видимой (отображаемой) PDF-подписи Шаблон в виде изображения
PDFCertificationLevel(o)		Уровень сертификации подписи. Описывает, в зависимости от уровня, тип изменений, которые можно вносить в документ. Возможные значение описаны ниже

PDFCertificationLevel

NOT_CERTIFIED	Подпись для утверждения
CERTIFIED_NO_CHANGES_ALLOWED	Сертифицирующая подпись, после сертификации изменения запрещены
CERTIFIED_FORM_FILLING	Сертифицирующая подпись, после сертификации разрешено заполнение полей форм и использование цифровых подписей
CERTIFIED_FORM_FILLING_AND_ANNOTATIONS	Сертифицирующая подпись, после сертификации разрешены комментарии, заполнение полей форм и использование цифровых подписей

Примечание

При создании видимой (отображаемой) подписи необходимо указание параметров PdfSignatureTemplateId и PdfSignatureAppearance

Подпись формата MS Office

Дополнительный параметры подписи отсутствуют

Подпись формата XMLDSig

Имя	ЗНАЧЕНИЕ	ОПИСАНИЕ
XMLDSigType	XMLEnveloped	Вложенная XMLDSig подпись
	XMLEnveloping	Присоединённая XMLDSig подпись
	XMLTemplate	XMLDSig подпись по шаблону

Подпись формата ГОСТ Р 34.10

Имя	ЗНАЧЕНИЕ	ОПИСАНИЕ
Hash	true/false	Подпись значения хэш-функции ГОСТ Р 34.11 - 94 В поле документ должно быть передано значение хэш-функции от документа

Дополнительные параметры операции шифрования

ПАРАМЕТР	ТИП	ОПИСАНИЕ
CipherOid	string	Объектный идентификатор симметричного ключа шифрования. Доступные значения: "1.2.643.2.2.31.1" (значение по умолчанию), "1.2.643.7.1.2.5.1.1".
UseFssScenario	string	Параметр, присутствие которого показывает, что необходимо использовать сценарий шифрования/расшифрования ФСС.

Дополнительные параметры операции расшифрования

ПАРАМЕТР	ТИП	ОПИСАНИЕ
UseFssScenario	string	Параметр, присутствие которого показывает, что необходимо использовать сценарий шифрования/расшифрования ФСС. В случае, если используется данный параметр, необходимо явно передавать идентификатор сертификата расшифрования, так как данный сценарий не поддерживает автоматический подбор сертификата.

Данные для создания запроса на сертификат (CertificateRequest)

Данные для создания запроса на сертификат передаются в структуре CertificateRequest в метод [/requests](#)

ПОЛЕ	ТИП	ОПИСАНИЕ
AuthorityId	int	Идентификатор Удостоверяющего Центра, к которому направлен запрос на сертификат.
PinCode	string	ПИН-код для доступа к закрытому ключу сертификата.
Template	string	Идентификатор шаблона сертификата, по которому создаётся запрос.
DistinguishedName	IDictionary<string, string>	Набор компонентов различительного имени субъекта в виде пар {oid, значение}.
RawDistinguishedName	string	Строковое представление различительного имени, закодированное в соответствии с RFC 1779 .
Parameters	IDictionary<RequestParams, string>	Словарь дополнительных параметров запроса.

Подробнее о выпуске запроса на сертификат смотреть раздел: [Выпуск сертификата](#)

Сертификат (DSSCertificateEx)

Объект данного типа возвращается при:

- Получении списка сертификатов (метод [/certificates](#))
- Получении сертификата по идентификатору (метод [/certificates](#))

ПОЛЕ	ТИП	ОПИСАНИЕ
ID	int	Идентификатор сертификата на Сервисе Подписи
DName	string	Различительное имя субъекта
CertificateBase64	string	Сертификат в формате X.509 закодированный в Base64
Status	CertificateStatus	Статус сертификата
IsDefault	bool	Сертификат назанчен по умолчанию
CertificateAuthorityID	int	Идентификатор удостоверяющего центра, к которому направлен запрос на сертификат
CspID	guid	Идентификатор криптопровайдера
HashAlgorithms	List<string>	Список поддерживаемых хэш-алгоритмов
ProviderName	string	Имя криптопровайдера, использованного при создании закрытого ключа
ProviderType	int	Тип криптопровайдера, использованного при создании закрытого ключа
PrivateKeyNotBefore	DateTime	Начало срока действия закрытого ключа сертификата
PrivateKeyNotAfter	DateTime	Окончание срока действия закрытого ключа сертификата
HasPin	bool	Наличие ПИН-кода на закрытый ключ сертификата
FriendlyName	string	Дружественное имя сертификата

Сведения об Удостоверяющем Центре, обработавшем запрос на сертификат, можно получить из [Политики Сервиса Подписи](#), по идентификатору `CertificateAuthorityID`.

Сведения о криптопровайдере, использованном при создании закрытого ключа, можно получить из [Политики Сервиса Подписи](#), по идентификатору `CspID`.

Пример


```

{
  "CertificateType": "ServerSide",
  "ID": 14,
  "DName": "CN=idonly",
  "CertificateBase64": "MIIDCDCCAreAwIBAgITEgA ... xX8V2DhYHzugFV8td4DaneG2/gno7T6Alohp6CF/yOu",
  "Status": {
    "Value": "ACTIVE",
    "RevocationInfo": null,
    "PinCode": null,
    "ActiveCertId": 0
  },
  "IsDefault": false,
  "CertificateAuthorityID": 11,
  "CspID": "e8e67f9e-7eed-4116-ad98-20582e4d766e",
  "HashAlgorithms": ["GOST R 34.11-94"],
  "ProviderName": null,
  "ProviderType": 0,
  "PrivateKeyNotBefore": null,
  "PrivateKeyNotAfter": null,
  "HasPin": false,
  "FriendlyName": ""
}

```

Запрос на сертификат (DSSCertRequest)

Созданный запрос на сертификат.
Объект данного типа возвращается при:

- Создании запроса на сертификат (метод [/requests](#))
- Получении списка запросов на сертификаты (метод [/requests](#))
- Получении запроса на сертификат по идентификатору (метод [/requests](#))

Подробнее о выпуске запроса на сертификат смотреть раздел [Создание запроса на сертификат](#)

ПОЛЕ	ТИП	ОПИСАНИЕ
ID	int	Идентификатор запроса на Сервисе Подписи
Base64Request	string	Запрос на сертификат в формате BASE64
CertificateAuthorityID	int	Идентификатор удостоверяющего центра, к которому направлен запрос на сертификат
CAResultID	string	Идентификатор запроса на сертификат в удостоверяющем центре
CADisplayName	string	Отображаемое имя УЦ
DistName	string	Различительное имя субъекта
Subject	string	Общее имя субъекта
Status	DSSRequestStatusEnum	Статус запроса в УЦ
CertificateID	int	Идентификатор сертификата, с которым связан запрос.
RequestType	CAResultTypeEnum	Тип запроса к удостоверяющему центру.
GroupID	guid	Идентификатор криптопровайдера

Сведения об Удостоверяющем Центре, обработавшем запрос на сертификат, можно получить из [Политики Сервиса Подписи](#), по идентификатору `CertificateAuthorityID`.

Сведения о криптопровайдере, использованном при создании закрытого ключа, можно получить из [Политики Сервиса Подписи](#), по идентификатору `GroupID`.

Статус сертификата (CertificateStatus)

Структура входит в состав сведений о сертификате ((DSSCertificateEx)[certstruct.md]), а так же используется при отзыве, приостановлении и восстановлении сертификата через метод /certificates/{cert_id}/status

ПОЛЕ	ТИП	ОПИСАНИЕ
Value	DSSCertificateStatusEnum	Статус сертификата
RevocationInfo	RevocationInfo	Сведения об отзыве/приостановлении сертификата
PinCode	string	ПИН-код на закрытый ключ действующего сертификата
ActiveCertId	int	Идентификатор действующего сертификата

Поля `ActiveCertId` и `PinCode` используются при отзыве, приостановлении и восстановлении сертификата через метод /certificates/{cert_id}/status

Статус сертификата (DSSCertificateStatusEnum)

ИМЯ	КОД	ЗНАЧЕНИЕ
NULL	0	Нет информации о статусе сертификата
ACTIVE	1	Сертификат действителен
REVOKED	2	Сертификат отозван
HOLD	3	Действие сертификата приостановлено
NOT_VALID	4	Недействительный
OUT_OF_ORDER	6	Модуль интеграции с Удостоверяющим Центром не смог обработать сертификат: - получить статус - отозвать - восстановить - и т.п

Информация об отзыве сертификата (RevocationInfo)

поле	тип	описание
RevocationReason	CertRevokeReasonEnum	Причина отзыва
RevocationDate	DateTime	Дата отзыва
RevocationComments	string	Комментарии администратора
UnholdDate	DateTime	Дата возобновления действия сертификата
DistName	string	Различительное имя субъекта
RequestDate	DateTime	Дата подачи запроса
ApproveDate	DateTime	Дата рассмотрения запроса
UnholdAction	string	Действие, которое требуется выполнить после наступления даты приостановления
SerialNumber	string	Серийный номер отзываемого сертификата
SignedRequest	string	Запрос на отзыв/приостановление/восстановление, сформированный на клиенте

Причины отзыва сертификата (CertRevokeReasonEnum)

имя	код	значение
CRL_REASON_UNSPECIFIED	0	Причина неизвестна.
CRL_REASON_KEY_COMPROMISE	1	Компрометация ключей.
CRL_REASON_CA_COMPROMISE	2	Компрометация Центра Сертификации
CRL_REASON_AFFILIATION_CHANGED	3	Имя пользователя или другая информация в сертификате изменена, но нет причины полагать, что секретный ключ скомпрометирован.
CRL_REASON_SUPERSEDED	4	Сертификат заменен другим, но нет причины полагать, что секретный ключ скомпрометирован.
CRL_REASON_CESSATION_OF_OPERATION	5	Сертификат более не нужен для целей, которых он выдавался, но нет причины полагать, что секретный ключ скомпрометирован.
CRL_REASON_CERTIFICATE_HOLD	6	Действие сертификата приостановлено.
CRL_REASON_REMOVE_FROM_CRL	8	Служебный код для возобновления действия сертификата, действие которого ранее было приостановлено. (CryptoProCA20 Version)

Сведения об отзыве/приостановлении/восстановлении (DSSRevRequest)

ПОЛЕ	ТИП	ОПИСАНИЕ
ID	string	идентификатор запроса в УЦ
Base64Request	string	запрос на сертификат в формате BASE64
CertificateAuthorityID	int	идентификатор УЦ
Status	DSSRequestStatusEnum	статус запроса
RevInfo	RevocationInfo	информацию о запросе

Данные для смены ПИН-кода (RequestChangePin)

поле	тип	описание
NewPin	string	Новый ПИН-код
OldPin	string	Текущий ПИН-код

Данные для назначения сертификата по умолчанию (DefaultProperty)

ПОЛЕ	ТИП	ОПИСАНИЕ
Default	bool	Назначить/сбросить сертификат по умолчанию

Данные для назначения дружественного имени (FriendlyNameProperty)

ПОЛЕ	ТИП	ОПИСАНИЕ
FriendlyName	string	Дружественное имя

Данные о подписываемом/расшифровываемом документе (Document)

Данные для подписания, усовершенствования, расшифрования или зашифрования документа передаются в структуре Document в метод [/documents](#)

ПОЛЕ	ТИП	ОПИСАНИЕ
Content	string	Документ (в кодировке Base64)
Name	string	Название документа
Signature	DocumentSignature	Информация о подписи документа.
Encryption	DocumentEncryption	Информация о параметрах шифрования документа.
Decryption	DocumentDecryption	Информация о параметрах расшифрования документа.
Hash	DocumentHash	Информация о параметрах хэширования документа.

Параметры подписи (DocumentSignature)

ПОЛЕ	ТИП	ОПИСАНИЕ
Type	SignatureType	Тип подписи
Parameters	Dictionary< SignatureParams , string>	Дополнительные параметры подписи
CertificateId	int	Идентификатор сертификата
PinCode	string	Пин-код для доступа к закрытому ключу сертификата

Параметры зашифрования (DocumentEncryption)

ПОЛЕ	ТИП	ОПИСАНИЕ
Type	EncryptionType	Тип шифрования.
Parameters	Dictionary<string, string>	Дополнительные параметры шифрования. Список доступных параметров можно найти здесь .
Certificates	List<byte[]>	Список сертификатов получателей.

Параметры расшифрования (DocumentDecryption)

ПОЛЕ	ТИП	ОПИСАНИЕ
Type	EncryptionType	Тип шифрования.
CertificateId	int	Идентификатор сертификата получателя.

ПОЛЕ	ТИП	ОПИСАНИЕ
PinCode	string	ПИН-код для доступа к закрытому ключу сертификата.

Параметры хэширования (DocumentHash)

ПОЛЕ	ТИП	ОПИСАНИЕ
Parameters	Dictionary< SignatureParams , string>	Параметры хэширования

В словаре необходимо передать значение ключа `HashAlgorithm`, которое может принимать значения:

- GOST R 34.11-94
- GR 34.11-2012 256
- GR 34.11-2012 512
- SHA-1
- SHA-256
- SHA-384
- SHA-512
- MD5

Параметры подписи документа

ИМЯ	ЗНАЧЕНИЕ	ОПИСАНИЕ
SignatureType (r)	CAAdES	Подпись формата CAAdES-BES, CAAdES-T, CAAdES-X Long Type 1
	CMS	Подпись формата CAAdES-BES
	PDF	Подпись PDF документов
	MSOffice	Подпись документов MS Word и Excel
	XMLDSig	Подпись документа в формате XMLDSig
	GOST3410	Электронная подпись по ГОСТ Р 34.10 - 2001 или ГОСТ Р 34.10 - 2012
CertificateId (r)		Идентификатор сертификата подписи
DocumentInfo (o)		Сведения о документе. Например, имя файла
DocumentType (o)		Тип документа. Параметр используется для визуализации документа

Типы подписи

ИМЯ	КОД	ЗНАЧЕНИЕ
XMLDSig	0	Подпись документа в формате XMLDSig
GOST3410	1	Электронная подпись по ГОСТ Р 34.10 - 2001 или ГОСТ Р 34.10 - 2012

ИМЯ	КОД	ЗНАЧЕНИЕ
CAdES	2	Подпись формата CAdES-BES, CAdES-T, CAdES-X Long Type 1
PDF	3	Подпись PDF документов
MSOffice	4	Подпись документов MS Word и Excel
CMS	5	Подпись формата CAdES-BES

Обычно в параметре **DocumentType** указывают расширение подписываемого файла. Допустим также любой другой строковый идентификатор. С данным идентификатором должен быть связан плагин для отображения документов на сервере DSS.

В зависимости от выбранного формата подписи необходимо указать сопутствующие параметры. При подписи без подтверждения вторым фактором аутентификации параметры передаются в поле `Parameters` структуры [Document](#).

При подписи с подтверждением вторым фактором аутентификации параметры передаются в поле `Parameters` структуры [Transaction](#).

Внимание!

Имена параметров, которые передаются в поле `Parameters`, регистрозависимы.

Подпись формата CAdES или CMS

ИМЯ	ЗНАЧЕНИЕ	ОПИСАНИЕ
CADESType	BES	Подпись в формате CAdES-BES
	T	Подпись в формате CAdES-T
	XLT1	Подпись в формате CAdES-X Long Type 1
IsDetached (o)	true/false	Отделённая/присоединённая подпись. По умолчанию false
TSPAddress (r)		Адрес TSP службы (используется только для формата T, XLT1)
Hash (o)	true/false	Подпись значения хэш-функции ГОСТ Р 34.11 - 94 В поле Content должно быть передано значение хэш-функции от документа.
CmsSignatureType (o)	sign	Подпись документа
	cosign	Соподпись документа
	counterSign	Заверяющая подпись документа
ContentEncoding (o)	base64	Содержимое документа закодировано в Base64
	binary	Содержимое документа в двоичном представлении

ИМЯ	ЗНАЧЕНИЕ	ОПИСАНИЕ
OriginalDocument (o)		Содержимое исходного документа (используется при соподписи подписи). При создании пакетной отделинной соподписи (как самого документа, так и его хэш-значения) содержимое исходного документа (или его хэш-значение) передаётся через структуру DocumentContent (в поле OriginalContent), параметр OriginalDocument не используется.
SignatureIndex (o)		Индекс подписи, для которой создается заверяющая подпись

Подпись формата PDF

ИМЯ	ЗНАЧЕНИЕ	ОПИСАНИЕ
PDFFormat	CMS	Подпись PDF документов с использованием формата PKCS7
	CAdEST	Подпись PDF документов с использованием формата CAdES-T
	CAdES	Подпись PDF документов с использованием формата CAdES-X Long Type 1
PDFReason (r)		Цель подписания документа
PDFLocation (r)		Место подписания документа
PdfSignatureAppearance (o)		Строковое представление шаблона видимой (отображаемой) PDF-подписи
PdfSignatureTemplateId (o)	1	Идентификатор шаблона видимой (отображаемой) PDF-подписи Простой текстовый шаблон
	2	Идентификатор шаблона видимой (отображаемой) PDF-подписи Шаблон с логотипом и текстом
	3	Идентификатор шаблона видимой (отображаемой) PDF-подписи Шаблон в виде изображения
PDFCertificationLevel(o)		Уровень сертификации подписи. Описывает, в зависимости от уровня, тип изменений, которые можно вносить в документ. Возможные значение описаны ниже

PDFCertificationLevel

NOT_CERTIFIED	Подпись для утверждения
CERTIFIED_NO_CHANGES_ALLOWED	Сертифицирующая подпись, после сертификации изменения запрещены
CERTIFIED_FORM_FILLING	Сертифицирующая подпись, после сертификации разрешено заполнение полей форм и использование цифровых подписей
CERTIFIED_FORM_FILLING_AND_ANNOTATIONS	Сертифицирующая подпись, после сертификации разрешены комментарии, заполнение полей форм и использование цифровых подписей

Примечание

При создании видимой (отображаемой) подписи необходимо указание параметров PdfSignatureTemplateId и PdfSignatureAppearance

Подпись формата MS Office

Дополнительный параметры подписи отсутствуют

Подпись формата XMLDSig

ИМЯ	ЗНАЧЕНИЕ	ОПИСАНИЕ
XMLDSigType	XMLEnveloped	Вложенная XMLDSig подпись
	XMLEnveloping	Присоединённая XMLDSig подпись
	XMLTemplate	XMLDSig подпись по шаблону

Подпись формата ГОСТ Р 34.10

ИМЯ	ЗНАЧЕНИЕ	ОПИСАНИЕ
Hash	true/false	Подпись значения хэш-функции ГОСТ Р 34.11 - 94 В поле документ должно быть передано значение хэш-функции от документа

Данные о подписываемом/расшифровываемом пакете документов (DocumentPackage)

Данные для подписания пакета документов, передаются в структуре DocumentPackage в метод [/documents](#)

Все документы в пакете будут подписаны однотипно (в соответствии с указанными [параметрами подписи](#)).

ПОЛЕ	ТИП	ОПИСАНИЕ
Documents	IList<DocumentContent>	Пакет документов
Signature	DocumentSignature	Информация о подписи документа.

Содержимое документа (DocumentContent)

ПОЛЕ	ТИП	ОПИСАНИЕ
Content	byte[]	Подписываемые данные
Name	string	Название документа
OriginalContent	byte[]	Содержимое исходного документа

Поле `OriginalContent` заполняется в случае отделённой соподписи.

Параметры подписи (DocumentSignature)

ПОЛЕ	ТИП	ОПИСАНИЕ
SignatureType	SignatureType	Тип подписи
Parameters	Dictionary<SignatureParams, string>	Дополнительные параметры подписи
CertificateId	int	Идентификатор сертификата
PinCode	string	Пин-код для доступа к закрытому ключу сертификата

Результат подписания пакета документов (DSSSignDocumentResponse)

Результат подписания пакета документов </documents/packagesignature>

поле	тип	описание
Results	List<string>	Список подписанных документов.
Errors	List<DSSFault>	Список ошибок, произошедших при подписи пакета документов.

Описание полей структуры DSSFault

поле	тип	описание
Message	string	Описание ошибки.

Если при подписании одного или нескольких документов возникли ошибки, то в ответе сервиса в поле `Results` на соответствующей позиции будет выставлено значение `null`, а в поле `Error` на соответствующей позиции будет сообщение об ошибке.

Транзакция (Transaction)

Данные для создания транзакции на Сервисе Подписи передаются в структуре Transaction в метод [/transactions](#)

ПОЛЕ	ТИП	ОПИСАНИЕ
OperationCode	DSSAction	Идентификатор операции, для которой формируется транзакция.
Document	byte[]	Данные транзакции.
Parameters	TransactionParameter []	Массив параметров транзакции.
Documents	List< DocumentContent >	Содержимое документов, входящих в пакет при пакетной подписи.

Параметры пакетной подписи (DocumentContent)

ПОЛЕ	ТИП	ОПИСАНИЕ
Name	string	Название документа.
Content	byte[]	Содержимое документа.

Параметры транзакций Сервиса Подписи

Транзакция - операция на Сервисе Подписи, которая должна быть подтверждена вторым фактором аутентификации (одноразовым паролем в SMS, myDSS и т.п).

Существуют следующие типы транзакций на Сервисе Подписи:

- Подпись документа
- Подпись пакета документов
- Создание запроса на сертификат
- Расшифрование документа
- Отзыв сертификата
- Приостановление действия серификата
- Восстановление действия серификата
- Обновление сертификата
- Смена ПИН-кода на закрытый ключ
- Удаление сертификата

Примечание

Параметры транзакции совпадают с параметрами вызова аналогичной операции без подтверждения вторым фактором аутентификации. Например: параметры, передаваемые при создании транзакции подписи, аналогичны параметра при подписи документа без подтверждения.

Коды типов транзакций приведены в разделе [Типы транзакций](#). Код транзакции задаётся в параметре `OperationCode` структуры `Transaction`.

Ниже приводятся списки параметров для каждого типа транзакции. Параметры транзакции передаются в `Parameters` структуры `Transaction`.

Подпись документа и пакета документов

Код транзакции 2 (SignDocument) или 4 (SignDocuments)

Параметры подписи документа

Имя	Значение	Описание
SignatureType (r)	CAdES	Подпись формата CAdES-BES, CAdES-T, CAdES-X Long Type 1
	CMS	Подпись формата CAdES-BES
	PDF	Подпись PDF документов
	MSOffice	Подпись документов MS Word и Excel
	XMLDSig	Подпись документа в формате XMLDSig
	GOST3410	Электронная подпись по ГОСТ Р 34.10 - 2001 или ГОСТ Р 34.10 - 2012
CertificateId (r)		Идентификатор сертификата подписи
DocumentInfo (o)		Сведения о документе. Например, имя файла
DocumentType (o)		Тип документа. Параметр используется для визуализации документа

Типы подписи

ИМЯ	КОД	ЗНАЧЕНИЕ
XMLDSig	0	Подпись документа в формате XMLDSig
GOST3410	1	Электронная подпись по ГОСТ Р 34.10 - 2001 или ГОСТ Р 34.10 - 2012
CAdES	2	Подпись формата CAdES-BES, CAdES-T, CAdES-X Long Type 1
PDF	3	Подпись PDF документов
MSOffice	4	Подпись документов MS Word и Excel
CMS	5	Подпись формата CAdES-BES

Обычно в параметре **DocumentType** указывают расширение подписываемого файла. Допустим также любой другой строковый идентификатор. С данным идентификатором должен быть связан плагин для отображения документов на сервере DSS.

В зависимости от выбранного формата подписи необходимо указать сопутствующие параметры. При подписи без подтверждения вторым фактором аутентификации параметры передаются в поле `Parameters` структуры [Document](#).

При подписи с подтверждением вторым фактором аутентификации параметры передаются в поле `Parameters` структуры [Transaction](#).

Внимание!

Имена параметров, которые передаются в поле `Parameters` регистрозависимы.

Подпись формата CAdES или CMS

ИМЯ	ЗНАЧЕНИЕ	ОПИСАНИЕ
CADESType	BES	Подпись в формате CAdES-BES
	T	Подпись в формате CAdES-T
	XLT1	Подпись в формате CAdES-X Long Type 1
IsDetached (o)	true/false	Отделённая/присоединённая подпись. По умолчанию false
TSPAddress (r)		Адрес TSP службы (используется только для формата T, XLT1)
Hash (o)	true/false	Подпись значения хэш-функции ГОСТ Р 34.11 - 94 В поле Content должно быть передано значение хэш-функции от документа.
CmsSignatureType (o)	sign	Подпись документа
	cosign	Соподпись документа
	counterSign	Заверяющая подпись документа

ИМЯ	ЗНАЧЕНИЕ	ОПИСАНИЕ
ContentEncoding (o)	base64	Содержимое документа закодировано в Base64
	binary	Содержимое документа в двоичном представлении
OriginalDocument (o)		Содержимое исходного документа (используется при соподписи подписи). При создании пакетной отделинной соподписи (как самого документа, так и его хэш-значения) содержимое исходного документа (или его хэш-значение) передаётся через структуру DocumentContent (в поле OriginalContent), параметр OriginalDocument не используется.
SignatureIndex (o)		Индекс подписи, для которой создается заверяющая подпись

Подпись формата PDF

ИМЯ	ЗНАЧЕНИЕ	ОПИСАНИЕ
PDFFormat	CMS	Подпись PDF документов с использованием формата PKCS7
	CAdEST	Подпись PDF документов с использованием формата CAdES-T
	CAdES	Подпись PDF документов с использованием формата CAdES-X Long Type 1
PDFReason (r)		Цель подписания документа
PDFLocation (r)		Место подписания документа
PdfSignatureAppearance (o)		Строковое представление шаблона видимой (отображаемой) PDF-подписи
PdfSignatureTemplateId (o)	1	Идентификатор шаблона видимой (отображаемой) PDF-подписи Простой текстовый шаблон
	2	Идентификатор шаблона видимой (отображаемой) PDF-подписи Шаблон с логотипом и текстом
	3	Идентификатор шаблона видимой (отображаемой) PDF-подписи Шаблон в виде изображения
PDFCertificationLevel(o)		Уровень сертификации подписи. Описывает, в зависимости от уровня, тип изменений, которые можно вносить в документ. Возможные значение описаны ниже

PDFCertificationLevel

NOT_CERTIFIED	Подпись для утверждения
CERTIFIED_NO_CHANGES_ALLOWED	Сертифицирующая подпись, после сертификации изменения запрещены
CERTIFIED_FORM_FILLING	Сертифицирующая подпись, после сертификации разрешено заполнение полей форм и использование цифровых подписей
CERTIFIED_FORM_FILLING_AND_ANNOTATIONS	Сертифицирующая подпись, после сертификации разрешены комментарии, заполнение полей форм и использование цифровых подписей

Примечание

При создании видимой (отображаемой) подписи необходимо указание параметров **PdfSignatureTemplateId** и **PdfSignatureAppearance**

Подпись формата MS Office

Дополнительный параметры подписи отсутствуют

Подпись формата XMLDSig

ИМЯ	ЗНАЧЕНИЕ	ОПИСАНИЕ
XMLDSigType	XMLEnveloped	Вложенная XMLDSig подпись
	XMLEnveloping	Присоединённая XMLDSig подпись
	XMLTemplate	XMLDSig подпись по шаблону

Подпись формата ГОСТ Р 34.10

ИМЯ	ЗНАЧЕНИЕ	ОПИСАНИЕ
Hash	true/false	Подпись значения хэш-функции ГОСТ Р 34.11 - 94 В поле документ должно быть передано значение хэш-функции от документа

Создание запроса на сертификат

Код транзакции 16 (CreateRequest)

ПАРАМЕТРЫ ТРАНЗАКЦИИ	ОПИСАНИЕ
CAId	Идентификатор удостоверяющего центра
CertTemplateOid	Идентификатор шаблона сертификата Используется при выпуске сертификата на КриптоПро УЦ 2.0. Параметр взаимоисключающий с <code>EkuString</code>
CertSubjectName	Различительное имя субъекта в строковом представлении
EkuString	Расширение Enhanced Key Usage. Параметр взаимоисключающий с <code>CertTemplateOid</code>
GroupId (o)	Идентификатор криптопровайдера.

Подробнее о создании запроса на сертификат читать в разделе [Создание запроса на сертификат](#)

Расшифрование документа

Код транзакции 8 (DecryptDocument)

Отзыв сертификата

Код транзакции 128 (RevokeCertificate)

Приостановление действия сертификата

Код транзакции 256 (HoldCertificate)

Восстановление действия сертификата

Код транзакции 512 (UnholdCertificate)

Обновление сертификата

Код транзакции 64 (RenewCertificate)

Смена ПИН-кода на закрытый ключ

Код транзакции 32 (ChangePin)

Удаление сертификата

Код транзакции 1024 (DeleteCertificate)

Типы транзакций (DSSActions)

Транзакция - операция на Сервисе Подписи или Центре Идентификации, которая может быть подтверждена вторым фактором аутентификации (одноразовым паролем в SMS, myDSS и т.п).

ИМЯ	КОД	ОПИСАНИЕ
Issue	1	Подтверждение входа пользователя
SignDocument	2	Подпись документа
SignDocuments	4	Подпись пакета документов
DecryptDocument	8	Расшифрование документа
CreateRequest	16	Создание запроса на сертификат
ChangePin	32	Смена ПИН-кода на закрытый ключ сертификата
RenewCertificate	64	Перевыпуск сертификата
RevokeCertificate	128	Отзыв сертификата
HoldCertificate	256	Приостановление действия сертификата
UnholdCertificate	512	Восстановление действия сертификата
DeleteCertificate	1024	Удаление сертификата
PrivateKeyAccess	2048	Создание ключа согласования (CloudCSP)

Коды транзакций используются:

- в **Центре Идентификации DSS** при назначении пользователю операций, требующих двухфакторной аутентификации. (метод /ums/{userId}/operationpolicy)
- в **Сервисе Подписи** при создании транзакций ([/transactions](#)).

Список операций, требующих подтверждения, может быть преднастроен Администратором на сервере DSS. Список операций может быть настроен на разных уровнях:

- для всех пользователей.
- для групп пользователей.

Имена транзакций используются Администратором DSS при настройке Центра Идентификации. Например, когда требуется задать список обязательных для подтверждения операций всеми пользователями DSS:

Пример:

```
Set-DssConfirmationPolicy -OpActions Issue,SignDocument -AllowChangeByOperator 0
```


Форматы представления сертификата и запроса на сертификат (DSSCertificateFormatEnum)

имя	код	описание
RAW	1	Сертификат в формате X509 Запрос на сертификат в формате PKCS#10
XML	2	XML-представление сертификата или запроса на сертификат
FORMATTED	4	HTML-представление сертификата или запроса на сертификат
FORMATTEDBASE64	5	HTML-представление сертификата или запроса на сертификат, закодированное в Base64
XMLBASE64	6	XML-представление сертификата или запроса на сертификат, закодированное в Base64

Для получения печатной формы сертификата/запроса на сертификат (HTML-представление) на сервере DSS должны быть настроены печатные формы. В настройках модуля интеграции с Удостоверяющим Центром должны быть заданы параметры:

- `PrintTemplate`
- `ReqPrintTemplate`
- `RevokeRequestPrintTemplate`

В состав Сервиса Подписи DSS входят шаблоны печати по-умолчанию: C:\Program Files\CryptoPro\DSS\SignServer\PrintTemplates\

Пример настройки печатных форм (выполняется Администратором DSS):

```
Set-DssEnrollmentOob -ID 11 -CertificatePrintTemplate "C:\Program Files\CryptoPro\DSS\SignServer\PrintTemplates\Cert.xml" -RequestPrintTemplate "C:\Program Files\CryptoPro\DSS\SignServer\PrintTemplates\Request.xml" -RevokeRequestPrintTemplate "C:\Program Files\CryptoPro\DSS\SignServer\PrintTemplates\RevRequest.xml"
```

Тип запроса к Удостоверяющему Центру (CARequestTypeEnum)

имя	код	значение
Certificate	0	Запрос на выпуск, обновление сертификата.
RevokeRequest	1	Запрос на отзыв, восстановление, приостановление сертификата.

Статусы запроса на сертификат (DSSRequestStatusEnum)

ИМЯ	КОД	ЗНАЧЕНИЕ
PENDING	2	Запрос на сертификат принят на обработку.
ACCEPTED	4	Запрос на сертификат одобрен. Выпущен сертификат по данному запросу.
REJECTED	8	Запрос на сертификат отклонён.
REGISTRATION	16	Запрос на регистрацию принят. Используется только для КриптоПро УЦ 2.0.

Результат расшифрования (DSSDecryptDocumentResponse)

Структура, содержащая ответ сервера, при выполнении операции шифрования. Объект данного типа возвращается при:

- Расшифровании доукмента (метод [/decrypt](#))
- Разборе зашифрованного документа (метод [/decrypt/parse](#))

ПОЛЕ	ТИП	ОПИСАНИЕ
Result	byte[]	Расшифрованный документ
CertificatesIDs	List<int>	Список идентификаторов сертификатов, с помощью которых можно выполнить операцию расшифрования.

Типы шифрования

Набор возможных типов шифрования

имя	код	значение
CMS	0	CMS шифрование (CMS Enveloped data)
XML	1	XML шифрование

Запрос на валидацию ПИН-кода (ValidatePinRequest)

поле	тип	описание
Pin	string	Валидируемое значение ПИН-кода
CheckAttempts	bool	Значение, показывающее, что запрашивается расширенная информация (включая число оставшихся попыток ввода ПИН-кода)

Результат валидации ПИН-кода (ValidatePinResult)

поле	тип	описание
IsValid	bool	Значение показывает, валиден ли переданный ПИН-код
ErrorDescription	string	Описание ошибки в случае завершившейся неудачно валидации
Remaining	int	Количество оставшихся попыток ввода ПИН-кода до блокировки ключа (значение -1 свидетельствует о том, что попытки не ограничены)

Типы ошибок:

поле	описание
unknown_error	Произошла неизвестная ошибка
invalid_pin	Введен неверный ПИН-код
key_blocked	Ключ заблокирован в результате превышения максимального числа попыток ввода ПИН-кода

REST API Сервиса Управления Пользователями (UMS)

Данный раздел содержит руководство разработчика по интеграции с Сервисом Управления Пользователями (UMS) КриптоПро DSS. В разделе приведено подробное описание методов и типов данных REST-интерфейса UMS, сценариев взаимодействия с UMS с примерами HTTP запросов и ответов.

Конечные точки

- [User](#)
- [Users](#)
- [Authtokens](#)
- [Policy](#)
- [GroupPolicy](#)

Типы данных

- Политика доступа к операциям (Access Policy)
- Информация о существующем токене аутентификации MobileAuth (AssignMobileAuthTokenInfo)
- Описание шага схемы аутентификации пользователя (AuthenticationInfo)
- Список доступных методов аутентификации (AuthnMethodDescription)
- Список профилей криптопровайдеров (CryptoProviderInfo)
- Список операций, для которых требуется подтверждение (DSSActions)
- Информация о средстве аутентификации (DssTokenInfo)
- Информация о пользователе (DssUserInfo)
- Коды ошибок (ErrorCodes)
- Запрос на создание внешнего логина пользователя (ExternalLoginInfo)
- Политика группы пользователей (GroupPolicy)
- Типы идентификаторов пользователя (IdentifierType)
- Список групп на Центре Идентификации (IdentityGroupInfo)
- Список доверенных Центров Идентификации (IdentityProviderInfo)
- Тип сообщения, связанного с жизненным циклом апплета (LifecycleMessageType)
- Информация о созданном ключе аутентификации MobileAuth (MobileAuthCreateInfoEx)
- Описание настроек аутентификации MobileAuth (MobileAuthSettings)
- Информация об аутентификации MobileAuth (MobileAuthTokenInfo)
- Информация о созданном ключе аутентификации MobileAuth (MobileAuthUpdateInfoEx)
- Информация об OATH-токене (OATHTokenInfo)
- Политика подтверждения операций (OperationPolicy)
- Ответ на запрос к апплету на SIM-карте (OperationResults)
- Информация о компоненте различительного имени пользователя (RdnInfo)
- Результат отправки запроса на смену ключа аутентификации (SimAuthChangeKeyRequestResult)
- Результат выполнения запроса к апплету на SIM-карте (SimAuthLifecycleMessageStatusRest)
- Информация о SIM-карте пользователя (SimAuthTokenInfo)
- Запрос на получение списка средств аутентификации (TokenRecordsRequest)
- Ответ на запрос о получении отфильтрованного списка средств аутентификации (TokenRecordsResponse)
- Политика Сервиса Управления Пользователями (UmsPolicy)
- Контактная информация пользователя (UserContactInfo)
- Информация о пользователе (UserEmailInfo)
- Информация о созданном ключе аутентификации MobileAuth (UserMobileAuthInfo)
- Информация об OTP-токене пользователя (UserOtpTokenInfo)
- Информация о номере телефона пользователя (UserPhoneInfo)
- Свойства учетной записи пользователя (UserProperty)
- Запрос на аутентификационную информацию пользователя (UserRawAuthDataRequest)
- Запрос на получение списка пользователей (UserRecordsRequest)
- Ответ на запрос о получении отфильтрованного списка пользователей (UserRecordsResponse)

- [Информация об аутентификации при помощи апплета на SIM-карте](#)

Конечные точки

- [User](#)
- [Users](#)
- [Authtokens](#)
- [Policy](#)
- [GroupPolicy](#)

Конечная точка User

Конечная точка User Сервиса Управления Пользователями представлена следующими группами HTTP-методов:

- [Операции с пользователями](#)
- [Настройка различительного имени пользователя](#)
- [Настройка групп пользователей](#)
- [Настройка логина пользователя](#)
- [Настройка пароля пользователя](#)
- [Настройка номера телефона пользователя](#)
- [Настройка адреса электронной почты пользователя](#)
- [Настройка протокола OATH \(одноразовые пароли на брелках\)](#)
- [Общие настройки аутентификации](#)
- [Настройка аутентификации по методу SimAuth](#)
- [Настройка аутентификации по методу MobileAuth](#)
- [Настройка аутентификации myDSS Client](#)
- [Настройка политики подтверждения операций](#)
- [Настройка политики доступа к операциям](#)
- [Блокировка/разблокировка пользователя](#)

Операции с пользователями

Данная группа методов предоставляет способы добавления, удаления и настройки пользователей.

Регистрация пользователя и добавление в группу по умолчанию

RegisterUser

Данный метод позволяет создать нового пользователя без явного указания его группы. При этом пользователь будет помещен в группу, в которой состоит создающий его Оператор. Если Оператор состоит в 2 и более группах, метод возвратит ошибку. В этом случае следует использовать метод "Регистрация пользователя и добавление в определенную группу" (см. ниже).

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user
Параметры	Dictionary < IdentifierType , string > - Словарь идентификаторов пользователя
Возвращаемое значение	id - Глобальный идентификатор пользователя

Пример запроса

```
POST https://<hostname>/STS/ums/user HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 24

{
  "Email": "sample@cp.ru"
}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 38

"7c9f047f-d539-4be9-ae5d-a81ed8495ce1"
```

Регистрация пользователя и добавление в определенную группу

RegisterUser

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/
Параметры	Dictionary <IdentifierType, string> - Словарь идентификаторов пользователя groupName - Имя группы, в которой будет состоять пользователь
Возвращаемое значение	id - Глобальный идентификатор пользователя

Пример запроса

```
POST https://<hostname>/STS/ums/user?groupName=TestGroup HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 24

{
  "Email": "sample@cp.ru"
}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 38

"7c9f047f-d539-4be9-ae5d-a81ed8495ce2"
```

Получение информации о пользователе по глобальному идентификатору

GetUserById

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	DssUserInfo - Учетные данные пользователя

Пример запроса

```
GET https://<hostname>/STS/ums/user/9fba6076-3509-4e8e-abc1-dae6d052a230 HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 415

{
  "UserId": "9fba6076-3509-4e8e-abc1-dae6d052a230",
  "Login": "Test1231231232",
  "PhoneNumber": "78889996655",
  "Email": null,
  "PhoneConfirmed": true,
  "EmailConfirmed": false,
  "DisplayName": null,
  "DistinguishName": "SN=Иванов, G=Иван Иванович, I=ИИИ, CN=Ivanov, C=RU",
  "AccountLocked": false,
  "Group": "CustomGroup",
  "CreationDate": "2019-05-30T16: 25: 35.77",
  "LockoutDate": null,
  "LastLoginDate": "2019-05-30T16: 25: 35.77"
}
```

Получение информации о пользователе по идентификатору

GetUser

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user
Параметры	IdentifierType - Тип идентификатора пользователя value - Значение идентификатора пользователя
Возвращаемое значение	DssUserInfo - Учетные данные пользователя

Пример запроса

```
GET https://<hostname>/STS/ums/user?type=PhoneNumber&value=78889996655 HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 415

{
  "UserId": "9fba6076-3509-4e8e-abc1-dae6d052a230",
  "Login": "Test1231231232",
  "PhoneNumber": "78889996655",
  "Email": null,
  "PhoneConfirmed": true,
  "EmailConfirmed": false,
  "DisplayName": null,
  "DistinguishName": "SN=Иванов, G=Иван Иванович, I=ИИИ, CN=Ivanov, C=RU",
  "AccountLocked": false,
  "Group": "CustomGroup",
  "CreationDate": "2019-05-30T16: 25: 35.77",
  "LockoutDate": null,
  "LastLoginDate": "2019-05-30T16: 25: 35.77"
}
```

Назначение отображаемого имени пользователя в общем списке пользователей

SetUserProperty

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	PATCH, POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}
Параметры	id - Глобальный идентификатор пользователя UserProperty - Свойства учетной записи пользователя
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/9cddeb1f-a36c-4f79-bbda-ec0dece4873c HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 44

{
  "PropertyType": 0,
  "Value": "UserDisplayName"
}
```

Пример ответа

```
HTTP/1.1 200 OK
```

Удаление пользователя

DeleteUser

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	DELETE

ПАРАМЕТР	ЗНАЧЕНИЕ
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	-

Пример запроса

```
DELETE https://<hostname>/STS/ums/user/486f64c8-d5da-4a61-ae0a-5ddb202f82bf HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
```

Настройка различительного имени пользователя

Данная группа методов предоставляет способы работы с различительным именем пользователя.

Назначение различительного имени пользователю

SetUserDistinguishedName

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/dn
Параметры	id - Глобальный идентификатор пользователя components - Словарь компонентов различительного имени формата IDictionary<int, string>
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/9b12bffc-3562-4db6-ada4-d857ad2c291a/dn HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 16

{
  "1": "SampleCN"
}
```

Пример ответа

```
HTTP/1.1 200 OK
```

Получение списка компонентов различительного имени пользователя

GetUserDistinguishedName

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/dn
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	Dictionary<int, string> - Компоненты различительного имени пользователя

Примечание: ключом словаря является идентификатор компонента имени пользователя. Список поддерживаемых ЦИ компонентов имени пользователя можно получить из политики Сервиса Управления Пользователями (ссылка на GetPolicy)

Пример запроса

```
GET https://<hostname>/STS/ums/user/81c71190-ff87-4457-9d6d-86cd66a22f83/dn HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 58

{
  "1": "UmsSampleUser-f3d5fdfe-9da7-436f-8245-908aafa1213e"
}
```

Настройка групп пользователей

Данная группа методов предоставляет способы работы с группами пользователей.

Назначение группы пользователю

SetUserGroup

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/group
Параметры	id - Глобальный идентификатор пользователя groupName - Имя группы newRdns - Новый список компонентов имени пользователя
Возвращаемое значение	-

Пример запроса


```
POST https://<hostname>/STS/ums/user/46778710-1d63-4691-885b-c0e895da3ba8/group?groupName=SampleGroup
HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 58

{
  "1": "UmsSampleUser-33a13f1f-b99d-4e71-8090-01e6aecff75e"
}
```

Пример ответа

HTTP/1.1 200 OK

Получение имени группы пользователя

GetUserGroup

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/group
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	Имя группы пользователя (String)

Пример запроса

GET https://<hostname>/STS/ums/user/40181add-c27f-4491-ab66-d60dc69a583f/group HTTP/1.1

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 9

"Default"
```

Настройка логина пользователя

Данная группа методов позволяет добавлять, удалять и получать сведения о логине учетной записи пользователя.

Установка локального логина учетной записи пользователя

SetLocalLogin

Добавляет локальный логин пользователю или заменяет его, если логин уже существует.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/login/local?newLogin={newLogin}
Параметры	id - Глобальный идентификатор пользователя newLogin - Новый локальный логин пользователя

ПАРАМЕТР	ЗНАЧЕНИЕ
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/4de1b1be-2229-40fe-a0e6-ece733277ccb/login/local?newLogin=TestLogin123
HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
```

Установка внешнего логина учетной записи пользователя

SetExternalLogin

Добавляет внешний логин пользователю или заменяет его, если логин уже существует. Если логин существует и передано пустое значение логина, логин будет удален.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/login
Параметры	id - Глобальный идентификатор пользователя ExternalLoginInfo - Структура для создания внешнего логина
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/4de1b1be-2229-40fe-a0e6-ece733277ccb/login HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 48

{
  "Login": "SampleLogin",
  "IssuerName": "SampleSts"
}
```

Пример ответа

```
HTTP/1.1 200 OK
```

Получение логинов пользователя

GetUserLogin

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/login

ПАРАМЕТР	ЗНАЧЕНИЕ
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	List < ExternalLoginInfo > - Список логинов пользователя

Пример запроса

```
GET https://<hostname>/STS/ums/user/80c275a1-f375-4bf1-aac3-9c478aee950c/login HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 50

[{"Login": "UmsSampleUser", "IssuerName": "realsts"}]
```

Удаление логина пользователя

DeleteUserLogin

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	DELETE
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/login
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	-

Пример запроса

```
DELETE https://<hostname>/STS/ums/user/9fba6076-3509-4e8e-abc1-dae6d052a230/login HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
```

Настройка пароля пользователя

Данная группа методов позволяет управлять паролями пользователей.

Сброс пароля пользователя

ResetPassword

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST

ПАРАМЕТР	ЗНАЧЕНИЕ
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/password
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	Новый пароль пользователя (String)

Примечание

При использовании данного метода требуется, чтобы у Пользователя в контактной информации был задан [Email](#) и/или [номер телефона для OTP-паролей](#).

Пример запроса

```
POST https://<hostname>/STS/ums/user/ef2ab692-f457-432a-bbe8-392ec44b6ee1/password HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 2

{}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 56

"стихийный аппарат доказывает"
```

Настройка номера телефона пользователя

Данная группа методов позволяет управлять номерами телефонов пользователей.

Получение списка телефонов пользователя

GetUserPhones

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/phones
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	List<UserContactInfo> - Контактная информация пользователя

Пример запроса

```
GET https://<hostname>/STS/ums/user/e5ec3a9e-76a8-4f99-8750-2180b47ef2e1/phones HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 115

[{
  "Type": "PhoneNumber",
  "Contact": "79991234568",
  "Confirmed": false,
  "Primary": false,
  "Notification": false,
  "Usages": []
}]
```

Добавление телефона пользователя

AddUserPhone

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/phones
Параметры	id - Глобальный идентификатор пользователя phone - Номер телефона пользователя
Возвращаемое значение	UserContactInfo - Контактная информация пользователя

Пример запроса

```
POST https://<hostname>/STS/ums/user/136d0be8-aebf-4322-8840-4f441f576a91/phones HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 18

"+7(999)123-45-60"
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 110

{
  "Type": "PhoneNumber",
  "Contact": "79991234560",
  "Confirmed": true,
  "Primary": true,
  "Notification": true,
  "Usages": []
}
```

Подтверждение номера телефона пользователя

Метод используется, если не требуется отправка одноразового пароля.

ConfirmUserPhone

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST, PATCH
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/phones/{phone}/confirm
Параметры	id - Глобальный идентификатор пользователя phone - Номер телефона пользователя
Возвращаемое значение	UserContactInfo - Контактная информация пользователя

Пример запроса

```
POST https://<hostname>/STS/ums/user/e5ec3a9e-76a8-4f99-8750-2180b47ef2e1/phones/79991234568/confirm HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 2

{}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 110

{
  "Type": "PhoneNumber",
  "Contact": "79991234568",
  "Confirmed": true,
  "Primary": true,
  "Notification": true,
  "Usages": []
}
```

Запрос отправки одноразового пароля для подтверждения номера телефона

RequireUserPhoneConfirmation

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST, PATCH
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/phones/{phone}/requireconfirm
Параметры	id - Глобальный идентификатор пользователя phone - Номер телефона пользователя
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/3d540f99-8451-4ad3-b8aa-ea8b446eb6f9/phones/71238889900/requireconfirm HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 2

{}
```

Пример ответа

HTTP/1.1 200 OK

Подтверждение номера телефона при помощи кода подтверждения (одноразового пароля)

SubmitUserPhoneConfirmation

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST, PATCH
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/phones/{phone}/submitconfirm
Параметры	id - Глобальный идентификатор пользователя phone - Номер телефона пользователя confirmationCode - Код подтверждения
Возвращаемое значение	UserContactInfo - Контактная информация пользователя

Пример запроса

POST https://<hostname>/STS/ums/user/3d540f99-8451-4ad3-b8aa-ea8b446eb6f9/phones/71238889900/submitconfirm
HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 7

"91830"

Пример ответа

HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 110

{
 "Type": "PhoneNumber",
 "Contact": "71238889900",
 "Confirmed": true,
 "Primary": true,
 "Notification": true,
 "Usages": []
}

Установка номера телефона пользователя для идентификации (входа)

SetUserPrimaryPhone

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST, PATCH
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/phones/{phone}/primary
Параметры	id - Глобальный идентификатор пользователя phone - Номер телефона пользователя primary - флаг, отвечающий за назначение телефона в качестве логина для идентификации (если это не запрещено на сервере)

ПАРАМЕТР	ЗНАЧЕНИЕ
Возвращаемое значение	UserContactInfo - Контактная информация пользователя

Пример запроса

```
POST https://<hostname>/STS/ums/user/3538fb42-a6c1-442c-9a27-70de9f8aeed7/phones/79991234569/primary HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 5

false
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 111

{
  "Type": "PhoneNumber",
  "Contact": "79991234569",
  "Confirmed": true,
  "Primary": false,
  "Notification": true,
  "Usages": []
}
```

Установка номера телефона для получения уведомлений

SetUserNotificationPhone

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST, PATCH
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/phones/{phone}/notification
Параметры	id - Глобальный идентификатор пользователя phone - Номер телефона пользователя notification - флаг, отвечающий за возможность получать уведомления на заданный номер
Возвращаемое значение	UserContactInfo - Контактная информация пользователя

Пример запроса

```
POST https://<hostname>/STS/ums/user/3538fb42-a6c1-442c-9a27-70de9f8aeed7/phones/79991234569/notification HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 5

false
```

Пример ответа


```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 112

{
  "Type": "PhoneNumber",
  "Contact": "79991234569",
  "Confirmed": true,
  "Primary": false,
  "Notification": false,
  "Usages": []
}
```

Назначение номера телефона для вторичной аутентификации и/или подтверждения операций

SetUserSecondaryAuthPhone

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST, PATCH
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/phones/{phone}/secondaryauth
Параметры	id - Глобальный идентификатор пользователя phone - Номер телефона пользователя
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/3d540f99-8451-4ad3-b8aa-ea8b446eb6f9/phones/71238889900/secondaryauth
HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 2

{}
```

Пример ответа

```
HTTP/1.1 200 OK
```

Удаление номера телефона пользователя

DeleteUserPhone

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	DELETE
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/phones/{phone}/
Параметры	id - Глобальный идентификатор пользователя phone - Номер телефона пользователя
Возвращаемое значение	-

Пример запроса

DELETE https://<hostname>/STS/ums/user/136d0be8-aebf-4322-8840-4f441f576a91/phones/79991234560/ HTTP/1.1

Пример ответа

HTTP/1.1 200 OK

(Для обратной совместимости) Назначение номера телефона для вторичной аутентификации и/или подтверждения операций

GetUserPhoneTokenInfo

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/phonenumbers
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	UserPhoneInfo - Информация о номере телефона пользователя

Пример запроса

GET https://<hostname>/STS/ums/user/530c285f-047f-40d1-8daa-f7f882347ee0/phonenumbers HTTP/1.1

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 46

{
  "PhoneNumber": "79150000000",
  "Confirmed": true
}
```

(Для обратной совместимости) Установка номера телефона как основного (для входа, вторичной аутентификации, оповещения)

SetUserMainPhone

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/phonenumbers
Параметры	id - Глобальный идентификатор пользователя phone - Номер телефона пользователя
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/a3a0a399-a918-4e8c-9008-2f652815de5f/phonenumbers HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 13

"79150000000"
```

Пример ответа

```
HTTP/1.1 200 OK
```

Удаление номера телефона пользователя

DeleteUserPhone

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	DELETE
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/phonenumbers
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	-

Пример запроса

```
DELETE https://<hostname>/STS/ums/user/84e67841-ceaa-4f18-a706-eab0d2a997a1/phonenumbers HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
```

Настройка адреса электронной почты пользователя

Данная группа методов позволяет управлять адресами электронной почты пользователей.

Получение списка адресов электронной почты пользователей

GetUserEmails

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/emails
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	UserContactInfo - Контактная информация пользователя

Пример запроса

```
GET https://<hostname>/STS/ums/user/b6e54b50-b047-46d5-9915-3d2a72fc5a45/emails HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 320

[{
  "Type": "EmailAddress",
  "Contact": "asdad@asd.rrurururu",
  "Confirmed": true,
  "Primary": true,
  "Notification": true,
  "Usages": [{
    "Type": "ОТР",
    "Title": "Одноразовые пароли",
    "Description": "Адресат получения одноразовых паролей для подтверждения операций"
  }]
}]
```

Добавление адреса электронной почты пользователя

AddUserEmail

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/emails
Параметры	id - Глобальный идентификатор пользователя email - Адрес электронной почты пользователя
Возвращаемое значение	UserContactInfo - Контактная информация пользователя

Пример запроса

```
POST https://<hostname>/STS/ums/user/d0d6f46e-0361-48d0-af76-5a6206d93b75/emails HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 33

"umscontactinfotestuser@test1.ru"
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 131

{
  "Type": "EmailAddress",
  "Contact": "umscontactinfotestuser@test1.ru",
  "Confirmed": true,
  "Primary": true,
  "Notification": true,
  "Usages": []
}
```

Подтверждение адреса электронной почты пользователя

Метод используется, если не требуется отправка одноразового пароля. ConfirmUserEmail

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST, PATCH
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/emails/{email}/confirm
Параметры	id - Глобальный идентификатор пользователя email - Адрес электронной почты пользователя
Возвращаемое значение	UserContactInfo - Контактная информация пользователя

Пример запроса

```
POST https://<hostname>/STS/ums/user/d5daff09-15bc-4519-928f-67aea39bae20/emails/umscontactinfotestuser@test2.ru/confirm HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 2

{}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 131

{
  "Type": "EmailAddress",
  "Contact": "umscontactinfotestuser@test2.ru",
  "Confirmed": true,
  "Primary": true,
  "Notification": true,
  "Usages": []
}
```

Запрос отправки одноразового пароля для подтверждения адреса электронной почты пользователя

RequireUserEmailConfirmation

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST, PATCH
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/emails/{email}/requireconfirm
Параметры	id - Глобальный идентификатор пользователя email - Адрес электронной почты пользователя
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/b6e54b50-b047-46d5-9915-3d2a72fc5a45/emails/asdad@asd.rrurururu/requireconfirm HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 2

{}
```

Пример ответа

```
HTTP/1.1 200 OK
```

Подтверждение адреса электронной почты при помощи кода подтверждения (одноразового пароля)

SubmitUserEmailConfirmation

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST, PATCH
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/emails/{email}/submitconfirm
Параметры	id - Глобальный идентификатор пользователя email - Адрес электронной почты пользователя confirmationCode - Код подтверждения
Возвращаемое значение	UserContactInfo - Контактная информация пользователя

Пример запроса

```
POST https://<hostname>/STS/ums/user/b6e54b50-b047-46d5-9915-3d2a72fc5a45/emails/asdad@asd.rrurururu/submitconfirm HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 7

"41069"
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 119

{
  "Type": "EmailAddress",
  "Contact": "asdad@asd.rrurururu",
  "Confirmed": true,
  "Primary": true,
  "Notification": true,
  "Usages": []
}
```

Установка адреса электронной почты пользователя для идентификации (входа)

SetUserPrimaryEmail

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST, PATCH

ПАРАМЕТР	ЗНАЧЕНИЕ
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/emails/{email}/primary
Параметры	id - Глобальный идентификатор пользователя email - Адрес электронной почты пользователя primary - флаг, отвечающий за назначение почтового адреса в качестве логина для идентификации (если это не запрещено на сервере)
Возвращаемое значение	UserContactInfo - Контактная информация пользователя

Пример запроса

```
POST https://<hostname>/STS/ums/user/1c4df002-ed3c-4c88-a395-8e2795a295c6/emails/umscontactinfotestuser@test3.ru/primary HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 5

false
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 132

{
  "Type": "EmailAddress",
  "Contact": "umscontactinfotestuser@test3.ru",
  "Confirmed": true,
  "Primary": false,
  "Notification": true,
  "Usages": []
}
```

Установка адреса электронной почты для получения уведомлений

SetUserNotificationEmail

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST, PATCH
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/emails/{email}/notification
Параметры	id - Глобальный идентификатор пользователя email - Адрес электронной почты пользователя notification - флаг, отвечающий за возможность получать уведомления на заданный почтовый адрес
Возвращаемое значение	UserContactInfo - Контактная информация пользователя

Пример запроса

```
POST https://<hostname>/STS/ums/user/1c4df002-ed3c-4c88-a395-
8e2795a295c6/emails/umscontactinfotestuser@test3.ru/notification HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 5

false
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 133

{
  "Type": "EmailAddress",
  "Contact": "umscontactinfotestuser@test3.ru",
  "Confirmed": true,
  "Primary": false,
  "Notification": false,
  "Usages": []
}
```

Установка адреса электронной почты для вторичной аутентификации

SetUserSecondaryAuthEmail

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST, PATCH
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/emails/{email}/secondaryauth
Параметры	id - Глобальный идентификатор пользователя email - Адрес электронной почты пользователя
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/b6e54b50-b047-46d5-9915-
3d2a72fc5a45/emails/asdad@asd.rurururu/secondaryauth HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 2

{}
```

Пример ответа

```
HTTP/1.1 200 OK
```

Удаление адреса электронной почты пользователя

DeleteUserEmail

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	DELETE

ПАРАМЕТР	ЗНАЧЕНИЕ
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/emails/{email}/
Параметры	id - Глобальный идентификатор пользователя email - Адрес электронной почты пользователя
Возвращаемое значение	-

Пример запроса

```
DELETE https://<hostname>/STS/ums/user/d2ebbe7d-55b1-468e-9e30-9c101377572f/emails/testuser4@dss.ru/ HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
```

(Для обратной совместимости) Назначение адреса электронной почты для вторичной аутентификации и/или подтверждения операций

GetUserEmailTokenInfo

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/email
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	UserEmailInfo - Информация об адресе электронной почты пользователя

Пример запроса

```
GET https://<hostname>/STS/ums/user/0fd10693-f4a8-46b2-bb9b-aa9e0ae9760d/email HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 32

{
  "Email": null,
  "Confirmed": false
}
```

(Для обратной совместимости) Установка адреса электронной почты пользователя в качестве основного (для входа, вторичной аутентификации и оповещения)

SetUserMainEmail

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST

ПАРАМЕТР	ЗНАЧЕНИЕ
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/email
Параметры	id - Глобальный идентификатор пользователя email - Адрес электронной почты пользователя
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/0929c5b2-1ef3-4b23-b544-99a9658fe421/email HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 14

"sample@cp.ru"
```

Пример ответа

```
HTTP/1.1 200 OK
```

Удаление адреса электронной почты пользователя

DeleteUserEmail

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	DELETE
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/email
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	-

Пример запроса

```
DELETE https://<hostname>/STS/ums/user/7d68d4f6-c398-4ce1-a217-8a097da42294/email HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
```

Настройка протокола OATH (одноразовые пароли на брелках)

Назначение пользователю токена аутентификации HOTP или TOTP

SetUserOtpToken

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/oath

ПАРАМЕТР	ЗНАЧЕНИЕ
Параметры	id - Глобальный идентификатор пользователя OathTokenInfo - информация о токене, назначаемом пользователю
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/6e9eeec-6a73-4b06-9274-4ba2e7216ed4/oath HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 62

{
  "Serial": "AA000001",
  "FirstOtp": "189034",
  "SecondOtp": "223303"
}
```

Пример ответа

```
HTTP/1.1 200 OK
```

Получение информации о прикрепленном к пользователю токене

GetUserOtpTokenInfo

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/oath
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	UserOtpTokenInfo - Информация об OTP-токене пользователя

Пример запроса

```
GET https://<hostname>/STS/ums/user/6e9eeec-6a73-4b06-9274-4ba2e7216ed4/oath HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 35

{
  "Serial": "AA000001",
  "Type": "НОТР"
}
```

Удаление закрепленного за пользователем токена

RemoveUserOtpToken

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	DELETE
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/oath
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	-

Пример запроса

```
DELETE https://<hostname>/STS/ums/user/612a7f3f-49a1-48ed-ac27-d47062f68b1a/oath HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
```

Общие настройки аутентификации

Получение схемы аутентификации пользователя

GetUserAuthenticationScheme

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/authmethod
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	List < AuthenticationInfo > - Список с информацией о шагах схемы аутентификации пользователя

Пример запроса

```
GET https://<hostname>/STS/ums/user/b6e54b50-b047-46d5-9915-3d2a72fc5a45/authmethod HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 194

[
  {
    "MethodUri": "http://schemas.microsoft.com/ws/2012/09/identity/authenticationmethod/none",
    "Level": 0
  },
  {
    "MethodUri": "http://dss.cryptopro.ru/identity/authenticationmethod/otpviaemail",
    "Level": 1
  }
]
```

Назначение метода аутентификации пользователя Identification Only

SetUserAuthenticationMethodIdOnly

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/authmethod/identity
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/e6c3240b-e9f8-44c1-b662-0e0937883eaa/authmethod/identity HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 2

{}
```

Пример ответа

```
HTTP/1.1 200 OK
```

Удаление метода аутентификации пользователя Identification Only

DeleteUserAuthenticationMethodIdentity

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	DELETE
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/authmethod/identity
Параметры	
Возвращаемое значение	-

Пример запроса

```
DELETE https://<hostname>/STS/ums/user/e33d3b69-4461-4789-a9cc-ce07f8bfca07/authmethod/identity HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
```

Назначение метода аутентификации пользователя по паролю

SetUserAuthenticationMethodPassword

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/authmethod/password
Параметры	id - Глобальный идентификатор пользователя

ПАРАМЕТР	ЗНАЧЕНИЕ
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/958186d2-5514-4ab3-8f22-ae9c34417675/authmethod/password HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 2

{ }
```

Пример ответа

```
HTTP/1.1 200 OK
```

Удаление метода аутентификации пользователя по паролю

DeleteUserAuthenticationMethodPasword

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	DELETE
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/authmethod/password
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	-

Пример запроса

```
DELETE https://<hostname>/STS/ums/user/87f0e53f-04d5-4462-8e8f-06a58e48e008/authmethod/password HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
```

Назначение метода аутентификации пользователя по сертификату

SetUserAuthenticationMethodCert

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/authmethod/cert
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/dc723f91-2728-426c-a18b-fe5c5d2499b9/authmethod/cert HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 2

{}
```

Пример ответа

```
HTTP/1.1 200 OK
```

Удаление метода аутентификации пользователя по сертификату

DeleteUserAuthenticationMethodCert

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	DELETE
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/authmethod/cert
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	-

Пример запроса

```
DELETE https://<hostname>/STS/ums/user/87f0e53f-04d5-4462-8e8f-06a58e48e008/authmethod/cert HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
```

Назначение метода аутентификации пользователя через внешний ЦИ

SetUserAuthenticationMethodSaml

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/authmethod/external
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/399397a2-7d86-42fa-a95a-5a943db1aea2/authmethod/external HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 2

{}
```

Пример ответа

HTTP/1.1 200 OK

Удаление метода аутентификации пользователя через внешний ЦИ

DeleteUserAuthenticationMethodSaml

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	DELETE
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/authmethod/external
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	-

Пример запроса

DELETE https://<hostname>/STS/ums/user/87f0e53f-04d5-4462-8e8f-06a58e48e008/authmethod/external HTTP/1.1

Пример ответа

HTTP/1.1 200 OK

Назначение метода аутентификации пользователя через одноразовые пароли по SMS

SetUserAuthenticationMethodOtpViaSms

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/authmethod/otpviasms
Параметры	id - Глобальный идентификатор пользователя level - Шаг аутентификации (int)
Возвращаемое значение	-

Пример запроса

POST https://<hostname>/STS/ums/user/7184647a-fc20-4a23-a77f-9a1c6172c82b/authmethod/otpviasms?level=1
HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 2

{}

Пример ответа

HTTP/1.1 200 OK

Удаление метода аутентификации пользователя через одноразовые пароли по SMS

DeleteUserAuthenticationMethodOtpViaSms

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	DELETE
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/authmethod/otpviasms
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	-

Пример запроса

```
DELETE https://<hostname>/STS/ums/user/87f0e53f-04d5-4462-8e8f-06a58e48e008/authmethod/otpviasms HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
```

Назначение метода аутентификации пользователя через одноразовые пароли по электронной почте
SetUserAuthenticationMethodOtpViaEmail

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/authmethod/otpviaemail
Параметры	id - Глобальный идентификатор пользователя level - Шаг аутентификации (int)
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/b6e54b50-b047-46d5-9915-3d2a72fc5a45/authmethod/otpviaemail?level=1
HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 2

{ }
```

Пример ответа

```
HTTP/1.1 200 OK
```

Удаление метода аутентификации пользователя через одноразовые пароли по электронной почте
DeleteUserAuthenticationMethodOtpViaEmail

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	DELETE
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/authmethod/otpviaemail

ПАРАМЕТР	ЗНАЧЕНИЕ
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	-

Пример запроса

```
DELETE https://<hostname>/STS/ums/user/7f0f902e-dc9b-462e-a3c3-c4d6b21ba013/authmethod/otpviaemail HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
```

Назначение метода аутентификации пользователя через OTP-токен

SetUserAuthenticationMethodOATH

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/authmethod/oath
Параметры	id - Глобальный идентификатор пользователя level - Шаг аутентификации (int)
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/6e9eeec-6a73-4b06-9274-4ba2e7216ed4/authmethod/oath?level=1 HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 2

{}
```

Пример ответа

```
HTTP/1.1 200 OK
```

Удаление метода аутентификации пользователя через OTP-токен

DeleteUserAuthenticationMethodOATH

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	DELETE
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/authmethod/oath
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	-

Пример запроса

```
DELETE https://<hostname>/STS/ums/user/87f0e53f-04d5-4462-8e8f-06a58e48e008/authmethod/oath HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
```

Назначение метода аутентификации пользователя при помощи апплета на SIM-карте (SimAuth)

SetUserAuthenticationMethodSimAuth

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/authmethod/simauth
Параметры	id - Глобальный идентификатор пользователя level - Шаг аутентификации (int)
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/58149f21-fbad-4606-a2d3-e80141e85a11/authmethod/simauth?level=1 HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 2

{}
```

Пример ответа

```
HTTP/1.1 200 OK
```

Удаление метода аутентификации пользователя при помощи апплета на SIM-карте (SimAuth)

DeleteUserAuthenticationMethodSimAuth

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	DELETE
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/authmethod/simauth
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	-

Пример запроса

```
DELETE https://<hostname>/STS/ums/user/4740e82c-0054-401c-aceb-c13ba6a0a4cd/authmethod/simauth HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
```

Назначение метода аутентификации пользователя при помощи мобильного приложения myDSS (MobileAuth)

SetUserAuthenticationMethodMobileAuth

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/authmethod/mobileauth
Параметры	id - Глобальный идентификатор пользователя level - Шаг аутентификации (int)
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/58149f21-fbad-4606-a2d3-e80141e85a11/authmethod/mobileauth?level=1
HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 2

{}
```

Пример ответа

```
HTTP/1.1 200 OK
```

Удаление метода аутентификации пользователя при помощи мобильного приложения myDSS (MobileAuth)

DeleteUserAuthenticationMethodMobileAuth

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	DELETE
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/authmethod/mobileauth
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	-

Пример запроса

```
DELETE https://<hostname>/STS/ums/user/87f0e53f-04d5-4462-8e8f-06a58e48e008/authmethod/mobileauth HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
```

Получение подробной аутентификационной информации пользователя

GetUserRawAuthenticationDataAsync

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/rawauthdata
Параметры	UserRawAuthDataRequest - Запрос на аутентификационную информацию пользователя
Возвращаемое значение	Сведения о запрошенных методах аутентификации пользователя (String)

Пример запроса

```
POST https://<hostname>/STS/ums/user/rawauthdata HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 208

{
  "UserId": "9fba6076-3509-4e8e-abc1-dae6d052a230",
  "AuthnMethods": ["http: //dss.cryptopro.ru/identity/authenticationmethod/password", "http:
//dss.cryptopro.ru/identity/authenticationmethod/otpviasms"],
  "Format": 0
}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 460

{"\"UserId\": \"9fba6076-3509-4e8e-abc1-dae6d052a230\", \"AuthMethodInfos\": [{\"AuthnMethod\": \"http:
//dss.cryptopro.ru/identity/authenticationmethod/password\", \"Confirmed\": true, \"Parameters\": [{\"Key\":
\"HasPassword\", \"Value\": \"True\"}]}, {\"AuthnMethod\": \"http:
//dss.cryptopro.ru/identity/authenticationmethod/otpviasms\", \"Confirmed\": true, \"Parameters\": [{\"Key\":
\"PhoneNumber\", \"Value\": \"78889996655\"}, {\"Key\": \"Confirmed\", \"Value\": \"true\"}]}]}}
```

Настройка аутентификации при помощи апплета на SIM-карте (SimAuth)

Назначение пользователю токена для аутентификации SimAuth

SetUserSimAuthToken

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/simauth
Параметры	id - Глобальный идентификатор пользователя SimAuthTokenInfo - Информация о SIM-карте
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/58149f21-fbad-4606-a2d3-e80141e85a11/simauth HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 81

{
  "IccId": "896010265977857677",
  "ProfileId": "3e906aff-8639-463e-af59-07a44a96dd21",
  "PhoneNumber": "79150000000"
}
```

Пример ответа

HTTP/1.1 200 OK

Получение сведений о назначенном пользователю токене аутентификации SimAuth

GetUserSimAuthTokenInfo

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/simauth
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	UserSimAuthInfo - Информация о SIM-карте пользователя

Пример запроса

GET https://<hostname>/STS/ums/user/58149f21-fbad-4606-a2d3-e80141e85a11/simauth HTTP/1.1

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 151

{
  "IccId": "896010265977857677",
  "ProfileId": "3e906aff-8639-463e-af59-07a44a96dd21",
  "PhoneNumber": "79150000000",
  "ActivationCode": null,
  "LastKnownStatus": 0
}
```

Удаление у пользователя токена для аутентификации SimAuth

RemoveUserSimAuthToken

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	DELETE
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/simauth
Параметры	id - Глобальный идентификатор пользователя

ПАРАМЕТР	ЗНАЧЕНИЕ
Возвращаемое значение	-

Пример запроса

DELETE https://<hostname>/STS/ums/user/4740e82c-0054-401c-aceb-c13ba6a0a4cd/authmethod/simauth HTTP/1.1

Пример ответа

HTTP/1.1 200 OK

Отправка запроса к апплету на SIM-карте

SendSimAuthTokenMessage

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/simauth/message/{messageType}
Параметры	id - Глобальный идентификатор пользователя LifecycleMessageType - Тип сообщения, связанного с жизненным циклом апплета
Возвращаемое значение	Идентификатор запроса к SIM-карте (String)

Пример запроса

POST https://<hostname>/STS/ums/user/ed124c3b-e518-4a3e-9100-c492ae0b2f0c/simauth/message/GetStatus HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 2

{ }

Пример ответа

HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 9

"7975336"

Отправка запроса к апплету на обновление ключа аутентификации SIM-карты

SendSimAuthTokenChangeKeyMessage

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/simauth/changekeymessage
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	SimAuthChangeKeyRequestResult - Результат отправки запроса на смену ключа аутентификации

```
POST user/{id}/simauth/changekeymessage
```

Параметры

- `id` - идентификатор пользователя

Ответ сервиса будет содержать:

- `TransactionId` - идентификатор транзакции смены ключа аутентификации
- `ActivationCode2` - код смены ключа.

Код смены ключа аутентификации представляет собой список из 10 блоков цифр, которые пользователю необходимо ввести на мобильном устройстве.

Результат смены ключа можно узнать по `TransactionId`, вызвав [метод получения результата транзакции](#)

Внимание!

Получение результата выполнения запроса на апплете возможно либо через CallBack-сервис, либо периодическим опросом [метода получения результата транзакции](#)

Пример запроса

```
POST https://<hostname>/STS/ums/user/e74dabec-6c91-4329-970f-361293c5f7a6/simauth/changekeymessage HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 2

{}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 162

{
  "TransactionId": "16211709",
  "ActivationCode2":
    ["4899304416", "4856874867", "3298082352", "4329367605", "3563813072", "1989944545", "306179350", "2431789802", "14006880"]
}
```

Получение результата выполнения запроса к апплету на SIM-карте

GetSimAuthTokenMessage

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/simauth/message/{transactionId}
Параметры	id - Глобальный идентификатор пользователя transactionId -
Возвращаемое значение	SimAuthLifecycleMessageStatusRest - Статус операции

Пример запроса


```
GET https://<hostname>/STS/ums/user/ed124c3b-e518-4a3e-9100-c492ae0b2f0c/simauth/message/7975336 HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 64

{
  "AppletResult": 0,
  "IsCompleted": false,
  "MessageType": "GetStatus"
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 64

{
  "AppletResult": 0,
  "IsCompleted": false,
  "MessageType": "GetStatus"
}
```

Настройка аутентификации при помощи мобильного приложения myDSS (MobileAuth)

Генерация и назначение пользователю токена аутентификации MobileAuth

CreateUserMobileAuthToken

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/mobileauth
Параметры	id - Глобальный идентификатор пользователя MobileAuthTokenInfo - Информация о токене аутентификации MobileAuth
Возвращаемое значение	MobileAuthCreateInfoEx - Информация о созданном ключе аутентификации MobileAuth

Пример запроса

```
POST https://<hostname>/STS/ums/user/9fba6076-3509-4e8e-abc1-dae6d052a230/mobileauth HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 118

{
  "UserContactInfo": "78889996655",
  "UserContactInfoType": "PhoneNumber",
  "NeedXmlKeyInfo": false,
  "DelayedActivation": false
}
```

```
POST https://<hostname>/STS/ums/user/9fba6076-3509-4e8e-abc1-dae6d052a230/mobileauth HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 118

{
  "UserContactInfo": "78889996655",
  "UserContactInfoType": "PhoneNumber",
  "NeedXmlKeyInfo": false,
  "DelayedActivation": false
}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 7687

{
  "XmlKeyInfo": "",
  "ExternalUserId": "6222662c-d0c5-4077-8942-be328999fdbd",
  "QRCode":
    "R0lGODlhLAESAfcAAAAAAMwAAZgAAmQAAZAAA/wArAAARmWArZgArmQArzAAR/wBVAABVMwBVZgBVMQBvZABV/wCAAACAMwCAZgCAmQCAzACA/wCqAACqMwCqZgCqmQCqzACq/wDVAADVWmDVZgDVmQDVzADV/wD/AAD/MwD/ZgD/mQD/zAD//zMAADMAMzMAZjMAmTMAzDMA/zMrADMrMzMrZjMrMTmrZDMr/zNVADNVmzNVZjNVmTNVzDNV/zOAAADOAMzOAZjOAmTOAzDOA/zOqADOqMzOqZjOqmTOqzDOq/zPVADPVMzPVZjPVmTPVzDPV/zP/ADP/MzP/ZjP/mTP/zDP//2YAAGYAM2YAZmYAmWYAZGYA/2YrAGYrM2YrZmYrmWYrzGYr/2ZVAGZVM2ZVZmZVmWZVzGZV/2aAAGaAM2aAZmAlhAAGaA/2aAGaAM2aAZmAlhAGaA/2bYAGbYAM2bYzmYbYzGYb/2bYAGbYAM2bYzmYbYzGYb/2cYAGcYAM2cYzmYcYzGYc/2cYAGcYAM2cYzmYcYzGYc/2dYAGdYAM2dYzmYdYzGYd/2dYAGdYAM2dYzmYdYzGYd/2eYAGeYAM2eYzmYeYzGYe/2eYAGeYAM2eYzmYeYzGYe/2fYAGfYAM2fYzmYfYzGYf/2fYAGfYAM2fYzmYfYzGYf/2gYAGgYAM2gYzmYgYzGYg/2gYAGgYAM2gYzmYgYzGYg/2hYAGhYAM2hYzmYhYzGYh/2hYAGhYAM2hYzmYhYzGYh/2iYAGiYAM2iYzmYiYzGYi/2iYAGiYAM2iYzmYiYzGYi/2jYAGjYAM2jYzmYjYzGYj/2jYAGjYAM2jYzmYjYzGYj/2kYAGkYAM2kYzmYkYzGYk/2kYAGkYAM2kYzmYkYzGYk/2lYAGlYAM2lYzmYlYzGYl/2lYAGlYAM2lYzmYlYzGYl/2mYAGmYAM2mYzmYmYzGYm/2mYAGmYAM2mYzmYmYzGYm/2nYAGnYAM2nYzmYnYzGYn/2nYAGnYAM2nYzmYnYzGYn/2oYAGoYAM2oYzmYoYzGYo/2oYAGoYAM2oYzmYoYzGYo/2pYAGpYAM2pYzmYpYzGYp/2pYAGpYAM2pYzmYpYzGYp/2qYAGqYAM2qYzmYqYzGYq/2qYAGqYAM2qYzmYqYzGYq/2rYAGrYAM2rYzmYrYzGYr/2rYAGrYAM2rYzmYrYzGYr/2sYAGsYAM2sYzmYsYzGYs/2sYAGsYAM2sYzmYsYzGYs/2tYAGtYAM2tYzmYtYzGYt/2tYAGtYAM2tYzmYtYzGYt/2uYAGuYAM2uYzmYuYzGYu/2uYAGuYAM2uYzmYuYzGYu/2vYAGvYAM2vYzmYvYzGYv/2vYAGvYAM2vYzmYvYzGYv/2wYAGwYAM2wYzmYwYzGYw/2wYAGwYAM2wYzmYwYzGYw/2xYAGxYAM2xYzmYxYzGYx/2xYAGxYAM2xYzmYxYzGYx/2yYAGyYAM2yYzmYyYzGYy/2yYAGyYAM2yYzmYyYzGYy/2zYAGzYAM2zYzmYzYzGYz/2zYAGzYAM2zYzmYzYzGYz/2AAAGAAAM2AAZmAAZGAA/2AAAGAAAM2AAZmAAZGAA/2BAGBAM2BAGZmBAGZGAB/2BAGBAM2BAGZmBAGZGAB/2CAGCAM2CAGZmCAGZGAC/2CAGCAM2CAGZmCAGZGAC/2DAGDAM2DAGZmDAGZGAD/2DAGDAM2DAGZmDAGZGAD/2EAGEAM2EAGZmEAGZGAE/2EAGEAM2EAGZmEAGZGAE/2FAGFAM2FAGZmFAGZGAF/2FAGFAM2FAGZmFAGZGAF/2GAGGAM2GAGZmGAGZGAG/2GAGGAM2GAGZmGAGZGAG/2HAGHAM2HAGZmHAGZGAH/2HAGHAM2HAGZmHAGZGAH/2IAGIAM2IAGZmIAGZGAI/2IAGIAM2IAGZmIAGZGAI/2JAGJAM2JAGZmJAGZGAJ/2JAGJAM2JAGZmJAGZGAJ/2KAGKAM2KAGZmKAGZGAJ/2KAGKAM2KAGZmKAGZGAJ/2LAGLAM2LAGZmLAGZGAL/2LAGLAM2LAGZmLAGZGAL/2MAGMAM2MAGZmMAGZGAM/2MAGMAM2MAGZmMAGZGAM/2NAGNAM2NAGZmNAGZGAN/2NAGNAM2NAGZmNAGZGAN/2OAGOAM2OGOZmOGOZGAO/2OAGOAM2OGOZmOGOZGAO/2PAGPAM2PAGZmPAGZGAO/2PAGPAM2PAGZmPAGZGAO/2QAGQAM2QAGZmQAGZGAO/2QAGQAM2QAGZmQAGZGAO/2RAGRAM2RAGZmRAGZGAO/2RAGRAM2RAGZmRAGZGAO/2SAGSAM2SAGZmSAGZGAO/2SAGSAM2SAGZmSAGZGAO/2TAGTAM2TAGZmTAGZGAO/2TAGTAM2TAGZmTAGZGAO/2UAGUAM2UAGZmUAGZGAO/2UAGUAM2UAGZmUAGZGAO/2VAGVAM2VAGZmVAGZGAO/2VAGVAM2VAGZmVAGZGAO/2WAGWAM2WAGZmWAGZGAO/2WAGWAM2WAGZmWAGZGAO/2XAGXAM2XAGZmXAGZGAO/2XAGXAM2XAGZmXAGZGAO/2YAGYAM2YAGZmYAGZGAO/2YAGYAM2YAGZmYAGZGAO/2ZAGZAM2ZAGZmZAGZGAO/2ZAGZAM2ZAGZmZAGZGAO/2aAGaAM2aAGZmaAGZGAO/2aAGaAM2aAGZmaAGZGAO/2bAGbAM2bAGZmbAGZGAO/2bAGbAM2bAGZmbAGZGAO/2cAGcAM2cAGZmcAGZGAO/2cAGcAM2cAGZmcAGZGAO/2dAGdAM2dAGZmdAGZGAO/2dAGdAM2dAGZmdAGZGAO/2eAGeAM2eAGZmeAGZGAO/2eAGeAM2eAGZmeAGZGAO/2fAGfAM2fAGZmfAGZGAO/2fAGfAM2fAGZmfAGZGAO/2gAGgAM2gAGZmgAGZGAO/2gAGgAM2gAGZmgAGZGAO/2hAGhAM2hAGZmhAGZGAO/2hAGhAM2hAGZmhAGZGAO/2iAGiAM2iAGZmiAGZGAO/2iAGiAM2iAGZmiAGZGAO/2jAGjAM2jAGZmjAGZGAO/2jAGjAM2jAGZmjAGZGAO/2kAGkAM2kAGZmkAGZGAO/2kAGkAM2kAGZmkAGZGAO/2lAGlAM2lAGZmlAGZGAO/2lAGlAM2lAGZmlAGZGAO/2mAGmAM2mAGZmmAGZGAO/2mAGmAM2mAGZmmAGZGAO/2nAGnAM2nAGZnnAGZGAO/2nAGnAM2nAGZnnAGZGAO/2oAGoAM2oAGZnoAGZGAO/2oAGoAM2oAGZnoAGZGAO/2pAGpAM2pAGZppAGZGAO/2pAGpAM2pAGZppAGZGAO/2qAGqAM2qAGZqqAGZGAO/2qAGqAM2qAGZqqAGZGAO/2rAGrAM2rAGZrrAGZGAO/2rAGrAM2rAGZrrAGZGAO/2sAGsAM2sAGZssAGZGAO/2sAGsAM2sAGZssAGZGAO/2tAGtAM2tAGZttAGZGAO/2tAGtAM2tAGZttAGZGAO/2uAGuAM2uAGZuuAGZGAO/2uAGuAM2uAGZuuAGZGAO/2vAGvAM2vAGZvvAGZGAO/2vAGvAM2vAGZvvAGZGAO/2wAGwAM2wAGZwwAGZGAO/2wAGwAM2wAGZwwAGZGAO/2xAGxAM2xAGZxxAGZGAO/2xAGxAM2xAGZxxAGZGAO/2yAGyAM2yAGZyyAGZGAO/2yAGyAM2yAGZyyAGZGAO/2zAGzAM2zAGZzzAGZGAO/2zAGzAM2zAGZzzAGZGAO/2AAAGAAAM2AAZmAAZGAA/2AAAGAAAM2AAZmAAZGAA/2BAGBAM2BAGZmBAGZGAB/2BAGBAM2BAGZmBAGZGAB/2CAGCAM2CAGZmCAGZGAC/2CAGCAM2CAGZmCAGZGAC/2DAGDAM2DAGZmDAGZGAD/2DAGDAM2DAGZmDAGZGAD/2EAGEAM2EAGZmEAGZGAE/2EAGEAM2EAGZmEAGZGAE/2FAGFAM2FAGZmFAGZGAF/2FAGFAM2FAGZmFAGZGAF/2GAGGAM2GAGZmGAGZGAG/2GAGGAM2GAGZmGAGZGAG/2HAGHAM2HAGZmHAGZGAH/2HAGHAM2HAGZmHAGZGAH/2IAGIAM2IAGZmIAGZGAI/2IAGIAM2IAGZmIAGZGAI/2JAGJAM2JAGZmJAGZGAJ/2JAGJAM2JAGZmJAGZGAJ/2KAGKAM2KAGZmKAGZGAJ/2KAGKAM2KAGZmKAGZGAJ/2LAGLAM2LAGZmLAGZGAL/2LAGLAM2LAGZmLAGZGAL/2MAGMAM2MAGZmMAGZGAM/2MAGMAM2MAGZmMAGZGAM/2NAGNAM2NAGZmNAGZGAN/2NAGNAM2NAGZmNAGZGAN/2OAGOAM2OGOZmOGOZGAO/2OAGOAM2OGOZmOGOZGAO/2PAGPAM2PAGZmPAGZGAO/2PAGPAM2PAGZmPAGZGAO/2QAGQAM2QAGZmQAGZGAO/2QAGQAM2QAGZmQAGZGAO/2RAGRAM2RAGZmRAGZGAO/2RAGRAM2RAGZmRAGZGAO/2SAGSAM2SAGZmSAGZGAO/2SAGSAM2SAGZmSAGZGAO/2TAGTAM2TAGZmTAGZGAO/2TAGTAM2TAGZmTAGZGAO/2UAGUAM2UAGZmUAGZGAO/2UAGUAM2UAGZmUAGZGAO/2VAGVAM2VAGZmVAGZGAO/2VAGVAM2VAGZmVAGZGAO/2WAGWAM2WAGZmWAGZGAO/2WAGWAM2WAGZmWAGZGAO/2XAGXAM2XAGZmXAGZGAO/2XAGXAM2XAGZmXAGZGAO/2YAGYAM2YAGZmYAGZGAO/2YAGYAM2YAGZmYAGZGAO/2ZAGZAM2ZAGZmZAGZGAO/2ZAGZAM2ZAGZmZAGZGAO/2aAGaAM2aAGZmaAGZGAO/2aAGaAM2aAGZmaAGZGAO/2bAGbAM2bAGZmbAGZGAO/2bAGbAM2bAGZmbAGZGAO/2cAGcAM2cAGZmcAGZGAO/2cAGcAM2cAGZmcAGZGAO/2dAGdAM2dAGZmdAGZGAO/2dAGdAM2dAGZmdAGZGAO/2eAGeAM2eAGZmeAGZGAO/2eAGeAM2eAGZmeAGZGAO/2fAGfAM2fAGZmfAGZGAO/2fAGfAM2fAGZmfAGZGAO/2gAGgAM2gAGZmgAGZGAO/2gAGgAM2gAGZmgAGZGAO/2hAGhAM2hAGZmhAGZGAO/2hAGhAM2hAGZmhAGZGAO/2iAGiAM2iAGZmiAGZGAO/2iAGiAM2iAGZmiAGZGAO/2jAGjAM2jAGZmjAGZGAO/2jAGjAM2jAGZmjAGZGAO/2kAGkAM2kAGZmkAGZGAO/2kAGkAM2kAGZmkAGZGAO/2lAGlAM2lAGZmlAGZGAO/2lAGlAM2lAGZmlAGZGAO/2mAGmAM2mAGZmmAGZGAO/2mAGmAM2mAGZmmAGZGAO/2nAGnAM2nAGZnnAGZGAO/2nAGnAM2nAGZnnAGZGAO/2oAGoAM2oAGZnoAGZ
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 7687

{
  "XmlKeyInfo": "",
  "ExternalUserId": "6222662c-d0c5-4077-8942-be328999fdbd",
  "QRCode":
    "R0lGODlhLAESAfcAAAAAAMwAAZgAAmQAAZAAA/wArAAARmWArZgArmQArzAAR/wBVAABVMwBVZgBVMQBvZABV/wCAAACAMwCAZgCAmQCAzACA/wCqAACqMwCqZgCqmQCqzACq/wDVAADVWmDVZgDVmQDVzADV/wD/AAD/MwD/ZgD/mQD/zAD//zMAADMAMzMAZjMAmTMAzDMA/zMrADMrMzMrZjMrMTmrZDMr/zNVADNVmzNVZjNVmTNVzDNV/zOAAADOAMzOAZjOAmTOAzDOA/zOqADOqMzOqZjOqmTOqzDOq/zPVADPVMzPVZjPVmTPVzDPV/zP/ADP/MzP/ZjP/mTP/zDP//2YAAGYAM2YAZmYAmWYAZGYA/2YrAGYrM2YrZmYrmWYrzGYr/2ZVAGZVM2ZVZmZVmWZVzGZV/2aAAGaAM2aAZmAlhAAGaA/2aAGaAM2aAZmAlhAGaA/2bYAGbYAM2bYzmYbYzGYb/2bYAGbYAM2bYzmYbYzGYb/2cYAGcYAM2cYzmYcYzGYc/2cYAGcYAM2cYzmYcYzGYc/2dYAGdYAM2dYzmYdYzGYd/2dYAGdYAM2dYzmYdYzGYd/2eYAGeYAM2eYzmYeYzGYe/2eYAGeYAM2eYzmYeYzGYe/2fYAGfYAM2fYzmYfYzGYf/2fYAGfYAM2fYzmYfYzGYf/2gYAGgYAM2gYzmYgYzGYg/2gYAGgYAM2gYzmYgYzGYg/2hYAGhYAM2hYzmYhYzGYh/2hYAGhYAM2hYzmYhYzGYh/2iYAGiYAM2iYzmYiYzGYi/2iYAGiYAM2iYzmYiYzGYi/2jYAGjYAM2jYzmYjYzGYj/2jYAGjYAM2jYzmYjYzGYj/2kYAGkYAM2kYzmYkYzGYk/2kYAGkYAM2kYzmYkYzGYk/2lYAGlYAM2lYzmYlYzGYl/2lYAGlYAM2lYzmYlYzGYl/2mYAGmYAM2mYzmYmYzGYm/2mYAGmYAM2mYzmYmYzGYm/2nYAGnYAM2nYzmYnYzGYn/2nYAGnYAM2nYzmYnYzGYn/2oYAGoYAM2oYzmYoYzGYo/2oYAGoYAM2oYzmYoYzGYo/2pYAGpYAM2pYzmYpYzGYp/2pYAGpYAM2pYzmYpYzGYp/2qYAGqYAM2qYzmYqYzGYq/2qYAGqYAM2qYzmYqYzGYq/2rYAGrYAM2rYzmYrYzGYr/2rYAGrYAM2rYzmYrYzGYr/2sYAGsYAM2sYzmYsYzGYs/2sYAGsYAM2sYzmYsYzGYs/2tYAGtYAM2tYzmYtYzGYt/2tYAGtYAM2tYzmYtYzGYt/2uYAGuYAM2uYzmYuYzGYu/2uYAGuYAM2uYzmYuYzGYu/2vYAGvYAM2vYzmYvYzGYv/2vYAGvYAM2vYzmYvYzGYv/2wYAGwYAM2wYzmYwYzGYw/2wYAGwYAM2wYzmYwYzGYw/2xYAGxYAM2xYzmYxYzGYx/2xYAGxYAM2xYzmYxYzGYx/2yYAGyYAM2yYzmYyYzGYy/2yYAGyYAM2yYzmYyYzGYy/2zYAGzYAM2zYzmYzYzGYz/2zYAGzYAM2zYzmYzYzGYz/2AAAGAAAM2AAZmAAZGAA/2AAAGAAAM2AAZmAAZGAA/2BAGBAM2BAGZmBAGZGAB/2BAGBAM2BAGZmBAGZGAB/2CAGCAM2CAGZmCAGZGAC/2CAGCAM2CAGZmCAGZGAC/2DAGDAM2DAGZmDAGZGAD/2DAGDAM2DAGZmDAGZGAD/2EAGEAM2EAGZmEAGZGAE/2EAGEAM2EAGZmEAGZGAE/2FAGFAM2FAGZmFAGZGAF/2FAGFAM2FAGZmFAGZGAF/2GAGGAM2GAGZmGAGZGAG/2GAGGAM2GAGZmGAGZGAG/2HAGHAM2HAGZmHAGZGAH/2HAGHAM2HAGZmHAGZGAH/2IAGIAM2IAGZmIAGZGAI/2IAGIAM2IAGZmIAGZGAI/2JAGJAM2JAGZmJAGZGAJ/2JAGJAM2JAGZmJAGZGAJ/2KAGKAM2KAGZmKAGZGAJ/2KAGKAM2KAGZmKAGZGAJ/2LAGLAM2LAGZmLAGZGAL/2LAGLAM2LAGZmLAGZGAL/2MAGMAM2MAGZmMAGZGAM/2MAGMAM2MAGZmMAGZGAM/2NAGNAM2NAGZmNAGZGAN/2NAGNAM2NAGZmNAGZGAN/2OAGOAM2OGOZmOGOZGAO/2OAGOAM2OGOZmOGOZGAO/2PAGPAM2PAGZmPAGZGAO/2PAGPAM2PAGZmPAGZGAO/2QAGQAM2QAGZmQAGZGAO/2QAGQAM2QAGZmQAGZGAO/2RAGRAM2RAGZmRAGZGAO/2RAGRAM2RAGZmRAGZGAO/2SAGSAM2SAGZmSAGZGAO/2SAGSAM2SAGZmSAGZGAO/2TAGTAM2TAGZmTAGZGAO/2TAGTAM2TAGZmTAGZGAO/2UAGUAM2UAGZmUAGZGAO/2UAGUAM2UAGZmUAGZGAO/2VAGVAM2VAGZmVAGZGAO/2VAGVAM2VAGZmVAGZGAO/2WAGWAM2WAGZmWAGZGAO/2WAGWAM2WAGZmWAGZGAO/2XAGXAM2XAGZmXAGZGAO/2XAGXAM2XAGZmXAGZGAO/2YAGYAM2YAGZmYAGZGAO/2YAGYAM2YAGZmYAGZGAO/2ZAGZAM2ZAGZmZAGZGAO/2ZAGZAM2ZAGZmZAGZGAO/2aAGaAM2aAGZmaAGZGAO/2aAGaAM2aAGZmaAGZGAO/2bAGbAM2bAGZmbAGZGAO/2bAGbAM2bAGZmbAGZGAO/2cAGcAM2cAGZmcAGZGAO/2cAGcAM2cAGZmcAGZGAO/2dAGdAM2dAGZmdAGZGAO/2dAGdAM2dAGZmdAGZGAO/2eAGeAM2eAGZmeAGZGAO/2eAGeAM2eAGZmeAGZGAO/2fAGfAM2fAGZmfAGZGAO/2fAGfAM2fAGZmfAGZGAO/2gAGgAM2gAGZmgAGZGAO/2gAGgAM2gAGZmgAGZGAO/2hAGhAM2hAGZmhAGZGAO/2hAGhAM2hAGZmhAGZGAO/2iAGiAM2iAGZmiAGZGAO/2iAGiAM2iAGZmiAGZGAO/2jAGjAM2jAGZmjAGZGAO/2jAGjAM2jAGZmjAGZGAO/2kAGkAM2kAGZmkAGZGAO/2kAGkAM2kAGZmkAGZGAO/2lAGlAM2lAGZmlAGZGAO/2lAGlAM2lAGZmlAGZGAO/2mAGmAM2mAGZmmAGZGAO/2mAGmAM2mAGZmmAGZGAO/2nAGnAM2nAGZnnAGZGAO/2nAGnAM2nAGZnnAGZGAO/2oAGoAM2oAGZnoAGZGAO/2oAGoAM2oAGZnoAGZGAO/2pAGpAM2pAGZppAGZGAO/2pAGpAM2pAGZppAGZGAO/2qAGqAM2qAGZqqAGZGAO/2qAGqAM2qAGZqqAGZGAO/2rAGrAM2rAGZrrAGZGAO/2rAGrAM2rAGZrrAGZGAO/2sAGsAM2sAGZssAGZGAO/2sAGsAM2sAGZssAGZGAO/2tAGtAM2tAGZttAGZGAO/2tAGtAM2tAGZttAGZGAO/2uAGuAM2uAGZuuAGZGAO/2uAGuAM2uAGZuuAGZGAO/2vAGvAM2vAGZvvAGZGAO/2vAGvAM2vAGZvvAGZGAO/2wAGwAM2wAGZwwAGZGAO/2wAGwAM2wAGZwwAGZGAO/2xAGxAM2xAGZxxAGZGAO/2xAGxAM2xAGZxxAGZGAO/2yAGyAM2yAGZyyAGZGAO/2yAGyAM2yAGZyyAGZGAO/2zAGzAM2zAGZzzAGZGAO/2zAGzAM2zAGZzzAGZGAO/2AAAGAAAM2AAZmAAZGAA/2AAAGAAAM2AAZmAAZGAA/2BAGBAM2BAGZmBAGZGAB/2BAGBAM2BAGZmBAGZGAB/2CAGCAM2CAGZmCAGZGAC/2CAGCAM2CAGZmCAGZGAC/2DAGDAM2DAGZmDAGZGAD/2DAGDAM2DAGZmDAGZGAD/2EAGEAM2EAGZmEAGZGAE/2EAGEAM2EAGZmEAGZGAE/2FAGFAM2FAGZmFAGZGAF/2FAGFAM2FAGZmFAGZGAF/2GAGGAM2GAGZmGAGZGAG/2GAGGAM2GAGZmGAGZGAG/2HAGHAM2HAGZmHAGZGAH/2HAGHAM2HAGZmHAGZGAH/2IAGIAM2IAGZmIAGZGAI/2IAGIAM2IAGZmIAGZGAI/2JAGJAM2JAGZmJAGZGAJ/2JAGJAM2JAGZmJAGZGAJ/2KAGKAM2KAGZmKAGZGAJ/2KAGKAM2KAGZmKAGZGAJ/2LAGLAM2LAGZmLAGZGAL/2LAGLAM2LAGZmLAGZGAL/2MAGMAM2MAGZmMAGZGAM/2MAGMAM2MAGZmMAGZGAM/2NAGNAM2NAGZmNAGZGAN/2NAGNAM2NAGZmNAGZGAN/2OAGOAM2OGOZmOGOZGAO/2OAGOAM2OGOZmOGOZGAO/2PAGPAM2PAGZmPAGZGAO/2PAGPAM2PAGZmPAGZGAO/2QAGQAM2QAGZmQAGZGAO/2QAGQAM2QAGZmQAGZGAO/2RAGRAM2RAGZmRAGZGAO/2RAGRAM2RAGZmRAGZGAO/2SAGSAM2SAGZmSAGZGAO/2SAGSAM2SAGZmSAGZGAO/2TAGTAM2TAGZmTAGZGAO/2TAGTAM2TAGZmTAGZGAO/2UAGUAM2UAGZmUAGZGAO/2UAGUAM2UAGZmUAGZGAO/2VAGVAM2VAGZmVAGZGAO/2VAGVAM2VAGZmVAGZGAO/2WAGWAM2WAGZmWAGZGAO/2WAGWAM2WAGZmWAGZGAO/2XAGXAM2XAGZmXAGZGAO/2XAGXAM2XAGZmXAGZGAO/2YAGYAM2YAGZmYAGZGAO/2YAGYAM2YAGZmYAGZGAO/2ZAGZAM2ZAGZmZAGZGAO/2ZAGZAM2ZAGZmZAGZGAO/2aAGaAM2aAGZmaAGZGAO/2aAGaAM2aAGZmaAGZGAO/2bAGbAM2bAGZmbAGZGAO/2bAGbAM2bAGZmbAGZGAO/2cAGcAM2cAGZmcAGZGAO/2cAGcAM2cAGZmcAGZGAO/2dAGdAM2dAGZmdAGZGAO/2dAGdAM2dAGZmdAGZGAO/2eAGeAM2eAGZmeAGZGAO/2eAGeAM2eAGZmeAGZGAO/2fAGfAM2fAGZmfAGZGAO/2fAGfAM2fAGZmfAGZGAO/2gAGgAM2gAGZmgAGZGAO/2gAGgAM2gAGZmgAGZGAO/2hAGhAM2hAGZmhAGZGAO/2hAGhAM2hAGZmhAGZGAO/2iAGiAM2iAGZmiAGZGAO/2iAGiAM2iAGZmiAGZGAO/2jAGjAM2jAGZmjAGZGAO/2jAGjAM2jAGZmjAGZGAO/2kAGkAM2kAGZmkAGZGAO/2kAGkAM2kAGZmkAGZGAO/2lAGlAM2lAGZmlAGZGAO/2lAGlAM2lAGZmlAGZGAO/2mAGmAM2mAGZmmAGZGAO/2mAGmAM2mAGZmmAGZGAO/2nAGnAM2nAGZnnAGZGAO/2nAGnAM2nAGZnnAGZGAO/2oAGoAM2oAGZnoAGZ
```



```
"KeyExpirationTime": "2019-06-29T00: 00: 00"
}
```

Назначение пользователю существующего токена аутентификации MobileAuth

AssignUserMobileAuthToken

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/mobileauth/assign
Параметры	id - Глобальный идентификатор пользователя AssignMobileAuthTokenInfo - Информация о существующем токене аутентификации MobileAuth
Возвращаемое значение	UserMobileAuthInfo - Информация о созданном ключе аутентификации MobileAuth

Пример запроса

```
POST https://<hostname>/STS/ums/user/9fba6076-3509-4e8e-abc1-dae6d052a230/mobileauth/assign HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 148

{
  "Serial": "418551002761789440",
  "UserContactInfo": "78889996655",
  "UserContactInfoType": "PhoneNumber",
  "NeedXmlKeyInfo": false,
  "DelayedActivation": false
}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 184

{
  "UserId": "9c937c8e-7e27-4105-b306-974fce87cdd1",
  "KeyExpirationTime": "2019-06-29T00: 00: 00",
  "IsInstalled": false,
  "InstallDate": "0001-01-01T00: 00: 00",
  "HasDelayedKey": false,
  "KeyVersion": 1
}
```

Получение информации о ключе аутентификации MobileAuth

GetUserMobileAuthTokenInfo

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/mobileauth
Параметры	id - Глобальный идентификатор пользователя

ПАРАМЕТР	ЗНАЧЕНИЕ
Возвращаемое значение	UserMobileAuthInfo - Информация о созданном ключе аутентификации MobileAuth

Пример запроса

```
GET https://<hostname>/STS/ums/user/9fba6076-3509-4e8e-abc1-dae6d052a230/mobileauth HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 184

{
  "UserId": "9c937c8e-7e27-4105-b306-974fce87cdd1",
  "KeyExpirationTime": "2019-06-29T00: 00: 00",
  "IsInstalled": false,
  "InstallDate": "0001-01-01T00: 00: 00",
  "HasDelayedKey": false,
  "KeyVersion": 1
}
```

Повторная отправка кода активации MobileAuth

ResendUserMobileAuthActivationCode

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/mobileauth/activationcode
Параметры	id - Глобальный идентификатор пользователя MobileAuthTokenInfo - Информация о токене аутентификации MobileAuth
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/9fba6076-3509-4e8e-abc1-dae6d052a230/mobileauth/activationcode HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 118

{
  "UserContactInfo": "78889996655",
  "UserContactInfoType": "PhoneNumber",
  "NeedXmlKeyInfo": false,
  "DelayedActivation": false
}
```

Пример ответа

```
HTTP/1.1 200 OK
```

Обновление ключевой информации MobileAuth пользователя

UpdateUserMobileAuthToken

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	PATCH
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/mobileauth
Параметры	id - Глобальный идентификатор пользователя MobileAuthTokenInfo - Информация о токене аутентификации MobileAuth
Возвращаемое значение	MobileAuthUpdateInfoEx - Информация об обновленном ключе аутентификации MobileAuth

Пример запроса

```
PATCH https://<hostname>/STS/ums/user/9fba6076-3509-4e8e-abc1-dae6d052a230/mobileauth HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 118

{
  "UserContactInfo": "78889996655",
  "UserContactInfoType": "PhoneNumber",
  "NeedXmlKeyInfo": false,
  "DelayedActivation": false
}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 7671

{
  "XmlKeyInfo": "",
  "QrCode":
    "R0lGODlhLAESAfCAAAAAAAAAAMwAAZgAAMQAazAAA/wArAAArMwArZgArmQArzAAr/wBVAABVMwBVZgBVmQBVzABV/wCAAACAMwCAZgCamQCA
    zACA/wCqAACqMwCqZgCqmQCqzACq/wDVAADVmDVZgDVmQDVzADV/wD/AAD/MwD/ZgD/mQD/zAD//zMAADMAMzMAZjMAMTMAzDMA/zMrADMrm
    zMrZjMrMTmrzDMr/zNVADNVmzNVZjNVmTNVzDNV/zOAAADOAMzOAZjOAmTOAzDOA/zOqADOqMzOqZjOqmTOqzDOq/zPVADPVMzPVZjPVmTPVzD
    PV/zP/ADP/MzP/ZjP/mTP/zDP//2YAAGYAM2YAZmYAmWYAzGYA/2YrAGYrM2YrZmYrmWYrzGYr/2ZVAGZVM2ZVZmZVmWZVzGVZ/2aAAGaAM2a
    AZmaAmWaAzGaA/2aqAGaqM2aqZmaqmWaqzGaQ/2bVAGbVM2bVZmbVmWbVzGbV/2b/AGb/M2b/Zmb/mWb/zGb//5kAAJkAM5kAZpkAmZkAzJkA
    /5krAJkrM5krZpkrmZkrzJkr/5lVAJlVM5lVZp1VmZlVzJlV/5mAAJmAM5mAZpmAmZmAzJmA/5mqAJmqM5mqZpmqmZmqzJmq/5nVAJnVM5nVZ
    pnVmZnVzJnV/5n/AJn/M5n/Zpn/mZn/zJn//8wAAMwAM8wAZswAmcwAzMwA/8wrAMwrM8wrZswrmcwrzMwr/8xVAMxVM8xVZsxVmcxVzMxV/8
    yAAMyAM8yAZsyAmcyAzMyA/8yqAMYqM8yqZsyqmcyqzMyq/8zVAMzVM8zVZszVmczVzMzV/8z/AMz/M8z/Zsz/mcz/zMz//8AAP8AM/8AZv8
    Amf8AZP8A//8rAP8rM/8rZv8rmf8rzP8r//9VAP9VM/9VZv9Vmfv9VzP9V//+AAP+AM/+AZv+Amf+AzP+A//+qAP+qM/+qZv+qmf+qzP+q//V
    AP/VM//VZv/Vmf/VzP/V///AP//M//Zv//mf//zP///wAAAAAAAAAAAAACH5BAEAAAPwALAAAAASwBAAj/APcJHEiwoMGDCBMqXMiwo
    cOHECNKnEixosWLDGNq3Mixo8ePIEOKHEmypoMmTKFOqXmMypcuXMGPKnEmzps2bOHPq3Mmzp8+fQIMKHUq0qNGjSJmQXcq0qdOnUKNKnUq1qt
    WrWLNq3cq1q9evYmOKHUu2rNmzaNOqXcu2rdu3cOPKnUu3rt27ePPq3cu3r9+/gAMLHky4sOHDiBMrXsy4seOoACJLnky5skHJAzEL1LyPs+X
    N1BNW5px5NIDLkUufBm26degCpGG/JujZ9eqDnxXad110d2vUt2shN507N3DjrfJF3Js7ct/LYtGerVv0bofLjzqH/zP68+XTW1Jtrck8ewvZo
    4N+Xc4c+Xj168uet19e9vj1P++bTF5/PXrxp7MPd9p56+L1HUXUGRidgfuAxmGCB6QkFYMP8mfhghXyZp5wovmH4YASIdjfhxSOqJ90C014k
    4ocwhdegAPuBmCED17nYIj/UUiJgju2mCBDKv/aJNx+GLbXHYn0xYebbTOWeKGM0tb4pIdAegefjxJ6iGWUCpoYEYIAHukjkVD+KOVkL6aZop
    UEXtiblmzuqCaPX+a4pIZdskkmk1yeikaLgCbPj5FvBkiok0X095Cnd4KI6JyQIjqknZG2qWSMcMJI1KR/Ztijog4x6qiZngZaKacvomiPy
    OaWeQNXHqqqZ5Itlhp1Umquugqx45Z5mn8tmopG5umumnnbIY54bC1odsq84tOqWtVxbL6rJAyYrtoQ1tWauI6z0L6reUrpnsuGfAcy610m1L
    66+r3ipntUo6yyWYzOKZK6rmrpsqttvFK5+h5dJ5LChvnintuQj/83unw/vqiW6fAxe68L+4nr1nwgYryzDG1cKbcZKzGinwsEN5jLHJDVNJc
    andqvqxJaGPHDJJ6tsbH20zurnz48GDbPPQHm9tMQ9I72ysFnyDPTRSUF9tNFUK1201VBPNsmwQfGMddVSXx121mJrPTbYZi+tr1g4j5zdrR
    vPmyHLcJ88qpcGB9b2iFyTi7DcEN+Y78WCkwrrXHv3Gu3N0wLuMspNxrXr4YdDtvjB3KY70eYF193yyJq/XbG5II90tlGic/4x425v3Tnrakf
    stOny0lv76YwWmTrLvTLKL8mus+s16C/zSvzplEMut+/hhkozsqPmGvu9itdbMN/H/5ra1PKqpy679JTbbbru1Wteet6Brpt8T9wPL+rgpIfP
    LqZ44n3+79h/bjPqwf8evcbly5e3gvU3+AmqX+Lr2LZmdqmu9e916ENZ7/zmuNYVsFbgA978yibA7KXsgQ2E3PjG1b8Bam9u1SuQCp+nMOpZL
    4QB21zk1Pe4FR4vcQYnVfqxLD8pY2DH5ShAWl4QRvS7YRH7B4Ev4dEmZ0NYD6x3xAZ6DAj5uxxFVsb/5zXwXR+7YsbzJYHuYeJ1UUQesZjGB
    NReDyR7e9uCbSY7dLHQg2iUYRzBwXEyc8J+rsjEpx18jWp8ccunB6hrQjdsdIxwtmUG5NI6AgXzbJPIk2Tn5aBOMEVQXDDnIHKtJcmAktGTU
    iwwFz3Kqk8UKXQPfFcSehnNYoK+muWGSud0jbJAmZxjp78e+GuXw1Hrd4sBkGMJF/XCUM4+e39eUEWqUU5imhyck7JvGRizw1N1XnFGgqcyI1
    vGIICz1LYeJtgpB0ITfzd0xJitNo91viIQHoSEper53M/P/hL6VYmWmak2NEtKE2G8nP891xm/DcGT+rCTuhxb0TETznJQFZzGVKjn7pFCOSb
    Pm0hf3RlDhsIyp3eUtewux/v7ydlTWIwDDW8nI4pF/fJLq2kxY010IkYEf3aE4Q+uuEMjUpTfEHxUTi1II/7acGX1pErGF0jhhkJvlc61Srcd
    KVdQzORBOqQP35c4T16+nYrnJVP7aSimh94kaDidSzjpS1EizqUYnaVF29z3D47Cpd/+nRqkZ0qyCc6z3rar58zjOuNfVh8aYIV8Uqlan0lM1
    Ma+hFRuJVihodLF9501b5VTGwr7nr7Z6a1Gm+c4EWdGMid3rZzf4Vs00iZE7/s0k7oK72a2z8JlcXaa2DLnZ9dRxtC0vr2dMCLKSafafewtG
```

```
3xo1immFbF7/+dmSQtRs1V0nEMkpTYXKErQUHSRlrXvXton3utvt7W8F28X27rWl57WmPKGW3fmCzVuR1ajHmBtUM5bVuijVrVhdK1CLsrevm
B2qWe2K3M0qVL0HjKpNhZld71I1tRZN6n9Jm9HqEk7Co3RufmGCX80yFpFZPBAWbys5K75Vh5UVLSxRi04WT3atYeyva9fo0USqMo1HFReGQR
w8lMq2nsq9sIt97EEU76yM+2tfd0nJ07YWNiNYVeuHR8ySw8WUoHIFKzGzHM4cwe9cOWwA82c/9uGJhnKQHzLWxcYjCy9sFsFtJ3X0hN4/q
SuX1+65/d6thB51l5exalcT1naNMC072u7DBJpdvc0CZ6zz3uZfNa68lMK/GTv2Xop0085nKG9biJ9fMxC1xRZJqYm6GmGZfbBWEmo/q9D2vw
c0lZwZUm1dMBDmSt4zzhGzN6lW596LHJy0Ae31TYL4Txo6lr5yZ2dtfmpsee00axd4EZ7kd5kLNqsZVpwc9TaJVZrt+/TbHDLurLJzXZXnUtaX
8P7sHycNYlXTNENnmx8s7gmNud5EIXtaBptq2r22rZcL9Z4f2tsn8nGu/ZhnfNUV50p/+tY9peFKpXri+Dr0lvCyRnkrC4Pbmx3T1xQaeV2i
H1ZUZjS0nKZhykACZ2vps16fGq/NX/B06jh1Mu9Mthl+CUJi0YY77pmUOXzb/+eKdh3reI9xzJNgc5mt/49DIW8qE3TnqyiR1fOccY2eqesp6
hTvCQG33pVaenqAMtZJoe/dA0kbjTF87VkgNc7dp+9lItXu7gzjhHp063E4W7629W5NWQGXj0JSzHXkdZ8Flm/EcNH+Ea13naw+V6EBH8dXXy
G9f1l1fU1W7p6jug7VmKN+mvnrPqs1h7HpCd8R16f99i3vbbhIrr3th9z6ZTN+I7xvSboVfflW03agzRx7wYEOVa2KNPldpjGmY/1tyhuY7GIuM
vBpaV/Os/vW28f274n7WlaK9Lyet75B3c0S1NQxV6d/z56Iw/Y/N8d4ZzmFY2HaN13ZXDmfJeGdzAnbfVGfe2HW9hXErI3f/yXf8w3a1Lmf2
LjeSKma5FEbkt3gMVXfadYX/HbBcmgib1gewnciqIe5oWUtAncLeXTFzkZKMHgvr3gsdHgsFngH2XWsnIQzjIgkY2fF8mgNwFg1P3fRXYa6i
FZ0FWWwK4gBzHYinWhA8ITgcHYUKUgScxguxnhYpEZLDFcvEHeBOWdQu3d+dnfApnbukHhaA3e0PkgA2IdVkhcUkoh5B2hq0WbFpIgX7oeFzB
h/wmaXXfGvP3cYPYiIXocVoxdC14b0r4bgzXgS8mdVS3aXZHfxhnYx6Ibv+WGDlsqGB3CFEXCFNoB2yh+HCVOHKmgIdXp4k4KHliV4AORXRbH
Iq1t4VHMikeJ4aCKHuA1mTbU4PrN40YV4vDqHsRBocBKGGUuBT2posUxoSzcIu/aIOwL260Vo0Jp36qmIihB1ZuiG+YeI1BKIU30I4u9oXdp4
h3Zlht9o1oh4o00IQ5SI40l3rN13f22Ievxo5SNWDQJo73iGeiVo/R11hCxX0o5k4WaH5d14iGeILPt4v4aH+3KJHyJYwIaFksKHmi920xqIg
TCH6wVmwL+Ieid347NIbrxZFeh4doSHFXCivMeFlgOBiYr4v8mF8rp5Evx1nbBoymp4AvEZQSNdMa8hDuGZ1f0Z7smhPCVaK/6UmkXHHdyB5
kC03h0Dog+5YhTFIzzLJlSuIlV01j+5HjeqIjYVHkkXHlL3nGyfXg1oYiQ8JlwG4ki5Jl4+4T8j4ewW2TQKGd22Jf4v5EcR4eCgp04dpj4n5e
F/JhXhZk5ip12mpgyZ5du8Xfs23bIlHms0GZDBYf1Npk2fJiojXhR0pg2jpSY0JgKrpqKVZl4DZjVEomxb5kBoGhBa2mlHZkbj0mhAZmykJca
znmn5XF5UpeIm5it1HfqlGciyHmKwJlTRnZQ4mmmDnnNnJkgLJa5S5nT+5EmTwm52Xfo02ni2pT1SpncWZniTxX2Kpaji3X+NVZk5Zd0gXaZb
GcP/GuI77SwcZ6J9NN34B6j5BB5zah5wHyp+LCJr/uZTB6aADSPatUmToChyniImaJ5zR2Ix2qXw6CaGzU4Zq5pM+x4utZ5vn2YYmJ5Qden0W
d4woZ5Y0qpRPSIXpG627lZpgmZJh9pT/t365uJ77B50c+ZiIZaTVhQqbeJnSdpxFCG1Dqo3vSF+tWKWhSWVe6FMNum6QaYINKXdoSsqYsaqXnK
KZyiZWC2JnBuJFCCIC5aJpxGpg/l5wjeqJNOYqXWIFeaaUQ9J5Rqqanppcy2oueaKH9CV43yaUqmoZZ+YcXmhTDA6DTBX/z5YqkT6K2KG4C+h
SbiqH9V30MSaSJjJqj/OtqnZtqPgUZXCAqBPYmor/qkzjhvvFiSBnqdtEqnToireZV5EHh721h5szpYwZqFmoil38mNwrmlcLgqoMWF9Relwa6S
gj5aEN7dS0kWK1rppvVlqQsiZJ0abv3qSv/qJ1JqmhImpEamHUJmfNVV2LMqJYNQ0zElriyqJ6WqGU8WXk2mGnmadeieU9ly3xqhedmvWzp3
GmeeT6l0jvhlM18mhywdeAdmXEoucDGij5sidsFed6yqKyulsFiTglXqwkacqv9iZAUeqWiadx+qZKpub0tqpcamza8ekLsuw0QqqkbmTogu0W
PiM4ypZ0UpoOMmT0eqNHvpx+Xp/s6eG/xjLhPEoqCF6qR+mkghZtYQ4rVOYqHaIp2PqtV0KtVwIqmN7lSmltDtnsjxbr29HTX5ZjIuzkVzPev
LGpLXZf7nXpMMqomCprQAbt/cIuBkruJwVq8/KqiGbjTH5ikE7uQz5otSah43aiYCIrSqaQ9vKVn5IiYzopYAYtKcLuqlruWbHdGGageFqtqe
rulzrg4V6kR0Jq2U6sZPKaox6qJy3efRqhFJ3oBlZubs7o3SYsHhbpP0qujHIu3TrvL8rfcFbjoJJvMors2PjFEoaZsa5t+raugabtLy6kjr5
s655jWnJsi3ZbUU5ujuah7MbW+2rjPbnszpxftfJskpblv+QV3kf2XChu5NzaqKZW65za6cJ7G1v2atDK5XYu7YK3L/V06/A06ARzKPPiZTMC
6RyisGmm4obzBTy2KaT97B76VfzSYTrC8MXm4xaq58tepyBG5Dk255L66NukcLS6LHfm6H0iJHQ65gHjMJ3i1Dim5T4arubGIUeLnRu6FrWa
LFScWwx65107CWGLMo27cuMZNMPJoBS4ZQbGsG55EB05v6NaVve6Ru+bpkyKFsOKrTqKVwDImLp5mmtrE7+o+nyV8THK/yycdCa7fJCcgVR6i
DXKzsScDDhYjm06l6ar9sKZ48vJtYe3gCbMM2i5mU7K5/rMKLzJs9PBANmNnNQ4Z87Yqs6PmoseumM9yPltlyyxrFXQmuS2nJZzyjYqwSRLlg
lKzGwgu+6Md2IumNQSq9F1HC0ujF2kvKZGu0pxmMPwyp1DqTaEuvlkrIUJu+b/vCmqrNfzq/I1yxrrr09fumxFm0tsySnGqpC0y006qP6VywF
PubxhufMozHbLtl+suvZehvtSjJ/vyJpTqozeqbyEyhFNm5HFy+t3yrJNrQ7zrQICzFt+uo2EidQZqp8Lq1mUm7J9uqgtmdH52Aib2gg8rQXv
PJ3cqmwf/st7QM0jGM0vbcyq15r/v6upk6iWmbpeqsyxPrwKxcxM97tan8d0+7y4Ybv0ddzV6pw20Looc41PKssBisxTfn1SKMR3m7lebazR0
MzW0Mmo0M1bBrrAmNv2W9pmHpuIGIsm6Hy1fcwJr0qWca159rzmg8q4BsuksMpkS81Rr1tR/Ko6jctSfsrE74wGJY064XtmXZsMIazh0b1jFa
wVgM2Y1twwaL2Vlor/Q52Rdt2Rcsqw36xKId1IaNHkVK0qzb1c582Dv4z7opfUw1mTHNFTj824qL2xZbxmQ8yVdMvWCs07tNuP6qbCGor5vrz
rZal9voi31stWQ1Ri9rgDrneMPVPZj03NWo2tTPTZbRrd3ftd2Zzc3sbGucktQVWbxxSZFzrN4+TIUDjL6Gat5hyN30/ZcyyZcKKdECztPh3N
9lme6EnPLNUGpiIq8gEvcanzM8lK9Db+5v0i899idXVundIHaoIXc8brdjne64U7eEU/trzGNZv2KXB6mVTjYid7JcrvtSbLdZOLcu0h6PjzMg
mSs0lXbpdB0FCyuMD7rQ/7rBB/qutPMtdntxw7Nc9Tr93fd2+G9t72t+9/Rhe/uVgHuZiPuZkXuZmfuZonuZquvZs3uZu/uZwHudyPud0Xud2
fud4NJ7ner7nfN7nfV7ngB7ogj7ohF7ohn7oiJ7oir7ojN7ojv7okB7pkj7p1F7p1n7pmC7pAQEA0w==" ,
"KeyExpirationTime": "2019-06-29T00: 00: 00"
}
```

Задание отпечатка мобильного устройства пользователя

SetUserMobileAuthDeviceThumbprint

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/mobileauth/thumbprint
Параметры	id - Глобальный идентификатор пользователя deviceThumbprint - Отпечаток устройства пользователя
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/9fba6076-3509-4e8e-abc1-dae6d052a230/mobileauth/thumbprint HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 14

"12AC65BE0424"
```

Пример ответа

```
HTTP/1.1 200 OK
```

Удаление ключа аутентификации MobileAuth

DeleteUserMobileAuthToken

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	DELETE
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/mobileauth
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	-

Пример запроса

```
DELETE https://<hostname>/STS/ums/user/9fba6076-3509-4e8e-abc1-dae6d052a230/mobileauth HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
```

Настройка аутентификации myDSS Client

- Назначить пользователю существующее мобильное устройство
- Создание QR-кода с Kinit
- Получение существующего QR-кода с Kinit
- Повторная отправка кода активации на Kinit
- Удалить QR-код с Kinit

Назначить пользователю существующее мобильное устройство

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/mydss/assign
Параметры	id - Глобальный идентификатор пользователя MyDssAssignKeyRequest - Информация о добавляемом устройстве
Возвращаемое значение	MyDssKeyInfo - Информация об устройстве

Пример запроса

POST https://<hostname>/STS/ums/user/8d16086d-676c-49d6-bf49-09e6f936a596/mydss/assign HTTP/1.1
Content-Type: application/json; charset=utf-8

{"Kid": "73128110"}

Пример ответа

HTTP/1.1 200 OK

```
{
  "Uid": "8d16086d-676c-49d6-bf49-09e6f936a596",
  "Kid": "73128110",
  "DeviceName": "MyApple",
  "NotBefore": 1574869065,
  "NotAfter": 1614440265,
  "State": "NotVerified",
  "UserName": "MdagTestUser-D547B268",
  "Profile": "{ \"Version\": 1, \"Keys\": {
    \"1\": \"%D0%9E%D0%93%D0%A0%D0%9D\\\", \"2\": \"%D0%9E%D0%93%D0%A0%D0%9D%D0%98%D0%9F\\\", \"3\": \"%D0%A1%D0%9D%D0%98%D0%9B%D0%A1\\\", \"4\": \"%D0%98%D0%9D%D0%9D\\\", \"5\": \"%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F%D0%BF%D0%BE%D1%87%D1%82%D0%B0\\\", \"6\": \"%D0%A1%D1%82%D1%80%D0%B0%D0%BD%D0%B0\\\", \"7\": \"%D0%9E%D0%B1%D0%BB%D0%B0%D1%81%D1%82%D1%8C\\\", \"8\": \"%D0%93%D0%BE%D1%80%D0%BE%D0%B4\\\", \"9\": \"%D0%9E%D1%80%D0%B3%D0%B0%D0%BD%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F\\\", \"10\": \"%D0%9F%D0%BE%D0%B4%D1%80%D0%B0%D0%B7%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5\\\", \"11\": \"%D0%9E%D0%B1%D1%89%D0%B5%D0%B5%D0%B8%D0%BC%D1%8F\\\", \"12\": \"%D0%90%D0%B4%D1%80%D0%B5%D1%81\\\", \"13\": \"%D0%94%D0%BE%D0%BB%D0%B6%D0%BD%D0%BE%D1%81%D1%82%D1%8C\\\", \"14\": \"%D0%98%D0%BD%D0%B8%D1%86%D0%B8%D0%B0%D0%BB%D1%8B\\\", \"15\": \"%D0%98%D0%BC%D1%8F\\\", \"16\": \"%D0%A4%D0%B0%D0%BC%D0%B8%D0%BB%D0%B8%D1%8F\\\" }, \"Values\": { } }\",
  \"NonceRequired\": true
}
```

Типовые ошибки

HTTP-код	ошибка	описание
404		Пользователь не найден
400	key_not_found	Анонимное устройство с идентификатором [kid] не найдено.

Создание QR-кода с Kinit

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/mydss/init
Параметры	id - Глобальный идентификатор пользователя MyDssKlnitRequest - Информация для отправки кода активации
Возвращаемое значение	MyDssCreatedKeyInfo - Информация об устройстве и K-init

Примечание

Если настройками сервера отключено требование кодов активации Klnit, то параметры Msisdn и Email игнорируются.

Примечание

Если настройками сервера включено требование подтверждения контактной информации, то указанные в параметрах

Msisdn и **Email** должны быть заранее зарегистрированы в профиле пользователя.

Если требование подтверждения контактной информации отключено, то код активации Kinit будет отправлен на указанный номер телефона или Email. Значения номера телефона или Email не заносятся в профиль пользователя.

Пример запроса

```
POST https://<hostname>/STS/ums/user/33f03deb-9e31-42ce-a660-1a456190f4a3/mydss/init HTTP/1.1
Content-Type: application/json; charset=utf-8

{}
```

Пример ответа

```
HTTP/1.1 200 OK

{
  "KeyInfo":
  {
    "EncryptedBlobs": "ASAAA ... ADQBS9V",
    "PublicKey": null,
    "Seed": null,
    "ActivationRequired": false,
    "ServiceUrl": "https://hostname:4430/mydss",
    "Alias": null,
    "Uid": "33f03deb-9e31-42ce-a660-1a456190f4a3",
    "Kid": "73128502",
    "DeviceName": null,
    "NotBefore": 1574880257,
    "NotAfter": 1614451457,
    "State": "Active",
    "UserName": null,
    "Profile": null,
    "NonceRequired": false
  },
  "QrCode": "R01GOD1hLAESAf...",
  "QRCodeData": "{ \"type\": \"Kinit\", \"version\": 1, \"data\": { \"kid\": \"73128502\", \"uid\": \"33f03deb-9e31-42ce-a660-1a456190f4a3\", \"service_url\": \"https://hostname:4430/mydss\", \"key_content\": \"ASAAAB5mAA ... BS9V\", \"activation_required\": false, \"weakness\": true } }"
}
```

HTTP-код	ОШИБКА	ОПИСАНИЕ
404		Пользователь не найден
400	initialization_key_already_exists	У пользователя уже есть ключ инициализации нового устройства.
400	invalid_contact_info	* Параметр [Msisdn] не задан или имеет невалидный формат. * Нет возможности отправить вторую часть ключевой информации: не задана контактная информация пользователя.
400	wrong_operation	* Нет возможности отправить вторую часть ключевой информации: не настроен модуль оповещения. * Контактная информация [contact] типа [contactType] не добавлена для этого пользователя.
400	invalid_license	
500	internal_error	Внутренняя ошибка сервиса. Необходимо обратиться к Администратору.

Получение существующего QR-кода с Kinit

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/mydss/init/get
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	MyDssCreatedKeyInfo - Информация об устройстве и K-init

Пример запроса

```
POST https://<hostname>/STS/ums/user/9fba6076-3509-4e8e-abc1-dae6d052a230/mydss/init/get HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK

{
  "KeyInfo":
  {
    "EncryptedBlobs": "ASAAA ... ADQBS9V",
    "PublicKey": null,
    "Seed": null,
    "ActivationRequired": false,
    "ServiceUrl": "https://hostname:4430/mydss",
    "Alias": null,
    "Uid": "33f03deb-9e31-42ce-a660-1a456190f4a3",
    "Kid": "73128502",
    "DeviceName": null,
    "NotBefore": 1574880257,
    "NotAfter": 1614451457,
    "State": "Active",
    "UserName": null,
    "Profile": null,
    "NonceRequired": false
  },
  "QrCode": "R0lGODlhLAESAf...",
  "QRCodeData": "{ \"type\": \"Kinit\", \"version\": 1, \"data\": { \"kid\": \"73128502\", \"uid\": \"33f03deb-9e31-42ce-a660-1a456190f4a3\", \"service_url\": \"https://hostname:4430/mydss\", \"key_content\": \"ASAAA B5mAA ... BS9V\", \"activation_required\": false, \"weakness\": true } }"
}
```

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404		Пользователь не найден
400	key_not_found	Ключ инициализации не найден.

Повторная отправка кода активации на Kinit

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/mydss/init/resentotp

ПАРАМЕТР	ЗНАЧЕНИЕ
Параметры	id - Глобальный идентификатор пользователя MyDssKInitRequest - Информация для отправки кода активации
Возвращаемое значение	MyDssCreatedKeyInfo - Информация об устройстве и K-init

Примечание

Если настройками сервера включено требование подтверждения контактной информации, то указанные в параметрах **Msisdn** и **Email** должны быть заранее зарегистрированы в профиле пользователя.

Если требование подтверждения контактной информации отключено, то код активации Kinit будет отправлен на указанный номер телефона или Email. Значения номера телефона или Email не заносятся в профиль пользователя.

Пример запроса

```
POST https://<hostname>/STS/ums/user/33f03deb-9e31-42ce-a660-1a456190f4a3/mydss/init/resendotp HTTP/1.1
Content-Type: application/json; charset=utf-8

{"msisdn":"71234567890"}
```

Пример ответа

```
HTTP/1.1 200 OK

{
  "KeyInfo":
  {
    "EncryptedBlobs":"ASAAA ... ADQBS9V",
    "PublicKey":null,
    "Seed":null,
    "ActivationRequired":false,
    "ServiceUrl":"https://hostname:4430/mydss",
    "Alias":null,
    "Uid":"33f03deb-9e31-42ce-a660-1a456190f4a3",
    "Kid":"73128502",
    "DeviceName":null,
    "NotBefore":1574880257,
    "NotAfter":1614451457,
    "State":"Active",
    "UserName":null,
    "Profile":null,
    "NonceRequired":false
  },
  "QrCode":"R0lGODlhLAEsAf...",
  "QRCodeData": "{ \"type\": \"Kinit\", \"version\": 1, \"data\": { \"kid\": \"73128502\", \"uid\": \"33f03deb-9e31-42ce-a660-1a456190f4a3\", \"service_url\": \"https://hostname:4430/mydss\", \"key_content\": \"ASAAAB5mAA ... BS9V\", \"activation_required\": false, \"weakness\": true } }"
}
```

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404		Пользователь не найден
400	key_not_found	Ключ инициализации не найден.

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_contact_info	* Параметр [Msisdn] не задан или имеет невалидный формат. * Нет возможности отправить вторую часть ключевой информации: не задана контактная информация пользователя.
400	wrong_operation	* У ключа инициализации нового устройства нет кода активации. * Нет возможности отправить вторую часть ключевой информации: не настроен модуль оповещения. * Контактная информация [contact] типа [contactType] не добавлена для этого пользователя.

Удалить QR-код с Kinit

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/mydss/init/delete
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	Отсутствует

Пример запроса

POST https://<hostname>/STS/ums/user/9fba6076-3509-4e8e-abc1-dae6d052a230/mydss/init/delete HTTP/1.1

Пример ответа

HTTP/1.1 200 OK

Получение QR-код с Nonce

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/mydss/verify/get
Параметры	id - Глобальный идентификатор пользователя MyDssGetVerificationDataRequest - Информация об устройстве
Возвращаемое значение	Отсутствует

Пример запроса

POST https://<hostname>/STS/ums/user/9fba6076-3509-4e8e-abc1-dae6d052a230/mydss/verify/get HTTP/1.1
Content-Type: application/json; charset=utf-8

{"Kid": "73128110"}

Пример ответа

HTTP/1.1 200 OK

```
{
  "QRCode": "R0lGODlhLA ... RczdZ8zdiczdq8zdzczd78zeAczuI8zuRczuZ8zuisEgEBADs=",
  "QRCodeData": "{ \"type\": \"Verification\", \"version\": 1, \"data\": { \"kid\": \"73128110\", \"uid\": \"8d16086d-676c-49d6-bf49-09e6f936a596\", \"service_url\": \"https://simdss.cryptopro.ru:4430/mydss\", \"seed\": \"DQiMn1xAvdQFQlAR4ceB3NkWaevDppM7k1IV0ZZssTc=\", \"nonce\": \"cKx+pdmejI/SRz30Qex32wd6vNF7oVni6LSAf7Sf0Zw=\\\" } }\",
  \"Data\":
    {
      \"type\": \"Verification\",
      \"version\": 1,
      \"data\":
        {
          \"kid\": \"73128110\",
          \"uid\": \"8d16086d-676c-49d6-bf49-09e6f936a596\",
          \"service_url\": \"https://hostname:4430/mydss\",
          \"seed\": \"DQiMn1xAvdQFQlAR4ceB3NkWaevDppM7k1IV0ZZssTc=\",
          \"nonce\": \"cKx+pdmejI/SRz30Qex32wd6vNF7oVni6LSAf7Sf0Zw=\"
        }
    }
}
```

Получает информацию о ключах аутентификации myDSS

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/mydss
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	UserMyDssAuthInfo - сведения об устройствах пользователя

Пример запроса

POST https://<hostname>/STS/ums/user/8d16086d-676c-49d6-bf49-09e6f936a596/mydss HTTP/1.1

Пример ответа

```
HTTP/1.1 200 OK

{
  "UserId": "8d16086d-676c-49d6-bf49-09e6f936a596",
  "Keys":
  [
    {
      "Uid": "8d16086d-676c-49d6-bf49-09e6f936a596",
      "Kid": "73128110",
      "DeviceName": "MyApple",
      "NotBefore": 1574869065,
      "NotAfter": 1614440265,
      "State": "NotVerified",
      "UserName": "MdagTestUser-D547B268",
      "Profile": "{ \"Version\": 1, \"Keys\":
{ \"1\": \"%D0%9E%D0%93%D0%A0%D0%9D\", \"2\": \"%D0%9E%D0%93%D0%A0%D0%9D%D0%98%D0%9F\", \"3\": \"%D0%A1%D0%9D%D0%98%D0%9B%D0%A1\", \"4\": \"%D0%98%D0%9D%D0%9D\", \"5\": \"%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F%D0%BF%D0%BE%D1%87%D1%82%D0%B0\", \"6\": \"%D0%A1%D1%82%D1%80%D0%B0%D0%BD%D0%B0\", \"7\": \"%D0%9E%D0%B1%D0%BB%D0%B0%D1%81%D1%82%D1%8C\", \"8\": \"%D0%93%D0%BE%D1%80%D0%BE%D0%B4\", \"9\": \"%D0%9E%D1%80%D0%B3%D0%B0%D0%BD%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F\", \"10\": \"%D0%9F%D0%BE%D0%B4%D1%80%D0%B0%D0%B7%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5\", \"11\": \"%D0%9E%D0%B1%D1%89%D0%B5%D0%B5%D0%B8%D0%BC%D1%8F\", \"12\": \"%D0%90%D0%B4%D1%80%D0%B5%D1%81\", \"13\": \"%D0%94%D0%BE%D0%BB%D0%B6%D0%BD%D0%BE%D1%81%D1%82%D1%8C\", \"14\": \"%D0%98%D0%BD%D0%B8%D1%86%D0%B8%D0%B0%D0%BB%D1%8B\", \"15\": \"%D0%98%D0%BC%D1%8F\", \"16\": \"%D0%A4%D0%B0%D0%BC%D0%B8%D0%BB%D0%B8%D1%8F\" }, \"Values\":
{ \"5\": \"E12\", \"7\": \"S10\", \"8\": \"L9\", \"9\": \"08\", \"10\": \"0U7\", \"11\": \"CN6\", \"12\": \"Street5\", \"13\": \"T4\", \"14\": \"I3\", \"15\": \"G2\", \"16\": \"SN1\" } }",
      "NonceRequired": true
    }
  ],
  "InitializationToken": null
}
```

Блокировка устройства пользователя

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/mydss/lockout
Параметры	id - Глобальный идентификатор пользователя MyDssLockoutDeviceRequest - Информация об устройстве
Возвращаемое значение	MyDssKeyInfo - Обновленная информация об устройстве

Пример запроса

```
POST https://<hostname>/STS/ums/user/9fba6076-3509-4e8e-abc1-dae6d052a230/mydss/lockout HTTP/1.1
Content-Type: application/json; charset=utf-8

{
  "Kid": "73128110",
  "Lock": true
}
```

Пример ответа

HTTP/1.1 200 OK

```
{
  "Uid": "9fba6076-3509-4e8e-abc1-dae6d052a230",
  "Kid": "73128110",
  "DeviceName": "MyApple",
  "NotBefore":1574869065,
  "NotAfter":1614440265,
  "State": "NotVerified,Blocked",
  "UserName": "MdagTestUser-D547B268",
  "Profile":{"Version":1,"Keys":
  {\"1\": \"%D0%9E%D0%93%D0%A0%D0%9D\\\", \"2\": \"%D0%9E%D0%93%D0%A0%D0%9D%D0%98%D0%9F\\\", \"3\": \"%D0%A1%D0%9D%D0%98%D0%9B%D0%A1\\\", \"4\": \"%D0%98%D0%9D%D0%9D\\\", \"5\": \"%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F%D0%BF%D0%BE%D1%87%D1%82%D0%B0\\\", \"6\": \"%D0%A1%D1%82%D1%80%D0%B0%D0%BD%D0%B0\\\", \"7\": \"%D0%9E%D0%B1%D0%BB%D0%B0%D1%81%D1%82%D1%8C\\\", \"8\": \"%D0%93%D0%BE%D1%80%D0%BE%D0%B4\\\", \"9\": \"%D0%9E%D1%80%D0%B3%D0%B0%D0%BD%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F\\\", \"10\": \"%D0%9F%D0%BE%D0%B4%D1%80%D0%B0%D0%B7%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5\\\", \"11\": \"%D0%9E%D0%B1%D1%89%D0%B5%D0%B5%D0%B8%D0%BC%D1%8F\\\", \"12\": \"%D0%90%D0%B4%D1%80%D0%B5%D1%81\\\", \"13\": \"%D0%94%D0%BE%D0%BB%D0%B6%D0%BD%D0%BE%D1%81%D1%82%D1%8C\\\", \"14\": \"%D0%98%D0%BD%D0%B8%D1%86%D0%B8%D0%B0%D0%BB%D1%8B\\\", \"15\": \"%D0%98%D0%BC%D1%8F\\\", \"16\": \"%D0%A4%D0%B0%D0%BC%D0%B8%D0%BB%D0%B8%D1%8F\\\"}, \"Values\":
  {\"5\": \"%E12\\\", \"7\": \"%S10\\\", \"8\": \"%L9\\\", \"9\": \"%O8\\\", \"10\": \"%OU7\\\", \"11\": \"%CN6\\\", \"12\": \"%Street5\\\", \"13\": \"%T4\\\", \"14\": \"%I3\\\", \"15\": \"%G2\\\", \"16\": \"%SN1\\\"}}\",
  \"NonceRequired\":true
}
```

Удаление устройства пользователя

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/mydss
Параметры	id - Глобальный идентификатор пользователя MyDssDeleteKeyRequest - Информация об устройстве
Возвращаемое значение	Отсутствует

Пример запроса

```
POST https://<hostname>/STS/ums/user/9fba6076-3509-4e8e-abc1-dae6d052a230/mydss HTTP/1.1
Content-Type: application/json; charset=utf-8

{"Kid": "73128110"}
```

Пример ответа

HTTP/1.1 200 OK

Удаление всех токенов аутентификации myDSS

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	DELETE
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/mydss
Параметры	

ПАРАМЕТР	ЗНАЧЕНИЕ
Возвращаемое значение	

Пример запроса

```
DELETE https://<hostname>/STS/ums/user/9fba6076-3509-4e8e-abc1-dae6d052a230/mydss HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
```

Настройка политики подтверждения операций

Получение политики подтверждения операций пользователя

GetUserOperationPolicy

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/operationpolicy
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	List<OperationPolicy> - Политика подтверждения операций пользователя

Пример запроса

```
GET https://<hostname>/STS/ums/user/1e4ba67f-937d-48a4-9852-93bcff012486/operationpolicy HTTP/1.1
```

Пример ответа

HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 680

```
[{
  "Action": "Issue",
  "ConfirmationRequired": false
},
{
  "Action": "SignDocument",
  "ConfirmationRequired": true
},
{
  "Action": "SignDocuments",
  "ConfirmationRequired": false
},
{
  "Action": "DecryptDocument",
  "ConfirmationRequired": false
},
{
  "Action": "CreateRequest",
  "ConfirmationRequired": true
},
{
  "Action": "ChangePin",
  "ConfirmationRequired": false
},
{
  "Action": "RenewCertificate",
  "ConfirmationRequired": false
},
{
  "Action": "RevokeCertificate",
  "ConfirmationRequired": false
},
{
  "Action": "HoldCertificate",
  "ConfirmationRequired": false
},
{
  "Action": "UnholdCertificate",
  "ConfirmationRequired": false
},
{
  "Action": "DeleteCertificate",
  "ConfirmationRequired": false
},
{
  "Action": "PrivateKeyAccess",
  "ConfirmationRequired": false
}]
```

Установка политики подтверждения операции пользователя

SetUserOperationPolicy

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/operationpolicy

ПАРАМЕТР	ЗНАЧЕНИЕ
Параметры	id - Глобальный идентификатор пользователя List<DSSActions> - Список операций, для которых требуется подтверждение
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/519900cc-fb54-40ab-8b44-794d5fd7dc1b/operationpolicy HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 8

[2,
1024]
```

Пример ответа

```
HTTP/1.1 200 OK
```

Настройка политики доступа к операциям

Получение политики доступа к операциям

GetUserAccessPolicy

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/accesspolicy
Параметры	id - Глобальный идентификатор пользователя
Возвращаемое значение	List<AccessPolicy> - Политика доступа к операциям пользователя

Пример запроса

```
GET https://<hostname>/STS/ums/user/55cd94d3-3cb4-48b0-90a0-6f3ead9cb0bd/accesspolicy HTTP/1.1
```

Пример ответа

HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 445

```
[{
  "Action": "SignDocument",
  "AccessDenied": true
},
{
  "Action": "DecryptDocument",
  "AccessDenied": false
},
{
  "Action": "CreateRequest",
  "AccessDenied": true
},
{
  "Action": "DeleteCertificate",
  "AccessDenied": false
},
{
  "Action": "RenewCertificate",
  "AccessDenied": false
},
{
  "Action": "RevokeCertificate",
  "AccessDenied": false
},
{
  "Action": "HoldCertificate",
  "AccessDenied": false
},
{
  "Action": "UnholdCertificate",
  "AccessDenied": false
},
{
  "Action": "ChangePin",
  "AccessDenied": false
}]
```

Установка политики доступа к операциям

SetUserAccessPolicy

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/accesspolicy
Параметры	id - Глобальный идентификатор пользователя IList<DSSActions> - Список операций, для которых требуется подтверждение
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/55cd94d3-3cb4-48b0-90a0-6f3ead9cb0bd/accesspolicy HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 6

[2,
16]
```

Пример ответа

```
HTTP/1.1 200 OK
```

Блокировка/разблокировка пользователя

SetUserLockoutState

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/lockout
Параметры	id - Глобальный идентификатор пользователя islocked - Требуемое состояние учетной записи (True или False)
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/9fba6076-3509-4e8e-abc1-dae6d052a230/lockout?lock=True HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 2

{}
```

Пример ответа

```
HTTP/1.1 200 OK
```

Отложенная блокировка пользователя

SetUserLockoutStateEx

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/user/{id}/lockoutex
Параметры	id - Глобальный идентификатор пользователя UserLockoutRequest - Параметры блокировки
Возвращаемое значение	-

Пример запроса

```
POST https://<hostname>/STS/ums/user/9fba6076-3509-4e8e-abc1-dae6d052a230/lockoutex HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 58
```

```
{
  "LockUser": true,
  "UseLockoutDelay": true,
  "LockoutDelay": 30
}
```

Пример ответа

```
HTTP/1.1 200 OK
```

Конечная точка Users

Получение списка пользователей по заданным фильтрам

GetUsers

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/users
Параметры	UserRecordsRequest - Запрос на получение списка пользователей
Возвращаемое значение	UserRecordsResponse - Список пользователей

Пример запроса

```
POST https://<hostname>/STS/ums/users HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 88

{
  "StartPosition": 0,
  "EndPosition": -1,
  "Filters": [{
    "Column": 1,
    "Operation": 1,
    "Value": "01"
  }]
}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 436

{
  "UserInfos": [{
    "UserId": "3820731f-7478-4787-9e18-017063fdf234",
    "Login": "UmsSampleUser2",
    "PhoneNumber": "02",
    "Email": null,
    "PhoneConfirmed": true,
    "EmailConfirmed": false,
    "DisplayName": null,
    "DistinguishName": "CN=UmsSampleUser2-32ab7d64-faff-4b98-a145-87da6a2b01f8",
    "AccountLocked": false,
    "Group": "Default",
    "CreationDate": "2019-05-15T17: 13: 38.297",
    "LockoutDate": null,
    "LastLoginDate": "2019-05-15T17: 13: 38.297"
  }],
  "TotalCount": 2,
  "AffectedCount": 1
}
```

Конечная точка Authtokens

Получение списка средств аутентификации по заданным фильтрам

GetAuthnTokens

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/authntokens
Параметры	TokenRecordsRequest - Запрос на получение списка средств аутентификации
Возвращаемое значение	TokenRecordsResponse - Список средств аутентификации

Пример запроса

```
POST https://<hostname>/STS/ums/authntokens HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 36

{
  "StartPosition": 0,
  "EndPosition": -1
}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 349

{
  "TokenInfos": [{
    "Id": 161,
    "Serial": "AA000001",
    "UserName": null,
    "TokenType": "HOTP",
    "Parameters": {
      "Digits": "6",
      "Crypto": "SHA1",
      "LookAheadWindow": "10",
      "IterationNumber": "1"
    }
  },
  {
    "Id": 162,
    "Serial": "896010265977857677",
    "UserName": null,
    "TokenType": "SimAuth",
    "Parameters": {
      "profileId": "3e906aff-8639-463e-af59-07a44a96dd21"
    }
  }
],
  "TotalCount": 2,
  "AffectedCount": 2
}
```

Конечная точка Policy

Получение политики Сервиса Управления Пользователями (UMS)

GetPolicy

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/policy
Параметры	-
Возвращаемое значение	UmsPolicy - Политика Сервиса Управления Пользователями

Пример запроса

GET https://<hostname>/STS/ums/policy HTTP/1.1

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 6489

{
  "AvaliableIdentifierTypes": [
    "Login",
    "PhoneNumber",
    "Email"
  ],
  "AuthMethods": [{
    "Identifier": "http: //schemas.microsoft.com/ws/2012/09/identity/authenticationmethod/none",
    "Type": "Primary"
  },
  {
    "Identifier": "http: //dss.cryptopro.ru/identity/authenticationmethod/certificate",
    "Type": "Primary"
  },
  {
    "Identifier": "http: //dss.cryptopro.ru/identity/authenticationmethod/password",
    "Type": "Primary"
  },
  {
    "Identifier": "http: //dss.cryptopro.ru/identity/authenticationmethod/saml",
    "Type": "Primary"
  },
  {
    "Identifier": "http: //dss.cryptopro.ru/identity/authenticationmethod/otpviasms",
    "Type": "Secondary"
  },
  {
    "Identifier": "http: //dss.cryptopro.ru/identity/authenticationmethod/oath",
    "Type": "Secondary"
  },
  {
    "Identifier": "http: //dss.cryptopro.ru/identity/authenticationmethod/simauth",
    "Type": "Secondary"
  },
  {
    "Identifier": "http: //dss.cryptopro.ru/identity/authenticationmethod/otpviaemail".
```



```

"Type": "Secondary"
},
{
  "Identifier": "http: //dss.cryptopro.ru/identity/authenticationmethod/mtmo",
  "Type": "Secondary"
},
{
  "Identifier": "http: //dss.cryptopro.ru/identity/authenticationmethod/mo",
  "Type": "Secondary"
}],
"Rdns": [{
  "Id": 1,
  "Oid": "2.5.4.3",
  "DisplayName": "Общее имя",
  "StringIdentifier": "CN",
  "Order": 6,
  "MinLength": 0,
  "MaxLength": 128,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 2,
  "Oid": "2.5.4.42",
  "DisplayName": "Имя Отчество",
  "StringIdentifier": "G",
  "Order": 2,
  "MinLength": 0,
  "MaxLength": 128,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 3,
  "Oid": "2.5.4.4",
  "DisplayName": "Фамилия",
  "StringIdentifier": "SN",
  "Order": 1,
  "MinLength": 0,
  "MaxLength": 40,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 4,
  "Oid": "2.5.4.43",
  "DisplayName": "Инициалы",
  "StringIdentifier": "I",
  "Order": 3,
  "MinLength": 0,
  "MaxLength": 5,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 5,
  "Oid": "1.2.840.113549.1.9.1",
  "DisplayName": "Электронная почта",
  "StringIdentifier": "E",
  "Order": 12,
  "MinLength": 0,
  "MaxLength": 128,
  "Required": false,
  "ValueSet": []
},

```

```

{
  "Id": 6,
  "Oid": "1.2.643.3.131.1.1",
  "DisplayName": "ИНН",
  "StringIdentifier": "INN",
  "Order": 13,
  "MinLength": 12,
  "MaxLength": 12,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 7,
  "Oid": "1.2.643.100.1",
  "DisplayName": "ОГРН",
  "StringIdentifier": "OGRN",
  "Order": 16,
  "MinLength": 13,
  "MaxLength": 13,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 8,
  "Oid": "1.2.643.100.5",
  "DisplayName": "ОГРНИП",
  "StringIdentifier": "OGRNIP",
  "Order": 15,
  "MinLength": 15,
  "MaxLength": 15,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 9,
  "Oid": "1.2.643.100.3",
  "DisplayName": "СНИЛС",
  "StringIdentifier": "SNILS",
  "Order": 14,
  "MinLength": 11,
  "MaxLength": 11,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 10,
  "Oid": "2.5.4.6",
  "DisplayName": "Страна",
  "StringIdentifier": "C",
  "Order": 11,
  "MinLength": 0,
  "MaxLength": 2,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 11,
  "Oid": "2.5.4.8",
  "DisplayName": "Область",
  "StringIdentifier": "S",
  "Order": 10,
  "MinLength": 0,
  "MaxLength": 128,
  "Required": false,
  "ValueSet": []
}

```

```

},
{
  "Id": 12,
  "Oid": "2.5.4.7",
  "DisplayName": "Город",
  "StringIdentifier": "L",
  "Order": 9,
  "MinLength": 0,
  "MaxLength": 128,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 13,
  "Oid": "2.5.4.9",
  "DisplayName": "Адрес",
  "StringIdentifier": "Street",
  "Order": 5,
  "MinLength": 0,
  "MaxLength": 30,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 14,
  "Oid": "2.5.4.10",
  "DisplayName": "Организация",
  "StringIdentifier": "O",
  "Order": 8,
  "MinLength": 0,
  "MaxLength": 64,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 15,
  "Oid": "2.5.4.11",
  "DisplayName": "Подразделение",
  "StringIdentifier": "OU",
  "Order": 7,
  "MinLength": 0,
  "MaxLength": 64,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 16,
  "Oid": "2.5.4.12",
  "DisplayName": "Должность",
  "StringIdentifier": "T",
  "Order": 4,
  "MinLength": 0,
  "MaxLength": 64,
  "Required": false,
  "ValueSet": []
}],
  "RdnPolicy": {
    "Default": [{
      "Id": 1,
      "Oid": "2.5.4.3",
      "DisplayName": "Общее имя",
      "StringIdentifier": "CN",
      "Order": 6,
      "MinLength": 0,
      "MaxLength": 128,
      "Required": false,

```

```

"ValueSet": []
},
{
  "Id": 2,
  "Oid": "2.5.4.42",
  "DisplayName": "Имя Отчество",
  "StringIdentifier": "G",
  "Order": 2,
  "MinLength": 0,
  "MaxLength": 128,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 3,
  "Oid": "2.5.4.4",
  "DisplayName": "Фамилия",
  "StringIdentifier": "SN",
  "Order": 1,
  "MinLength": 0,
  "MaxLength": 40,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 4,
  "Oid": "2.5.4.43",
  "DisplayName": "Инициалы",
  "StringIdentifier": "I",
  "Order": 3,
  "MinLength": 0,
  "MaxLength": 5,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 5,
  "Oid": "1.2.840.113549.1.9.1",
  "DisplayName": "Электронная почта",
  "StringIdentifier": "E",
  "Order": 12,
  "MinLength": 0,
  "MaxLength": 128,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 6,
  "Oid": "1.2.643.3.131.1.1",
  "DisplayName": "ИНН",
  "StringIdentifier": "INN",
  "Order": 13,
  "MinLength": 12,
  "MaxLength": 12,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 7,
  "Oid": "1.2.643.100.1",
  "DisplayName": "ОГРН",
  "StringIdentifier": "OGRN",
  "Order": 16,
  "MinLength": 13,
  "MaxLength": 13,

```

```

"Required": false,
"ValueSet": []
},
{
  "Id": 8,
  "Oid": "1.2.643.100.5",
  "DisplayName": "ОГРНИП",
  "StringIdentifier": "OGRNIP",
  "Order": 15,
  "MinLength": 15,
  "MaxLength": 15,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 9,
  "Oid": "1.2.643.100.3",
  "DisplayName": "СНИЛС",
  "StringIdentifier": "SNILS",
  "Order": 14,
  "MinLength": 11,
  "MaxLength": 11,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 10,
  "Oid": "2.5.4.6",
  "DisplayName": "Страна",
  "StringIdentifier": "C",
  "Order": 11,
  "MinLength": 0,
  "MaxLength": 2,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 11,
  "Oid": "2.5.4.8",
  "DisplayName": "Область",
  "StringIdentifier": "S",
  "Order": 10,
  "MinLength": 0,
  "MaxLength": 128,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 12,
  "Oid": "2.5.4.7",
  "DisplayName": "Город",
  "StringIdentifier": "L",
  "Order": 9,
  "MinLength": 0,
  "MaxLength": 128,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 13,
  "Oid": "2.5.4.9",
  "DisplayName": "Адрес",
  "StringIdentifier": "Street",
  "Order": 5,
  "MinLength": 0,
  "MaxLength": 30.

```

```

    "Required": false,
    "ValueSet": []
  },
  {
    "Id": 14,
    "Oid": "2.5.4.10",
    "DisplayName": "Организация",
    "StringIdentifier": "O",
    "Order": 8,
    "MinLength": 0,
    "MaxLength": 64,
    "Required": false,
    "ValueSet": []
  },
  {
    "Id": 15,
    "Oid": "2.5.4.11",
    "DisplayName": "Подразделение",
    "StringIdentifier": "OU",
    "Order": 7,
    "MinLength": 0,
    "MaxLength": 64,
    "Required": false,
    "ValueSet": []
  },
  {
    "Id": 16,
    "Oid": "2.5.4.12",
    "DisplayName": "Должность",
    "StringIdentifier": "T",
    "Order": 4,
    "MinLength": 0,
    "MaxLength": 64,
    "Required": false,
    "ValueSet": []
  }
],
"AllowUserRegistration": true,
"IdentityProviders": [{
  "Description": null,
  "IssuerName": "realsts",
  "DisplayName": null
},
{
  "Description": null,
  "IssuerName": "ADFS",
  "DisplayName": null
},
{
  "Description": null,
  "IssuerName": "SampleSts",
  "DisplayName": null
}],
"Groups": [{
  "IdentityProviderName": "realsts",
  "GroupList": ["Default",
  "SampleGroup"]
}],
"CryptoProviders": [],
"MobileAuthSettings": null,
"AirKeyAuthSettings": null
}

```

Конечная точка GroupPolicy

Получение политики группы пользователей на Центре Идентификации

GetGroupPolicy

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://<адрес сервера>/<имя приложения ЦИ>/ums/groupPolicy/{groupName}
Параметры	groupName - Имя группы, политику которой необходимо получить
Возвращаемое значение	GroupPolicy - Политика группы

Пример запроса

```
GET https://<hostname>/STS/ums/groupPolicy/Default HTTP/1.1
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 2538

{
  "AllowOpcPolicyChange": true,
  "Rdns": [{
    "Id": 1,
    "Oid": "2.5.4.3",
    "DisplayName": "Общее имя",
    "StringIdentifier": "CN",
    "Order": 6,
    "MinLength": 0,
    "MaxLength": 128,
    "Required": false,
    "ValueSet": []
  },
  {
    "Id": 2,
    "Oid": "2.5.4.42",
    "DisplayName": "Имя Отчество",
    "StringIdentifier": "G",
    "Order": 2,
    "MinLength": 0,
    "MaxLength": 128,
    "Required": false,
    "ValueSet": []
  },
  {
    "Id": 3,
    "Oid": "2.5.4.4",
    "DisplayName": "Фамилия",
    "StringIdentifier": "SN",
    "Order": 1,
    "MinLength": 0,
    "MaxLength": 40,
    "Required": false,
    "ValueSet": []
  },
  {
```

```

{
  "Id": 4,
  "Oid": "2.5.4.43",
  "DisplayName": "Инициалы",
  "StringIdentifier": "I",
  "Order": 3,
  "MinLength": 0,
  "MaxLength": 5,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 5,
  "Oid": "1.2.840.113549.1.9.1",
  "DisplayName": "Электронная почта",
  "StringIdentifier": "E",
  "Order": 12,
  "MinLength": 0,
  "MaxLength": 128,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 6,
  "Oid": "1.2.643.3.131.1.1",
  "DisplayName": "ИНН",
  "StringIdentifier": "INN",
  "Order": 13,
  "MinLength": 12,
  "MaxLength": 12,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 7,
  "Oid": "1.2.643.100.1",
  "DisplayName": "ОГРН",
  "StringIdentifier": "OGRN",
  "Order": 16,
  "MinLength": 13,
  "MaxLength": 13,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 8,
  "Oid": "1.2.643.100.5",
  "DisplayName": "ОГРНИП",
  "StringIdentifier": "OGRNIP",
  "Order": 15,
  "MinLength": 15,
  "MaxLength": 15,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 9,
  "Oid": "1.2.643.100.3",
  "DisplayName": "СНИЛС",
  "StringIdentifier": "SNILS",
  "Order": 14,
  "MinLength": 11,
  "MaxLength": 11,
  "Required": false,
  "ValueSet": []
},

```



```

{
  "Id": 10,
  "Oid": "2.5.4.6",
  "DisplayName": "Страна",
  "StringIdentifier": "C",
  "Order": 11,
  "MinLength": 0,
  "MaxLength": 2,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 11,
  "Oid": "2.5.4.8",
  "DisplayName": "Область",
  "StringIdentifier": "S",
  "Order": 10,
  "MinLength": 0,
  "MaxLength": 128,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 12,
  "Oid": "2.5.4.7",
  "DisplayName": "Город",
  "StringIdentifier": "L",
  "Order": 9,
  "MinLength": 0,
  "MaxLength": 128,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 13,
  "Oid": "2.5.4.9",
  "DisplayName": "Адрес",
  "StringIdentifier": "Street",
  "Order": 5,
  "MinLength": 0,
  "MaxLength": 30,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 14,
  "Oid": "2.5.4.10",
  "DisplayName": "Организация",
  "StringIdentifier": "O",
  "Order": 8,
  "MinLength": 0,
  "MaxLength": 64,
  "Required": false,
  "ValueSet": []
},
{
  "Id": 15,
  "Oid": "2.5.4.11",
  "DisplayName": "Подразделение",
  "StringIdentifier": "OU",
  "Order": 7,
  "MinLength": 0,
  "MaxLength": 64,
  "Required": false,
  "ValueSet": []
}

```

```
f,
{
  "Id": 16,
  "Oid": "2.5.4.12",
  "DisplayName": "Должность",
  "StringIdentifier": "T",
  "Order": 4,
  "MinLength": 0,
  "MaxLength": 64,
  "Required": false,
  "ValueSet": []
}]
}
```

Типы данных

- Политика доступа к операциям (Access Policy)
- Информация о существующем токене аутентификации MobileAuth (AssignMobileAuthTokenInfo)
- Описание шага схемы аутентификации пользователя (AuthenticationInfo)
- Список доступных методов аутентификации (AuthnMethodDescription)
- Список профилей криптопровайдеров (CryptoProviderInfo)
- Список операций, для которых требуется подтверждение (DSSActions)
- Информация о средстве аутентификации (DssTokenInfo)
- Информация о пользователе (DssUserInfo)
- Коды ошибок (ErrorCodes)
- Запрос на создание внешнего логина пользователя (ExternalLoginInfo)
- Политика группы пользователей (GroupPolicy)
- Типы идентификаторов пользователя (IdentifierType)
- Список групп на Центре Идентификации (IdentityGroupInfo)
- Список доверенных Центров Идентификации (IdentityProviderInfo)
- Тип сообщения, связанного с жизненным циклом апплета (LifecycleMessageType)
- Информация о созданном ключе аутентификации MobileAuth (MobileAuthCreateInfoEx)
- Описание настроек аутентификации MobileAuth (MobileAuthSettings)
- Информация об аутентификации MobileAuth (MobileAuthTokenInfo)
- Информация о созданном ключе аутентификации MobileAuth (MobileAuthUpdateInfoEx)
- Информация об OATH-токене (OATHTokenInfo)
- Политика подтверждения операций (OperationPolicy)
- Ответ на запрос к апплету на SIM-карте (OperationResults)
- Информация о компоненте различительного имени пользователя (RdnInfo)
- Результат отправки запроса на смену ключа аутентификации (SimAuthChangeKeyRequestResult)
- Результат выполнения запроса к апплету на SIM-карте (SimAuthLifecycleMessageStatusRest)
- Информация о SIM-карте пользователя (SimAuthTokenInfo)
- Запрос на получение списка средств аутентификации (TokenRecordsRequest)
- Ответ на запрос о получении отфильтрованного списка средств аутентификации (TokenRecordsResponse)
- Политика Сервиса Управления Пользователями (UmsPolicy)
- Контактная информация пользователя (UserContactInfo)
- Информация о пользователе (UserEmailInfo)
- Информация о созданном ключе аутентификации MobileAuth (UserMobileAuthInfo)
- Информация об OTP-токене пользователя (UserOtpTokenInfo)
- Информация о номере телефона пользователя (UserPhoneInfo)
- Свойства учетной записи пользователя (UserProperty)
- Запрос на аутентификационную информацию пользователя (UserRawAuthDataRequest)
- Запрос на получение списка пользователей (UserRecordsRequest)
- Ответ на запрос о получении отфильтрованного списка пользователей (UserRecordsResponse)
- Информация об аутентификации при помощи апплета на SIM-карте

Политика доступа к операциям (Access Policy)

Политику доступа к операциям Сервиса Управления Пользователями можно получить, вызвав метод [user/{id}/accesspolicy](#).

ПОЛЕ	ТИП	ОПИСАНИЕ
Action	DSSActions	Операция, доступ к которой должен быть настроен
AccessDenied	Bool	Флаг, показывающий, запрещен ли доступ к операции

Информация о существующем токене аутентификации MobileAuth (AssignMobileAuthTokenInfo)

Информация о существующем токене аутентификации MobileAuth передается в структуре AssignMobileAuthTokenInfo в метод \AssignUserMobileAuthToken.

ПОЛЕ	ТИП	ОПИСАНИЕ
UserContactInfo	String	Контактная информация пользователя
UserContactInfoType	String	Тип контактной информации пользователя
NeedXmlKeyInfo	Bool	Флаг, определяющий, нужен ли ключ аутентификации в формате XML
KeyInfoPinCode	String	ПИН-код для расшифрования ключа
DelayedActivation	Bool	Флаг, определяющий требуется ли немедленная активация
Serial	String	Серийный номер токена аутентификации MobileAuth

Описание шага схемы аутентификации пользователя (AuthenticationInfo)

Структура AuthenticationInfo содержит описание шага схемы аутентификации пользователя.

ПОЛЕ	ТИП	ОПИСАНИЕ
MethodUri	String	Идентификатор метода аутентификации
Level	Int	Уровень, к которому привязан метод аутентификации

Список доступных методов аутентификации (AuthnMethodDescription)

Список доступных методов аутентификации AuthnMethodDescription.

ПОЛЕ	ТИП	ОПИСАНИЕ
Identifier	String	Идентификатор метода аутентификации
Type	AuthnLevel	Тип метода аутентификации

Типы методов аутентификации (AuthnLevel)

ТИП	ЗНАЧЕНИЕ	ОПИСАНИЕ
Primary	1	Первичная аутентификация
Secondary	2	Вторичная аутентификация

Список профилей криптопровайдеров (CryptoProviderInfo)

Список профилей криптопровайдеров CryptoProviderInfo.

ПОЛЕ	ТИП	ОПИСАНИЕ
Id	String	Идентификатор профиля криптопровайдера
Name	String	Имя профиля криптопровайдера
Description	String	Описание профиля криптопровайдера
Type	CryptoProviderProfileType	Тип профиля криптопровайдера

Типы профилей криптопровайдеров (CryptoProviderProfileType)

ТИП	ЗНАЧЕНИЕ	ОПИСАНИЕ
GOST	1	Алгоритмы семейства ГОСТ
RSA_AES	2	Алгоритмы RSA+AES

Список операций, для которых требуется подтверждение (DSSActions)

Список операций, для которых требуется подтверждение, передается в перечислении DSSActions в группы методов "Политика доступа к операциям" и "Политика подтверждений операций".

ОПЕРАЦИЯ	КОД ОПЕРАЦИИ	ОПИСАНИЕ
Issue	1	Выпуск маркера (Вход)
SignDocument	2	Подпись документа
SignDocuments	4	Пакетная подпись документа
DecryptDocument	8	Расшифрование документа
CreateRequest	16	Создание запроса на сертификат
ChangePin	32	Смена ПИН-кода для доступа к закрытому ключу
RenewCertificate	64	Перевыпуск сертификата
RevokeCertificate	128	Отзыв сертификата
HoldCertificate	256	Приостановление сертификата
UnholdCertificate	512	Возобновление сертификата
DeleteCertificate	1024	Удаление сертификата
PrivateKeyAccess	2048	Доступ к закрытому ключу

Информация о средстве аутентификации (DssTokenInfo)

Структура DssTokenInfo содержит информацию о средстве аутентификации.

ПОЛЕ	ТИП	ОПИСАНИЕ
Id	Long	Идентификатор средства аутентификации (токена)
Serial	String	Серийный номер токена
UserName	String	Логин пользователя, которому назначено средство аутентификации
TokenType	String	Тип средства аутентификации
Parameters	Dictionary<string, string>	Параметры средства аутентификации

Информация о пользователе (DssUserInfo)

Структура DssUserInfo содержит информацию о пользователе.

ПОЛЕ	ТИП	ОПИСАНИЕ
UserId	String	Глобальный идентификатор пользователя
Login	String	Логин пользователя
PhoneNumber	String	Номер мобильного телефона пользователя
Email	String	Адрес электронной почты пользователя
PhoneConfirmed	Bool	Флаг, показывающий, подтвержден ли мобильный телефон
EmailConfirmed	Bool	Флаг, показывающий, подтвержден ли адрес электронной почты
DisplayName	String	Отображаемое имя пользователя
DistinguishName	String	Различительное имя пользователя
AccountLocked	Bool	Флаг, показывающий, заблокирован ли аккаунт пользователя
Group	String	Группа, в которой состоит пользователь
CreationDate	DateTime	Дата создания учетной записи пользователя в формате уууу-мм-ддТ00:00:00
LockoutDate	DateTime	Дата снятия блокировки учетной записи пользователя в формате уууу-мм-ддТ00:00:00
LastLoginDate	DateTime	Дата последней аутентификации пользователя в формате уууу-мм-ддТ00:00:00

Коды ошибок (ErrorCodes)

код	ОПИСАНИЕ
access_denied	
user_not_found	
token_not_found	
transaction_not_found	
wrong_operation	
internal_error	
invalid_identifiers	
invalid_issuer	
invalid_group	
invalid_unique_identifiers	
invalid_login	
invalid_phone	
invalid_email	
contact_confirmation_required	
invalid_confirmation	
invalid_rdns	
invalid_serial	
invalid_sync_otp	
invalid_token	
invalid_profile	
invalid_uri	
invalid_authn_method	
invalid_authentication_scheme	
authn_method_not_confirmed	

КОД	ОПИСАНИЕ
invalid_params	
invalid_contact_info	
invalid_activation_code	
invalid_user_property	
invalid_filter	

Запрос на создание внешнего логина пользователя (ExternalLoginInfo)

Запрос на создание внешнего логина пользователя передается в структуре ExternalLoginInfo в метод [/SetExternalLogin](#).

ПОЛЕ	ТИП	ОПИСАНИЕ
Login	String	Значение логина
IssuerName	String	Имя домена

Политика группы пользователей (GroupPolicy)

Политику группы пользователей можно получить, вызвав метод [user/{id}/accesspolicy](#).

ПОЛЕ	ТИП	ОПИСАНИЕ
AllowOpсPolicyChange	Bool	Флаг, показывающий, разрешено ли оператору изменять настройки подтверждения операций пользователей.
Rdns	IList< RdnInfo >	Список компонентов различительного имени пользователя

Типы идентификаторов пользователя (IdentifierType)

Типы идентификаторов пользователя передаются в перечислении IdentifierType в методы [/RegisterUser](#), [/GetUser](#).

тип	описание
Login	Логин пользователя
PhoneNumber	Номер мобильного телефона
Email	Адрес электронной почты

Список групп на Центре Идентификации (IdentityGroupInfo)

Список групп на Центре Идентификации IdentityGroupInfo.

ПОЛЕ	ТИП	ОПИСАНИЕ
IdentityProviderName	String	Имя доверенного ЦИ (аналогично IssuerName в IdentityProviderInfo)
GroupList	IList<string>	Список групп на доверенном ЦИ

Список доверенных Центров Идентификации (IdentityProviderInfo)

Список доверенных Центров Идентификации IdentityProviderInfo.

ПОЛЕ	ТИП	ОПИСАНИЕ
Description	String	Описание доверенного ЦИ
IssuerName	String	Имя доверенного ЦИ
DisplayName	String	Отображаемое имя ЦИ

Тип сообщения, связанного с жизненным циклом апплета (LifecycleMessageType)

Тип сообщения, связанного с жизненным циклом апплета, передается в структуре LifecycleMessageType в метод [/SetUserSimAuthToken](#).

ИМЯ СООБЩЕНИЯ	КОД СООБЩЕНИЯ	ОПИСАНИЕ
Activate	0xAA	Запрос на активацию апплета
GetStatus	0xAB	Запрос на получение статуса апплета
ChangePin	0xAC	Запрос на изменение ПИН-кода

Информация о созданном ключе аутентификации MobileAuth (MobileAuthCreateInfoEx)

Структура MobileAuthCreateInfoEx содержит информацию о создаваемом ключе аутентификации MobileAuth. Поле QR-код содержит QR-код, который необходимо отсканировать на устройстве пользователя для завершения настройки мобильного приложения myDSS.

ПОЛЕ	ТИП	ОПИСАНИЕ
XmlKeyInfo	String	Ключ аутентификации в формате XML
ExternalUserId	String	Идентификатор пользователя на внешнем сервисе аутентификации
QrCode	String	QR-код с ключом аутентификации MobileAuth
KeyExpirationTime	DateTime	Дата и время истечения ключа аутентификации в формате уууу-mm-ddT00:00:00

Описание настроек аутентификации MobileAuth (MobileAuthSettings)

Описание настроек аутентификации MobileAuth.

ПОЛЕ	ТИП	ОПИСАНИЕ
DeviceFingerprintRequired	Bool	Флаг, показывающий, необходим ли отпечаток устройства
KeyInfoDivideRequired	Bool	Флаг, показывающий, необходимо ли разделение ключа

Информация об аутентификации MobileAuth (MobileAuthTokenInfo)

Информация об аутентификации MobileAuth передается в структуре MobileAuthTokenInfo в методы группы "[Метод аутентификации MobileAuth](#)".

ПОЛЕ	ТИП	ОПИСАНИЕ
UserContactInfo	String	Контактная информация пользователя
UserContactInfoType	String	Тип контактной информации пользователя
NeedXmlKeyInfo	Bool	Флаг, определяющий, нужен ли ключ аутентификации в формате XML
KeyInfoPinCode	String	ПИН-код для расшифрования ключа
DelayedActivation	Bool	Флаг, определяющий требуется ли немедленная активация

Информация о созданном ключе аутентификации MobileAuth (MobileAuthUpdateInfoEx)

Структура MobileAuthUpdateInfoEx содержит информацию об обновляемом ключе аутентификации MobileAuth.

ПОЛЕ	ТИП	ОПИСАНИЕ
XmlKeyInfo	String	Ключ аутентификации в формате XML
QrCode	String	QR-код с ключом аутентификации MobileAuth
KeyExpirationTime	DateTime	Дата и время истечения ключа аутентификации в формате уууу-мм-ддТ00:00:00

Информация об OATH-токене (OATHTokenInfo)

Информация об OATH-токене передается в структуре OATHTokenInfo в метод [/SetUserOtpToken](#).

ПОЛЕ	ТИП	ОПИСАНИЕ
Serial	String	Серийныйномер токена
FirstOtp	String	Первый пароль для синхронизации
SecondOtp	String	Второй пароль для синхронизации

Политика подтверждения операций (OperationPolicy)

Политику подтверждения операций Сервиса Управления Пользователями можно получить, вызвав метод [user/{id}/operationpolicy](#).

поле	тип	описание
Action	DSSActions	Операция, для которой требуется подтверждение
ConfirmationRequired	Bool	Флаг, показывающий, требуется ли подтверждение операций

Ответ на запрос к апплету на SIM-карте (OperationResults)

Перечисление OperationResults содержит результат отправки запроса на получение статуса апплета на SIM-карте.

ИМЯ	КОД	ОПИСАНИЕ
OperationInProgress	0x00	Операция выполняется (результат будет отправлен в исходящем сообщении).
AppletActivated	0x01	Апплет активирован.
AppletInactive	0x02	Апплет не активирован.
AppletBlocked	0x03	Апплет заблокирован (превышено количество попыток ввода кода активации/PIN).
InvalidMessage	0x04	Неправильный формат сообщения.
AppletIsBusy	0x05	Апплет занят.
AppletAlreadyActivated	0x06	Апплет уже активирован.
AppletActivated2	0x07	Апплет активирован (ключ 2).
OperationConfirmed	0x10	Операция одобрена пользователем.
OperationCancelled	0x11	Операция отменена пользователем.
OperationTimeOut	0x12	Операция не выполнена из-за превышения времени ожидания.
DstkError	0x13	Ошибка обработки DSTK.
PinNotChanged	0x14	ПИН-код не изменён.
PinChanged	0x15	ПИН-код изменён.
TextLengthExceeded	0x17	Превышена длина текста.
KeyAlreadyChanged	0x18	Ключ уже был обновлён.
InvalidKey	0x19	Неверный ключ.
KeyChangeSucceed	0x20	Ключ успешно сменён.

Информация о компоненте различительного имени пользователя (RdnInfo)

Структура RdnInfo содержит информацию о компоненте различительного имени пользователя.

ПОЛЕ	ТИП	ОПИСАНИЕ
Id	Int	Идентификатор компонента различительного имени пользователя
Oid	String	Объектный идентификатор компонента различительного имени пользователя
DisplayName	String	Отображаемое имя компонента различительного имени пользователя
StringIdentifier	String	Строковый идентификатор компонента различительного имени пользователя
Order	Int	Порядок следования компонента различительного имени пользователя
MinLength	Int	Минимальная длина компонента различительного имени пользователя
MaxLength	Int	Максимальная длина компонента различительного имени пользователя
Required	Bool	Флаг, показывающий, является ли компонент различительного имени пользователя обязательным
ValueSet	List<string>	Набор возможных значений компонента различительного имени пользователя. Если набор пуст - компонент может принимать любое значение

Результат отправки запроса на смену ключа аутентификации (SimAuthChangeKeyRequestResult)

Структура SimAuthChangeKeyRequestResult содержит результат отправки запроса на смену ключа аутентификации.

ПОЛЕ	ТИП	ОПИСАНИЕ
TransactionId	String	Номер телефона пользователя
ActivationCode2	List<string>	Код активации второго ключа в виде набора чисел

Результат выполнения запроса к апплету на SIM-карте (SimAuthLifecycleMessageStatusRest)

Структура SimAuthLifecycleMessageStatusRest содержит результат выполнения запроса к апплету на SIM-карте.

ПОЛЕ	ТИП	ОПИСАНИЕ
AppletResult	Byte	Результат отправки запроса на получение статуса апплета на SIM-карте
IsCompleted	Bool	флаг, показывающий завершена ли обработка запроса апплетом (например, в случае получения OperationInProgress в поле AppletResult)
MessageType	LifecycleMessageType	Тип запроса к апплету
BackwardCompatibility	SimAuthLifecycleMessageStatus	Преобразование для обратной совместимости с WCF

Результат выполнения запроса к апплету на SIM-карте (SimAuthLifecycleMessageStatus)

Структура SimAuthLifecycleMessageStatus содержит результат выполнения запроса к апплету на SIM-карте в формате, обеспечивающем обратную совместимость с WCF.

ПОЛЕ	ТИП	ОПИСАНИЕ
AppletResult	Byte	Результат отправки запроса на получение статуса апплета на SIM-карте
IsCompleted	Bool	флаг, показывающий завершена ли обработка запроса апплетом (например, в случае получения OperationInProgress в поле AppletResult)

Информация о SIM-карте пользователя (SimAuthTokenInfo)

Информация о SIM-карте пользователя передается в структуре SimAuthTokenInfo в метод [/simauth](#).

поле	тип	описание
Icclid	String	Идентификатор SIM-карты
ProfileId	String	Идентификатор профиля криптопровайдера
PhoneNumber	String	Номер телефона пользователя

Запрос на получение списка средств аутентификации (TokenRecordsRequest)

Запрос на получение списка средств аутентификации передается в структуре TokenRecordsRequest в метод [/GetAuthnTokens](#).

ПОЛЕ	ТИП	ОПИСАНИЕ
StartPosition	int	Стартовая позиция выборки
EndPosition	int	Конечная позиция выборки
Filters	IList<TokenFilter>	Список фильтров

Список фильтров по средствам аутентификации

ФИЛЬТР	ОПИСАНИЕ	ДОСТУПНЫЕ ЗНАЧЕНИЯ
Column	Свойство средства аутентификации, по которому осуществляется выборка	Id - (0) Идентификатор токена Serial - (1) Серийный номер токена UserId - (2) Идентификатор пользователя Alias - (3) Псевдоним анонимного устройства TokenType - (4) Тип токена TokenType - (4) Тип токена LicenseSerialNumber (5) - Лицензия на метод аутентификации
Operation	Тип применяемой фильтрации	0 - Равно 1 - Неравно 2 - Содержит 3 - Больше чем 4 - Меньше чем
Value	Значение фильтра	-

Ответ на запрос о получении отфильтрованного списка средств аутентификации (TokenRecordsResponse)

Ответ на запрос о получении отфильтрованного списка пользователей передается в структуре TokenRecordsResponse в метод [/users](#).

поле	тип	описание
TokenInfos	IList< DssTokenInfo >	Информация о средствах аутентификации
TotalCount	Int	Общее количество средств аутентификации в БД
AffectedCount	Int	Количество средств аутентификации в списке, соответствующее фильтру

Политика Сервиса Управления Пользователями (UmsPolicy)

Политику Сервиса Управления Пользователями можно получить, вызвав метод [/policy](#).

ПОЛЕ	ТИП	ОПИСАНИЕ
AvaiableIdentifierTypes	ICollection<IdentifierType>	Список доступных типов идентификаторов пользователя
AuthMethods	ICollection<AuthnMethodDescription>	Список доступных методов аутентификации
Rdns	ICollection<RdnInfo>	Список компонентов различительного имени пользователя для группы по умолчанию
RdnPolicy	IDictionary<string, ICollection<RdnInfo>>	Политика компонентов различительного имени пользователя по группам
AllowUserRegistration	Bool	Флаг, показывающий разрешена ли саморегистрация пользователей
IdentityProviders	ICollection<IdentityProviderInfo>	Список доверенных Центров Идентификации
Groups	ICollection<IdentityGroupInfo>	Список групп на Центре Идентификации
CryptoProviders	ICollection<CryptoProviderInfo>	Список профилей криптопровайдеров
MobileAuthSettings	MobileAuthSettings	Описание настроек аутентификации MobileAuth

Контактная информация пользователя (UserContactInfo)

Структура UserContactInfo содержит контактную информацию пользователя.

поле	тип	описание
Type	ContactInfoType	Тип контактной информации пользователя (Телефон или электронная почта)
Contact	String	Контакт пользователя
Confirmed	Bool	Флаг, показывающий, подтвержден ли контакт пользователя
Primary	Bool	Флаг, показывающий, может ли быть использован контакт пользователя для первичной аутентификации
Notification	Bool	Флаг, показывающий, может ли быть использован контакт пользователя для рассылки уведомлений
Usages	List<ContactInfoUsage>	

Перечисление типов контактной информации пользователя (ContactInfoType)

Данное перечисление содержит типы контактной информации пользователя.

тип	описание
EmailAddress	Адрес электронной почты
PhoneNumber	Номер телефона

ContactInfoUsage

поле	тип	описание
Type	String	Тип метода аутентификации, для которого используется контактная информация
Title	String	Заголовок описания использования контактной информации
Description	String	Описание использования контактной информации

Информация о пользователе (UserEmailInfo)

Структура UserEmailInfo содержит информацию об адресе электронной почты пользователя.

ПОЛЕ	ТИП	ОПИСАНИЕ
Email	String	Адрес электронной почты пользователя
Confirmed	Bool	Флаг, показывающий, является ли адрес электронной почты пользователя подтвержденным

Информация о созданном ключе аутентификации MobileAuth (UserMobileAuthInfo)

Структура UserMobileAuthInfo содержит информацию о ключе аутентификации MobileAuth.

ПОЛЕ	ТИП	ОПИСАНИЕ
UserId	String	Идентификатор пользователя на внешнем сервисе аутентификации
KeyExpirationTime	DateTime	Время истечения ключа аутентификации MobileAuth
IsInstalled	Bool	Флаг, показывающий, установлен ли ключ на мобильное устройство пользователя
InstallDate	DateTime	Время установки ключа на мобильное устройство в формате уууу-мм-ддТ00:00:00
HasDelayedKey	Bool	Флаг, показывающий наличие отложенного (неактивного) ключа аутентификации MobileAuth
KeyVersion	Int	Версия ключа аутентификации MobileAuth

Информация об OTP-токене пользователя (UserOtpTokenInfo)

Структура UserOtpTokenInfo содержит информацию об OTP-токене пользователя.

поле	тип	описание
Serial	String	Серийный номер OTP-токена
Type	String	Тип OTP-токена

Информация о номере телефона пользователя (UserPhoneInfo)

Структура UserPhoneInfo содержит информацию о номере телефона пользователя.

ПОЛЕ	ТИП	ОПИСАНИЕ
PhoneNumber	String	Номер телефона пользователя
Confirmed	Bool	Флаг, показывающий, является ли номер телефона пользователя подтвержденным

Свойства учетной записи пользователя (UserProperty)

Свойства учетной записи пользователя передаются в структуре ExternalLoginInfo в метод [/SetUserProperty](#) и изменяются в методе [/SetUserProperty](#).

SetUserLockoutState

ПОЛЕ	ТИП	ОПИСАНИЕ
PropertyType	UserPropertyType	Тип свойства
Value	DataMember	Значение свойства

[Перечисление типов свойств пользователя \(UserPropertyType\)](#)

СВОЙСТВО	ОПИСАНИЕ
DisplayName	Отображаемое имя пользователя
AccountLockedState	Состояние блокировки аккаунта

Запрос на аутентификационную информацию пользователя (UserRawAuthDataRequest)

Запрос на аутентификационную информацию пользователя передается в структуре UserRawAuthDataRequest в метод [/GetUserRawAuthenticationDataAsync](#).

ПОЛЕ	ТИП	ОПИСАНИЕ
UserId	String	Глобальный идентификатор пользователя
AuthnMethods	String[]	Набор идентификаторов интересующих методов аутентификации
Format	UserRawAuthDataFormats	Формат отображения данных

Форматы отображения подробных аутентификационных данных пользователя (UserRawAuthDataFormats)

ЗНАЧЕНИЕ	ОПИСАНИЕ
Json	Данные в формате Json
Xml	Данные в формате XML
FormattedHtml	Данные, к которым применилось XSL-преобразование, в формате HTML
FormattedPdf	Данные, к которым применилось XSL-преобразование, в формате PDF

Запрос на получение списка пользователей (UserRecordsRequest)

Запрос на получение списка пользователей передается в структуре UserRecordsRequest в метод [/GetUsers](#).

ПОЛЕ	ТИП	ОПИСАНИЕ
StartPosition	int	Стартовая позиция выборки
EndPosition	int	Конечная позиция выборки
Filters	IList<UserFilter>	Список фильтров

Список фильтров по пользователям

ФИЛЬТР	ОПИСАНИЕ	ДОСТУПНЫЕ ЗНАЧЕНИЯ
Column	Свойство пользователя, по которому осуществляется выборка	Login - Логин пользователя PhoneNumber - Номер телефона пользователя Email - Адрес электронной почты пользователя CreateDate - Дата создания учетной записи пользователя GroupId - Идентификатор группы пользователя
Operation	Тип применяемой фильтрации	0 - Равно 1 - Неравно 2 - Содержит 3 - Больше чем 4 - Меньше чем
Value	Значение фильтра	-

Ответ на запрос о получении отфильтрованного списка пользователей (UserRecordsResponse)

Ответ на запрос о получении отфильтрованного списка пользователей передается в структуре UserRecordsResponse в метод [/users](#).

поле	тип	описание
UserInfos	IList< DssUserInfo >	Информация о пользователях
TotalCount	Int	Общее количество пользователей в БД
AffectedCount	Int	Количество пользователей в списке, соответствующее фильтру

Информация об аутентификации при помощи апплета на SIM-карте (UserSimAuthInfo)

Структура UserSimAuthInfo содержит информацию об аутентификации при помощи апплета на SIM-карте.

ПОЛЕ	ТИП	ОПИСАНИЕ
Icldd	String	Идентификатор SIM-карты
ProfileId	String	Идентификатор партии SIM-карт
PhoneNumber	String	Номер телефона пользователя
ActivationCode	String	Код активации апплета
LastKnownStatus	Byte	Последний известный статус апплета

Запрос на блокировку/разблокировку пользователя (UserLockoutRequest)

Структура UserLockoutRequest содержит параметры запроса на установление состояния блокировки учетной записи пользователя.

ПОЛЕ	ТИП	ОПИСАНИЕ
LockUser	Bool	<code>true</code> , если требуется заблокировать пользователя, <code>false</code> - если требуется разблокировать.
UseLockoutDelay	Bool	Требуется ли использовать отложенную блокировку пользователя.
LockoutDelay	Int?	Количество дней, по прошествию которых учётная запись пользователя должна считаться заблокированной. Если не указано никакое значение для данного параметра, но требуется отложенная блокировка, то учётная запись будет считаться заблокированной по прошествию указанных в конфигурации ЦИ количества дней.

Состояния мобильного устройства

В данном разделе описаны возможные статусы мобильного устройства.

состояние	описание
Created	Мобильное Устройство (МУ) создано.
Installed	Векторы аутентификации загружены в МУ.
NotVerified	Учётная запись пользователя требует подтверждения.
Active	Мобильное Устройство готово к использованию.
Expired	Срок действия векторы аутентификации истёк.
Inactive	МУ заблокировано из-за недоступности метода аутентификации для пользователя.
NotConfirmed	Новое МУ ожидает подтверждения добавления на другом устройстве.
Rejected	Добавление нового МУ отклонено.
Blocked	Устройство заблокировано.

Информация о добавляемом устройстве пользователя (MyDssAssignKeyRequest)

Сведения о присоединяемом устройстве пользователя. Используется в методах групп [Аутентификация myDSS 2.0](#)

поле	тип	описание
kid	String	Идентификатор устройства пользователя

Информация о зарегистрированном устройстве myDSS (MyDssCreatedKeyInfo)

Информация о зарегистрированном устройстве myDSS Используется в методах групп [Аутентификация myDSS 2.0](#)

ПОЛЕ	ТИП	ОПИСАНИЕ
kid	String	Идентификатор устройства пользователя
uid	String	Идентификатор (guid) пользователя DSS.
DeviceName	String	Имя устройства, на котором сохранён ключ.
DeviceParameters	IDictionary<string, string>	Параметры устройства, на котором сохранён ключ.
NotBefore	long	Срок начала действия ключа.
NotAfter	long	Срок окончания действия ключа.
State	string	Статус ключа (DeviceStates).
UserName	string	Имя пользователя (он же external user id, он же логин).
Profile	string	рофиль пользователя.
NonceRequired	bool	Требование подтверждения ключа с помощью Nonce
EncryptedBlobs	byte[]	Зашифрованные ключи аутентификации.
PublicKey	byte[]	Открытый ключ сервера.
Seed	byte[]	Зерно для инициализации ДСЧ.
ActivationRequired	bool	Требуется ли активация ключа с помощью кода активации.
ServiceUrl	string	Адрес сервиса.
Alias	string	Псевдоним устройства

QR-код KInit (MyDssCreatedKeyInfoEx)

Данные для инициализации устройства пользователя Используется в методах групп [Аутентификация myDSS 2.0](#)

ПОЛЕ	ТИП	ОПИСАНИЕ
KeyInfo	MyDssCreatedKeyInfo	информацию о зарегистрированном устройстве myDSS.
QrCode	string	данные о ключе в формате QR-кода.
QrCodeData	string	Данные для инициализации устройства в формате JSON

Информация об удаляемом устройстве пользователя (MyDssDeleteKeyRequest)

Сведения об удаляемом устройстве пользователя. Используется в методах групп [Аутентификация myDSS 2.0](#)

поле	тип	описание
kid	String	Идентификатор устройства пользователя

Информация о блокируемом устройстве пользователя (MyDssLockoutDeviceRequest)

Сведения о блокируемом устройстве пользователя. Используется в методах групп [Аутентификация myDSS 2.0](#)

поле	тип	описание
kid	string	Идентификатор устройства пользователя
lock	bool	Устройство требуется заблокировать (имеет значение "true") или разблокировать (имеет значение "false")

Информация об устройстве пользователя

(MyDssGetVerificationDataRequest)

Сведения об устройстве пользователя. Используется в методах групп [Аутентификация myDSS 2.0](#)

ПОЛЕ	ТИП	ОПИСАНИЕ
kid	String	Идентификатор устройства пользователя

Информация об устройстве пользователя (MyDssKeyInfo)

Сведения об устройстве пользователя. Используется в методах групп [Аутентификация myDSS 2.0](#)

ПОЛЕ	ТИП	ОПИСАНИЕ
kid	String	Идентификатор устройства пользователя
uid	String	Идентификатор (guid) пользователя DSS.
DeviceName	String	Имя устройства, на котором сохранён ключ.
DeviceParameters	IDictionary<string, string>	Параметры устройства, на котором сохранён ключ.
NotBefore	long	Срок начала действия ключа.
NotAfter	long	Срок окончания действия ключа.
State	string	Статус ключа (DeviceStates).
UserName	string	Имя пользователя (он же external user id, он же логин).
Profile	string	рофиль пользователя.
NonceRequired	bool	Требование подтверждения ключа с помощью Nonce

Сведения для получения QR-кода K-init (MyDssKInitRequest)

Сведения для получения QR-кода K-init Используется в методах групп [Аутентификация myDSS 2.0](#)

поле	тип	описание
Msisdn	string	(Optional) Используются для отправки кода активации (если требуется).
Email	string	(Optional) Используются для отправки кода активации (если требуется).

QR-код Nonce (MyDssVerificationData)

Данные для инициализации устройства. Используется в методах групп [Аутентификация myDSS 2.0](#)

ПОЛЕ	ТИП	ОПИСАНИЕ
Data	QRCodeVerificaton	Данные для инициализации устройства.
QrCode	string	данные инициализации устройства в формате QR-кода.
QrCodeData	string	Данные для инициализации устройства в формате JSON.

Данные QR-кода Nonce (QRCodeVerificaton)

Данные для инициализации устройства пользователя Используется в методах групп [Аутентификация myDSS 2.0](#)

ПОЛЕ	ТИП	ОПИСАНИЕ
type	string	тип данных. Имеет фиксированное значение <code>Verification</code>
kid	string	Идентификатор ключа
uid	string	Идентификатор пользователя
service_url	string	Адрес MyDSS Server
seed	byte[]	Данные инициализации ДСЧ
nonce	byte[]	Данные для подтверждения УЗ

Информация об устройствах пользователя (UserMyDssAuthInfo)

Информация о зарегистрированном устройстве myDSS Используется в методах групп [Аутентификация myDSS 2.0](#)

ПОЛЕ	ТИП	ОПИСАНИЕ
UserId	string	Идентификатор пользователя
Keys	MyDssKeyInfo[]	устройства пользователя
InitializationToken	MyDssCreatedKeyInfoEx	Данные для инициализации устройства пользователя

REST API Сервиса аудита

Данный раздел содержит руководство разработчика по интеграции с программным интерфейсом (API) Сервиса Аудита КriptoПро DSS.

В разделе приведено описание методов и типов данных REST-интерфейса Сервиса Аудита.

Конечные точки

- [Конечная точка Audit](#)
- [Конечная точка Reports](#)

Типы данных

- [Запрос на создание отчета \(CreateReportInput\)](#)
- [Запрос выборки записей аудита \(GetAuditRecordsInput\)](#)
- [Список зарегистрированных плагинов отчетов \(ReportPolicy\)](#)
- [Описание плагина формирования отчета \(ReportPluginDescription\)](#)
- [Описание параметра плагина формирования отчета \(ReportParameter\)](#)
- [Фильтр записей аудита \(DssRestAuditFilter\)](#)
- [Типы параметра плагина формирования отчета \(ReportParameterType\)](#)
- [Формат документа с отчетом \(DocumentFormatEnum\)](#)
- [Список колонок записи аудита \(DssAuditColumn\)](#)
- [Список операций фильтра аудита \(FilterOperation\)](#)

Конечные точки

- [Конечная точка Audit](#)
- [Конечная точка Reports](#)

Конечная точка Audit

Конечная точка для получения записей аудита

Описание

Конечная точка Audit позволяет осуществлять выборку записей аудита из БД Сервиса Аудита на основании фильтров, переданных в запросе.

Конечная точка /audit предоставляет следующие методы:

- [Запрос записей аудита](#)

Запрос записей аудита

Метод предназначен для осуществления фильтрованной выбоки записей аудита из БД Сервиса Аудита.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://dss.cryptopro.ru/AnalyticsService/api/audit
Параметры	GetAuditRecordsInput - запрос на создание отчета
Возвращаемое значение	DssAuditRecordsResponse - запрошенные записи аудита

Пример запроса:

```
POST /AnalyticsService/api/audit HTTP/1.1
Content-Type: application/json

{
  "StartPosition": 1,
  "EndPosition": 10,
  "Filters":
  [
    {
      "Column": 5,
      "Operation": 4,
      "FilterValue": "%прос%"
    }
  ]
}
```

Пример ответа:

```

{
  "Records": [
    {
      "Id": 674,
      "EventCode": 27,
      "EventSource": "SignServer",
      "SourceDetails": "1/SignServer",
      "Data": "Запрос списка сертификатов пользователя.",
      "Login": "Test1",
      "Realm": "realsts",
      "Date": "2020-04-29 13:59:04",
      "ExtendedData": "...",
      "DelegatedUserLogin": "",
      "EventLevel": 4
    },
    ...
    {
      "Id": 653,
      "EventCode": 288,
      "EventSource": "UserManagementService",
      "SourceDetails": "1/STS",
      "Data": "...",
      "Login": "audit",
      "Realm": "realsts",
      "Date": "2020-04-28 14:06:44",
      "ExtendedData": "...",
      "DelegatedUserLogin": "Test1",
      "EventLevel": 4
    }
  ],
  "AffectedCount": 396,
  "TotalCount": 828
}

```

Типовые ошибки

HTTP КОД	КОД ОШИБКИ	ОПИСАНИЕ
400	error_bad_bounds	Границы выборки заданные некорректно
400	error_invalid_filter_value	Значение фильтра задано некорректно
400	error_operation_not_supported	Запрошенная операция не поддерживается для выбранной колонки
500	-	Внутренняя ошибка сервера

Конечная точка Reports

Конечная точка для построения отчетов Сервиса Аудита

Описание

Конечная точка Reports позволяет формировать отчеты о деятельности ПАК КриптоПро DSS на основании информации, расположенной в БД Аудита.

Конечная точка /reports предоставляет следующие методы:

- [Создание отчета](#)
- [Получение информации о доступных отчетах](#)

Создание отчета

Предназначен для создания выбранного отчета.

Список доступных для создания на сервисе отчетов, а также требуемый список параметров можно получить при помощи конечной точки [Policy](#).

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://dss.cryptopro.ru/AnalyticsService/api/reports/create
Параметры	CreateReportInput - запрос на создание отчета
Возвращаемое значение	string - Base64-представление двоичных данных файла с отчетом

Примечание

Данный метод может быть вызван только для пользователей с ролями `Admins`, `Readonly` или `Audit`.

Пример запроса:

```
POST /AnalyticsService/api/reports/create HTTP/1.1
Content-Type: application/json

{
  "Parameters": {
    "StartDate": "01.01.2000",
    "EndDate": "10.10.2020"
  },
  "ReportFormat": "Xml",
  "ReportType": "gdr"
}
```

Пример ответа:

Ответ содержит base64

```
"77u/...ydD4="
```

Типовые ошибки

HTTP КОД	КОД ОШИБКИ	ОПИСАНИЕ
400	plugin_not_registered	Запрошенный плагин формирования отчета не зарегистрирован
400	required_param_not_set	Не задан обязательный параметр отчета
500	-	Внутренняя ошибка сервера

Получение информации о доступных отчетах

Данный метод предназначен для получения информации о плагинах формирования отчетов, доступных на сервисе.

Информация включает в себя Тип отчета, а также список параметров, с указанием того, какие из параметров являются обязательными.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://dss.cryptopro.ru/AnalyticsService/api/reports/policy
Возвращаемое значение	ReportPolicy - список описаний зарегистрированных плагинов формирования отчета

Пример запроса:

```
GET /AnalyticsService/api/reports/policy HTTP/1.1
```

Пример ответа:

Ответ содержит закодированное в Base-64 двоичное содержимое файла с отчетом.

```
[
  {
    "ReportType": "gdr",
    "ReportName": "Обобщенный отчет по СЭП DSS",
    "Parameters": [
      {
        "Type": 0,
        "Name": "StartDate",
        "Description": "Начальная дата",
        "Required": true
      },
      {
        "Type": 0,
        "Name": "EndDate",
        "Description": "Конечная дата",
        "Required": true
      }
    ]
  }
]
```

Типовые ошибки

HTTP КОД	КОД ОШИБКИ	ОПИСАНИЕ
500	-	Внутренняя ошибка сервера

Типы данных

- [Запрос на создание отчета \(CreateReportInput\)](#)
- [Запрос выборки записей аудита \(GetAuditRecordsInput\)](#)
- [Список зарегистрированных плагинов отчетов \(ReportPolicy\)](#)
- [Описание плагина формирования отчета \(ReportPluginDescription\)](#)
- [Описание параметра плагина формирования отчета \(ReportParameter\)](#)
- [Фильтр записей аудита \(DssRestAuditFilter\)](#)
- [Типы параметра плагина формирования отчета \(ReportParameterType\)](#)
- [Формат документа с отчетом \(DocumentFormatEnum\)](#)
- [Список колонок записи аудита \(DssAuditColumn\)](#)
- [Список операций фильтра аудита \(FilterOperation\)](#)

Тип данных `CreateReportInput`

Запрос на создание отчета.

ПОЛЕ	ТИП	ОПИСАНИЕ
Parameters	<code>Dictionary<string, string></code>	Параметры отчета.
ReportFormat	<code>DocumentFormatEnum</code>	Тип формата возвращаемого документа с отчетом.
ReportType	<code>string</code>	Тип отчета.

Тип данных

GetAuditRecordsInput

Запрос на фильтрованный выборку записей аудита из БД Сервиса Аудита.

поле	тип	описание
StartPosition	int	Стартовый индекс выборки относительно всех отфильтрованных записей.
EndPosition	int	Конечный индекс выборки относительно всех отфильтрованных записей.
Filters	ICollection<DssRestAuditFilter>	Список фильтров.

Тип данных `reportpolicy`

Список зарегистрированных плагинов формирования отчетов на сервисе.

поле	тип	описание
Parameters	IList< ReportPluginDescription >	Список зарегистрированных плагинов формирования отчета.

Тип данных ReportPluginDescription

Запрос на создание отчета.

поле	тип	описание
ReportType	<div>string</div>	Тип плагина формирования отчета.
ReportName	<div>string</div>	Имя отчета.
Parameters	<div>ICollection<ReportParameter></div>	Список параметров отчета.

Тип данных

ReportParameter

Описание параметра отчета.

поле	тип	описание
Type	ReportParameterType	Тип параметра отчета.
Name	string	Имя параметра отчета.
Description	string	Описание параметра отчета.
Required	bool	Значение, показывающее, является ли параметр отчета обязательным.

Тип данных DssRestAuditFilter

Объект, содержащий информацию о фильтре записей аудита.

поле	тип	описание
Column	<div>DssAuditColumn</div>	Колонка в БД, по которой осуществляется фильтрация.
Operation	<div>FilterOperation</div>	Операция, которая применяется фильтром.
FilterValue	<div>string</div>	Значение фильтра.

Тип данных `reportparametertype`

Перечисление с типом параметров отчета.

Возможные значения:

- String = 0
- Int = 1
- Bool = 2

Тип данных `documentformatenum`

Тип формата возвращаемого документа с отчетом.

Возможные значения:

- Html - представление в виде HTML,
- Xml - представление в виде XML,
- Pdf - представление в виде PDF-документа,
- Word - представление в виде Word-документа

Тип данных `DssAuditColumn`

Перечисление, содержащие список колонок, по которым можно осуществлять фильтрацию.

- Id (0) - идентификатор записи;
- Level (1) - уровень события;
- Code (2) - код события;
- Source (3) - источник события;
- SourceDetails (4) - информация об источнике;
- Data (5) - информация о событии;
- Date (6) - дата события;
- UserId (8) - идентификатор пользователя;
- Login (9) - имя пользователя;
- Realm (10) - realm пользователя;
- DelegatedUserId (11) - идентификатор пользователя, делегировавшего полномочия.

Тип данных `FilterOperation`

Перечисление, содержащие возможные операции фильтрации записей аудита.

- Equal (0) - операция равенства;
- NotEqual (1) - операция неравенства;
- Less (2) - операция "Меньше, чем";
- Greater (3) - операция "Больше, чем";
- Like (4) - операция подобия.

REST API Сервиса Сведений об Операциях

Данный раздел содержит руководство разработчика по интеграции с Сервисом Сведений об Операциях КriptoПро DSS. В разделе приведено подробное описание методов и типов данных REST-интерфейса.

Сервис предназначен для получения сведений об операциях пользователя.

Доступ к сервису может получить Пользователь или Информационная Система зарегистрированная как [Сервисный OAuth-клиент](#). Оператор DSS не может получить сведений об операциях пользователя.

Конечные точки

- [Operations](#) - предоставляет доступ к сведениям об операциях пользователя

Типы данных:

- [Сведения об операции пользователя \(OperationDto\)](#)
- [Действия входящие в операцию \(OperationDto\)](#)
- [Состояния действий входящих в операцию \(OperationActionStates\)](#)
- [Статус действий входящих в операцию \(OperationActionStatuses\)](#)
- [Состояния операции \(OperationStates\)](#)

Конечные точки

- [Operations](#) - предоставляет доступ к сведениям об операциях пользователя

Конечная точка Operations

Получение активных операций пользователя

Метод возвращает сведения о текущих (активных) операциях пользователя. Метод возвращает операции в State Created, Challenged.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://<адрес сервера>/<имя приложения ЦИ>/operations
Параметры	-
Возвращаемое значение	List<OperationDto> - Список активных операций пользователя

Пример запроса

```
GET /STS/operations HTTP/1.1
Host: hostname
Authorization: Bearer eyJ0eXAiOiJKV1 ... -80lQN4zfaZw
```

Пример ответа

```
[
  {
    "Id": "69a5b5c6-64d6-498d-8e0e-453ed7dc6de9",
    "Type": "SignDocument",
    "Parameters": "
{\\TSPAddress\\\": \"http://testca2012.cryptopro.ru/tsp/tsp.srf\\\", \\CADESType\\\": \\\"T\\\", \\IsDetached\\\": \\\"true\\\", \\
ProcessingTemplate\\\": \\\"1\\\", \\SignatureType\\\": \\\"CAdES\\\", \\CertificateID\\\": \\\"0\\\", \\DocumentInfo\\\": \\\"rpp.0401060
\\\", \\CertCommonName\\\": \\\"CN6\\\", \\CertSubjectName\\\": \\\"CN=CN6, SN=SN1, G=G2, S=S10, L=L9, STREET=Street5, O=08,
OU=OU7, T=T4, E=E12, I=I3\\\", \\CertIssuerName\\\": \\\"CN=\\\"\\\"Тестовый подчиненный УЦ ООО \\\"\\\"\\\"КРИПТО-
ПРО\\\"\\\"\\\" ГОСТ 2012 (УЦ 2.0)\\\"\\\", O=\\\"\\\"ООО \\\"\\\"\\\"КРИПТО-ПРО\\\"\\\"\\\"\\\"\\\", STREET=ул. Суцёвский вал д. 18,
L=Москва, S=77 Москва, C=RU, ИНН=007717107991, ОГРН=1037700085444,
E=info@cryptopro.ru\\\", \\CertSerialNumber\\\": \\\"01C716100152AB649F4E7DA54202ED9B13\\\", \\CertFriendlyName\\\": \\\"\\\"\\\"}
\",
    "Description": "Подпись документа. rpp.0401060. Тип подписи: CAdES. Сертификат: CN6.",
    "State": "Challenged",
    "CreatedAt": 1581522329,
    "CompleteBefore": 1581522929,
    "ConfirmBefore": 1581522360,
    "ConfirmedAt": 0,
    "CompletedAt": 0,
    "UpdatedAt": 1581522330,
    "UserId": "ae4d28b5-6d1a-4ef1-a1a2-f9d814de9cb9",
    "Context": null,
    "Proof": null,
    "AuthenticationType": "http://dss.cryptopro.ru/identity/authenticationmethod/mydss",
    "ExternalId": null,
    "Actions": [
      {
        "Id": "9f5f93f1-c521-4ba0-94a9-fdfa5a32e050",
        "DocumentId": "ba24eecc-a7b6-4ccf-8b6f-ebdabed73536",
        "OriginalDocumentId": null,
        "Status": "Considering",
        "State": "Pending",
        "ResultValue": null,
        "Error": null,
        "ErrorDescription": null
      }
    ]
  }
]
```

Получение сведений об операции пользователя

Метода возвращает сведения о запрошенной операции.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://<адрес сервера>/<имя приложения ЦИ>/operations/{op_id}
Параметры	op_id -идентификатор операции пользователя
Возвращаемое значение	OperationDto - сведения о запрошенной операции

Пример запроса

```
GET /STS/operations/69a5b5c6-64d6-498d-8e0e-453ed7dc6de9 HTTP/1.1
Host: hostname
Authorization: Bearer eyJ0eXAiOiJKV1 ... -80lQN4zfaZw
```

Пример ответа

```
{
  "Id": "69a5b5c6-64d6-498d-8e0e-453ed7dc6de9",
  "Type": "SignDocument",
  "Parameters": "
{\\\"TSPAddress\\\":\\\"http://testca2012.cryptopro.ru/tsp/tsp.srf\\\",\\\"CADESType\\\":\\\"T\\\",\\\"IsDetached\\\":\\\"true\\\",\\\"
ProcessingTemplate\\\":\\\"1\\\",\\\"SignatureType\\\":\\\"CADES\\\",\\\"CertificateID\\\":\\\"0\\\",\\\"DocumentInfo\\\":\\\"rpp.0401060
\\\",\\\"CertCommonName\\\":\\\"CN6\\\",\\\"CertSubjectName\\\":\\\"CN=CN6, SN=SN1, G=G2, S=S10, L=L9, STREET=Street5, O=O8,
OU=OU7, T=T4, E=E12, I=I3\\\",\\\"CertIssuerName\\\":\\\"CN=\\\"\\\"Тестовый подчиненный УЦ ООО \\\"\\\"\\\"КРИПТО-
ПРО\\\"\\\"\\\" ГОСТ 2012 (УЦ 2.0)\\\"\\\", O=\\\"\\\"ООО \\\"\\\"\\\"КРИПТО-ПРО\\\"\\\"\\\"\\\"\\\", STREET=ул. Суцёвский вал д. 18,
L=Москва, S=77 Москва, C=RU, ИНН=007717107991, ОГРН=1037700085444,
E=info@cryptopro.ru\\\",\\\"CertSerialNumber\\\":\\\"01C716100152AB649F4E7DA54202ED9B13\\\",\\\"CertFriendlyName\\\":\\\"\\\"}\"
,
  "Description": "Подпись документа. rpp.0401060. Тип подписи: CADES. Сертификат: CN6.",
  "State": "Challenged",
  "CreatedAt": 1581522329,
  "CompleteBefore": 1581522929,
  "ConfirmBefore": 1581522360,
  "ConfirmedAt": 0,
  "CompletedAt": 0,
  "UpdatedAt": 1581522330,
  "UserId": "ae4d28b5-6d1a-4ef1-a1a2-f9d814de9cb9",
  "Context": null,
  "Proof": null,
  "AuthenticationType": "http://dss.cryptopro.ru/identity/authenticationmethod/mydss",
  "ExternalId": null,
  "Actions": [
    {
      "Id": "9f5f93f1-c521-4ba0-94a9-fdfa5a32e050",
      "DocumentId": "ba24eccc-a7b6-4ccf-8b6f-ebdabed73536",
      "OriginalDocumentId": null,
      "Status": "Considering",
      "State": "Pending",
      "ResultValue": null,
      "Error": null,
      "ErrorDescription": null
    }
  ]
}
```

Статистика текущих операций

Получение количества текущих операций

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://<адрес сервера>/<имя приложения ЦИ>/operations/statistics
Параметры	отсутствуют
Возвращаемое значение	OperationStatisticOutput - статистика

Примечание

Метод не требует аутентификации.

Пример запроса

```
POST /STS/operations/statistics HTTP/1.1
Host: hostname
```

Пример ответа

```
{
  "TotalPendingOperations":100
  "CreatedOperations":10
  "CurrentProcessingOperation":70
}
```

Типы данных:

- Сведения об операции пользователя (OperationDto)
- Действия входящие в операцию (OperationDto)
- Состояния действий входящих в операцию (OperationActionStates)
- Статус действий входящих в операцию (OperationActionStatuses)
- Состояния операции (OperationStates)

Состояния операции (OperationStates)

поле	тип
Created	Операция создана на Сервисе Подписи
Challenged	Запрошено подтверждение операции
Cancelled	Операция отменена
Declined	Операция отклонена пользователем
Confirmed	Операция подтверждена пользователем
Completed	Операция выполнена
Error	Ошибка обработки операции
Expired	Операция истекла

REST API Сервиса Обработки Документов

Данный раздел содержит руководство разработчика по интеграции с программным интерфейсом (API) Сервиса Обработки Документов КriptoПро DSS.

В разделе приведено описание методов и типов данных REST-интерфейса Сервиса Обработки Документов.

Конечные точки

- [Конечная точка Documents](#)
- [Конечная точка Policy](#)

Типы данных

- [Информация о конвертации документа \(ConvertedDocumentInfo\)](#)
- [Информация о документе \(DocumentInfo\)](#)
- [Политика Сервиса Обработки Документов \(Policy\)](#)
- [Параметры загружаемого документа \(PostDocumentInput\)](#)
- [Результат загрузки документа \(UploadResult\)](#)
- [Дополнительные \(опциональные\) параметры документа \(AdditionalDocumentInfo\)](#)
- [Информация о поддерживаемом типе конвертации \(DocumentConvertations\)](#)
- [Информация о родительском документе \(LinkedDocument\)](#)
- [Результат загрузки пакета документов на Сервис Обработки Документов \(PackUploadResult\)](#)
- [Параметры загружаемого пакета документов \(PostDocumentPack\)](#)
- [Параметры загружаемого документа \(DocumentItem\)](#)

Конечные точки

- [Конечная точка Documents](#)
- [Конечная точка Policy](#)

Конечная точка Policy

Получение настроек Сервиса Обработки Документов

Описание

Конечная точка предназначена для получения настроек Сервиса Обработки Документов:

- Срок хранения документов
- Срок хранения временных документов
- Лимит памяти на пользователя
- Форматы документов

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://dss.cryptopro.ru/DocumentStore/rest/api/policy
Параметры	Метод не имеет параметров
Возвращаемое значение	Policy - Политика Сервиса Обработки Документов

Конечная точка Documents

Конечная точка /documents предоставляет следующие методы:

- Загрузка документа
- Загрузка пакета документов
- Получение информации о загруженном документе
- Получение информации о загруженном документе по хэшу
- Получение хранящегося документа
- Изменение информации о документе
- Связывание документа
- Получение привязанного документа
- Запрос на конвертацию документа
- Получение информации о сконвертированном документе
- Получение сконвертированного документа
- Получение информации о доступных преобразованиях для документа
- Получение сконвертированного документа для отображения
- Удаление документа

Загрузка документа

Предназначен потоковой загрузки документа в хранилище. При загрузке считается хэш документа. Содержимое документа передается в теле запроса с Content-Type: application/octet-stream, а дополнительные параметры – в отдельном HTTP заголовке, закодированном в формат Base64.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://dss.cryptopro.ru/DocumentStore/rest/api/documents
Тело запроса	Content byte[] - Содержимое документа
Заголовок CPDSS-POSTDOC	PostDocumentInput - Параметры загружаемого документа, в формате Base64 URL
Возвращаемое значение	UploadResult - Результат загрузки документа

Пример

Содержимое документа `test.txt`:

```
ATTRIBUTES:
GlobalID=fe16c231-804d-4681-ac1b-a7612bd55cf8
DocType=PayDocCur
DocTypeVersion=2.0
CreateDate=2019-05-16 14:28:08.58
CreateOrg=348409
FIELDS:
DocInfo.DocAccount=40702840838050000545
DocInfo.DocDate=2019-05-16
DocInfo.DocNumber=104
DocInfo.DocSum=7.00
DocInfo.DocSumCurrency=840
PayDocCur.AdditionalInfo=/FULLPAY/
PayDocCur.AmountTransfer=7.00
PayDocCur.BenefAccountCurISOCODE=USD
PayDocCur.BenefBankBIC=CHASUS31HPQ
PayDocCur.BenefBankBICType=SWIFT
PayDocCur.BenefBankCountryCode=840
```

```
PayDocCur.BenefBankCountryCode=040
PayDocCur.BenefBankCountryISOCODE=US
PayDocCur.BenefBankName=JPMORGAN CHASE BANK, N.A.
PayDocCur.BenefBankPlace=NEW YORK,NY
PayDocCur.BenefCountryCode=044
PayDocCur.BenefCountryISOCODE=BS
PayDocCur.Beneficiary=ASKOT INVESTMENTS LTD
PayDocCur.BeneficiaryAccount=40807840955130000002
PayDocCur.BeneficiaryAddress=NASSAU, EST HILL STREET, 3944
PayDocCur.BeneficiaryCountry=БАГАМЫ
PayDocCur.BeneficiaryPlace=NASSAU
PayDocCur.BranchShortName=0038-7981-01835
PayDocCur.ChargeBEN=false
PayDocCur.ChargeOUR=true
PayDocCur.ChargeSHA=false
PayDocCur.ChargesType=OUR
PayDocCur.CurrCodeISO=USD
PayDocCur.CurrCodeISOTransfer=USD
PayDocCur.CurrCodeTransfer=840
PayDocCur.Flag_TargetAssignment=false
PayDocCur.IMediaBankAddress=2, KRASNOGO TEKSTILSHCHIKA STREET
PayDocCur.IMediaBankBIC=SABRRU2P
PayDocCur.IMediaBankBICType=SWIFT
PayDocCur.IMediaBankCountryCode=643
PayDocCur.IMediaBankCountryISOCODE=RU
PayDocCur.IMediaBankName=SBERBANK (SEVERO-ZAPADNY HEAD OFFICE)
PayDocCur.IMediaBankPlace=ST. PETERSBURG
PayDocCur.IMediaClirCodeSymbol=RU
PayDocCur.IMediaClirCountryCode=RU
PayDocCur.IMediaClirShortName=Банковский идентификационный код (БИК)
PayDocCur.IsMultiCurr=false
PayDocCur.OfficialsPhone=76502587430
PayDocCur.Option50a=F
PayDocCur.Option56a=A
PayDocCur.Option57a=A
PayDocCur.Payer=JSC 'SRI PI.'
PayDocCur.PayerAddress=LENINA
PayDocCur.PayerBIC=044525225
PayDocCur.PayerBankBIC=SABRRUMM
PayDocCur.PayerBankName=SBERBANK (HEAD OFFICE - ALL BRANCHE
PayDocCur.PayerBankPlace=Москва
PayDocCur.PayerCountryCode=643
PayDocCur.PayerCountryISOCODE=RU
PayDocCur.PayerCountryNameRu=РОССИЯ
PayDocCur.PayerFiscalCode=7715784155
PayDocCur.PayerName=АО "НИИ ТП"
PayDocCur.PayerPlace=PERM
PayDocCur.PaymentDirection=0
PayDocCur.PaymentsDetails=FLOWING EXPENSES
PayDocCur.SenderOfficials=niitp_rut niitp_rut niitp_rut
PayDocCur.Urgent=false
```

Объект `PostDocumentInput` в формате JSON:

[illegible]

Значение заголовка CPDSS-POSTDOC (переносы строк добавлены для удобства чтения):

ewogICJGaWxlbmFtZSI6ICJ0ZXN0LnR4dCIsCiAgIkFkZG10aW9uYXwJbmZvIiA6ICIIAogICJJC1R1bXBvcnFyeSIgOiAiVHJ1ZSIsCiAgIkFkZG10aW9uYXwJbmZvIiA6IAogIHSKICAgICJTBmlwcGV0VGvtcGxhdGUiIDogIjxkaXYgc3R5bGU9XCJib3JkZXItcmFkaXVzOiA3cHg7XHJcb1x0XHRcdGJveC1zaGFkb3c6IDAgaCA3cHggcmdiYSgwLDA5MCwwLjUpO1xyXG5cdFw0XHRwYWRkaw5nOiAxNXB401xyXG5cdFw0XHRtYXJnaW46IDEwcHg7XHJcb1x0XHRcdGxpbmUtaGVpZ2h0OiAxLjM7XHJcb1x0XHRcdGZvbnQtZmFtaWw5OiBUYWhvbWE7XHJcb1x0XHRcdFwiPlxyXG5cdDxkaXYgc3R5bGU9XCJmb250LXNpemU6IDE2cHg7IGZvbnQtZ2VpZ2h0OiBib2xkOyBwYWRkaw5nLWJvdHRvbTogNXB401wiPlxyXG5cdFw0ezA6Rk1FERTLlBheURvY0N1ci5BZGRpdGlvbmFsSW5mb31cc1xuXHQ8XC9kaXy+XHJcb1x0PGRpdjBzdHlsZT1cImZvbnQtZ2VpZ2h0MTJweDs9Y29sb3I6IGdyYXk7XCI+XHJcb1x0XHR7MDpGSUVMRfMuUGF5RG9jQ3VyLkK1bmVmQmFua0NvdW50cn1JU09Db2RlFvxyXG5cdDxc1x2Rpdj5cc1xuXHRcc1xuXHQ8ZG12IHN0ewx1PVwiZm9udC1zaXplOiAxNnB40yBmb250LXdldodDogYm9sZDsgcGFkZGluZy1ib3R0b206IDVweDs9cGFkZGluZy10b3A6IDE4cHg7XCI+XHJcb1x0XHR7MDpGSUVMRfMuUGF5RG9jQ3VyLkNoYXJnZUJFTn1cc1xuXHQ8XC9kaXy+XHJcb1x0PGRpdjBzdHlsZT1cImZvbnQtZ2VpZ2h0MTJweDs9Y29sb3I6IGdyYXk7XCI+XHJcb1x0XHRcdTA0MTIgaXUuNDQyXHUuNDNFxHUuNDNDIFx1MDQ0N1x1MDQzOFx1MDQ0MVx1MDQzQ1x1MDQzNSBcdTA0MURcdTA0MTRcdTA0MjEgMTU1IC0gNC42MCBcdTA0M0ZcdTA0NDBcdTA0MzhcdTA0M0NcdTA0MzVcdTA0NDBcdTA0M0RcdTA0M0UgODkxIFx1MDQ0M1x1MDQ0Q1x1MDQ0MVx1MDQ0R1x1MDQ0NyBcdTA0NDBcdTA0NDNcdTA0MzFcdTA0M0JcdTA0MzVcdTA0Mzlcc1xuXHQ8XC9kaXy+XHJcb1x0XHJcb1x0PGRpdjBzdHlsZT1cImZvbnQtZ2VpZ2h0MTJweDs9cGFkZGluZy10b3A6IDE4cHg7XCI+XHJcb1x0XHQ8Yj57MDpGSUVMRfMuUGF5RG9jQ3VyLk1NZWRpYUJhbmtdDb3VudHJ5Q29kZX08XC9iPiBcdTIwQkRcc1xuXHQ8XC9kaXy+XHJcb1x0PCFbQ0RBVEFbe1xyXG5cdFwiRk1FERTLlBheURvY0N1ci5BZGRpdGlvbmFsSW5mb1wiOiBcc1xuXHR7XHJcb1x0XHRcIlwvRlVMTFBbWvXCI6IFwiXHUuNDFGXHUuNDNFxHUuNDNCXHUuNDNEXHUuNDMwXHUuNDRGIFx1MDQzRVx1MDQzR1x1MDQzQ1x1MDQzMFx1MDQ0M1x1MDQzMFwiLFxyXG5cdH0sXHJcb1x0XCJGSUVMRfMuUGF5RG9jQ3VyLkNoYXJnZUJFTlwiO1xyXG5cdFw0e1xyXG5cdFw0XCJmYXxzZVwiOiBcIlx1MDQzMVx1MDQzNVx1MDQzNyBcdTA0M0VcdTA0M0ZcdTA0M0JcdTA0MzBcdTA0NDJcdTA0NEIgaQkVOXCI5XHJcb1x0XHRcInRydWVcIjogXCJcdTA0NDEgXHUuNDNFxHUuNDNGXHUuNDNCXHUuNDMwXHUuNDQyXHUuNDNFxHUuNDM5IEJFTlwiXHJcb1x0fVxyXG59XV0+XHJcbjxcL2Rpdj4iLAogICAgIkRvY3VtZW50VGvtcGxhdGUiIDogIiIKICB9Cn0=

Пример запроса

POST /documentstore/api/documents HTTP/1.1
Host: simdss.cryptopro.ru
Content-Type: application/octet-stream
CPDSS-POSTDOC:
ewogICJGaWw1bmFtZSI6ICJ0ZXN0LnR4dCIsc2AgICFkFkZG10aW9uYXxJbmZvIiA6ICJlLAogICJjc1RlbXBvcnFyeSIgOiAiVHJ1ZSIsc2AgICFkFkZG10aW9uYXxJbmZvIiA6IAogIHSKICAgICJTBmlwcGV0VGVTcGxhdGUiIDogIjxkaXYgc3R5bGU9XCJib3JkZXItcmFkaXVzOiA3cHg7XHJcb1x0XHRcdGJveC1zaGFkb3c6IDAgMCA3cHggcmdiYSgwLdasMCwwLjUpO1xyXG5cdF0XHRwYWRkaw5n0iAxNXB401xyXG5cdF0XHRtYXJnaW46IDEwchG7XHJcb1x0XHRcdGxpbmUtaGVpZ2h00iAxLjM7XHJcb1x0XHRcdGZvbnQtZmFtaWx50iBUyWhvbWE7XHJcb1x0XHRcdFwiPlxyXG5cdDxkaXYgc3R5bGU9XCJmb250LXNpemU6IDE2cHg7IGZvbnQtZ2VpZ2h00iBib2xkOyBwYWRkaw5nLWJvdHRvbTogNXB401wiPlxyXG5cdF0ezA6RklFTERTLlBheURvY0N1ci5BZGRpdGlvbmfS5W5mb31cc1xuXHQ8XC9kaXY+XHJcb1x0PGRpdibZdHlsZT1cImZvbnQtZ2l6ZTogMTJweDsgY29sb3I6IGdyYXk7XCI+XHJcb1x0XHR7MDpGSUVMRfMuUGF5RG9jQ3VyLk1lbmVmQmFua0NvdW50cn1JU09Db2RlFVxyXG5cdDxcL2Rpdj5cc1xuXHRcc1xuXHQ8ZG12IHNoewx1PVwiZm9udC1zaXp10iAxNnB40yBmb250LXdlaWdodDogYm9sZDsgcGFkZGluZy1ib3R0b206IDVweDsgcGFkZGluZy10b3A6IDE4cHg7XCI+XHJcb1x0XHR7MDpGSUVMRfMuUGF5RG9jQ3VyLkNoYXJnZUJFTn1cc1xuXHQ8XC9kaXY+XHJcb1x0PGRpdibZdHlsZT1cImZvbnQtZ2l6ZTogMTJweDsgY29sb3I6IGdyYXk7XCI+XHJcb1x0XHRcdTA0MTIgxHUwNDQyXHUwNDNFxHUwNDNDIFx1MDQ0N1x1MDQzOFx1MDQ0MVx1MDQzQ1x1MDQzNSBcdTA0MURcdTA0MTRcdTA0MjEgMTU1IC0gNC42MCBcdTA0M0ZcdTA0NDBcdTA0MzhcdTA0M0NcdTA0MzVcdTA0NDBcdTA0M0RcdTA0M0UgODkxIFx1MDQ0M1x1MDQ0Q1x1MDQ0MVx1MDQ0R1x1MDQ0NyBcdTA0NDBcdTA0NDNcdTA0MzFcdTA0M0JcdTA0MzVcdTA0Mzlc1xuXHQ8XC9kaXY+XHJcb1x0XHR7MDpGSUVMRfMuUGF5RG9jQ3VyLk1nZWRpYUJhbmtdb3VudHJ5Q29kZX08XC9iPiBcdTIwQkRcc1xuXHQ8XC9kaXY+XHJcb1x0PCFbQ0RBVEFbe1xyXG5cdFwiRklFTERTLlBheURvY0N1ci5BZGRpdGlvbmfS5W5mb1wi0iBcc1xuXHR7XHJcb1x0XHRcIlwvRlVMTFBWVwvXCI6IFwiXHUwNDFGXHUwNDNFxHUwNDNCXHUwNDNEXHUwNDMwXHUwNDRGIFx1MDQzRVx1MDQzR1x1MDQzQ1x1MDQzMf1MDQ0M1x1MDQzMfwiLFXyXG5cdH0sXHJcb1x0XCJGSUVMRfMuUGF5RG9jQ3VyLkNoYXJnZUJFTlwi0lxyXG5cdF0e1xyXG5cdF0XCJmYXxzZVwioiBcIlx1MDQzMVx1MDQzNVx1MDQzNyBcdTA0M0VcdTA0M0ZcdTA0M0JcdTA0MzBcdTA0NDJcdTA0NEIgQkVOXCIsXHJcb1x0XHRcInRydWVcIjogXCJcdTA0NDEgXHUwNDNFxHUwNDNGXHUwNDNCXHUwNDMwXHUwNDQyXHUwNDNFxHUwNDM5IEJFTlwiXHJcb1x0fVxyXG59XV0+XHJcbjxcL2Rpdj4iLAogICAgICRvY3VtZW50VGVTcGxhdGUiIDogIiIKICB9Cn0=
Authorization: Bearer eyJ0eXAiOiJ....c0xtQ25ZEsw
cache-control: no-cache

Пример ответа

HTTP/1.1 200
Content-Type:"application/json; charset=utf-8"
Server:"Microsoft-IIS/10.0"
Date:"Tue, 27 Aug 2019 10:55:30 GMT"
Content-Length:"53"

{ "DocumentId": "1671c6bb-7c16-47e2-9c26-7a3c9e388f05" }

Типовые ошибки

HTTP КОД	КОД ОШИБКИ	ОПИСАНИЕ
400	invalid_object_state	Превышена квота для пользователя
400	invalid_request	Не найден связанный документ для загружаемого файла; Передан пустой файл.
500	-	Внутренняя ошибка сервера

Загрузка пакета документов

Предназначен для загрузки небольшого пакета документов в хранилище. При загрузке считается хэш документа. Содержимое документов передается в теле запроса.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://dss.cryptopro.ru/DocumentStore/rest/api/documents/pack

ПАРАМЕТР	ЗНАЧЕНИЕ
Тело запроса	PostDocumentPack - Содержимое документов
Возвращаемое значение	PackUploadResult - Результат загрузки документов

Пример

Содержимое документа `test.txt`:

```
test
```

Объект `PostDocumentPack` в формате JSON:

```
{
  "Documents":
    [

    {"Filename":"first.txt","IsTemporary":false,"AdditionalInfo":null,"ParentDocumentId":null,"FileType":null,"Content":"dGVzdA=="},

    {"Filename":"second.txt","IsTemporary":false,"AdditionalInfo":null,"ParentDocumentId":null,"FileType":null,"Content":"dGVzdA=="}
    ]
}
```

Пример запроса

```
POST /documentstore/api/documents/pack HTTP/1.1
Host: simdss.cryptopro.ru
Content-Type: application/json
Authorization: Bearer eyJ0eXAiOiJ...c0XtQ25ZEsw
cache-control: no-cache

{
  "Documents":
    [
    {"Filename":"first.txt","IsTemporary":false,"AdditionalInfo":null,"ParentDocumentId":null,"FileType":null,"Content":"dGVzdA=="},
    {"Filename":"second.txt","IsTemporary":false,"AdditionalInfo":null,"ParentDocumentId":null,"FileType":null,"Content":"dGVzdA=="}
    ]
}
```

Пример ответа

```
HTTP/1.1 200
Content-Type:"application/json; charset=utf-8"
Server:"Microsoft-IIS/10.0"
Date:"Tue, 27 Aug 2019 10:55:30 GMT"
Content-Length:"53"

{"DocumentIds":["8a211775-18fd-425e-ba35-9672bc2cf505","3e9ed5ec-da03-46a1-83c5-519a2ba1be2e"]}
```

Типовые ошибки

HTTP КОД	КОД ОШИБКИ	ОПИСАНИЕ
400	invalid_object_state	Превышена квота для пользователя

HTTP КОД	КОД ОШИБКИ	ОПИСАНИЕ
400	invalid_request	Не найден связанный документ для загружаемого файла; Передан пустой файл.
500	-	Внутренняя ошибка сервера

Получение информации о загруженном документе

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://dss.cryptopro.ru/DocumentStore/rest/api/documents/{id}/info
Параметры	id (обязательный), тип string - ID файла в хранилище документов
Возвращаемое значение	DocumentInfo - Информация о документе

Пример запроса

```
GET /documentstore/api/documents/1671c6bb-7c16-47e2-9c26-7a3c9e388f05/info HTTP/1.1
Host: simdss.cryptopro.ru
Authorization: Bearer eyJ0eXAiOiJ...c0XtQ25ZEsw
Accept: */*
Cache-Control: no-cache
Host: simdss.cryptopro.ru
Accept-Encoding: gzip, deflate
Connection: keep-alive
cache-control: no-cache
```

Пример ответа

```
HTTP/1.1 200
status: 200
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/10.0
Date: Tue, 27 Aug 2019 11:00:55 GMT
Content-Length: 1542
```

[illegible]

Типовые ошибки

HTTP КОД	КОД ОШИБКИ	ОПИСАНИЕ
404	-	Документ не найден
500	-	Внутренняя ошибка сервера

Получение информации о загруженном документе по хэшу

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://dss.cryptopro.ru/DocumentStore/rest/api/documents/hash{hash}/info
Параметры	hash (обязательный), тип string - хэш файла в хранилище документов
Возвращаемое значение	DocumentInfo - Информация о документе

Пример запроса


```
HTTP/1.1 200
status: 200
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/10.0
Date: Thu, 29 Aug 2019 08:38:13 GMT
Content-Length: 2703
```

[illegible]

Типовые ошибки

HTTP код	код ошибки	описание
404	-	Документ не найден
500	-	Внутренняя ошибка сервера

Получение хранящегося документа

Предназначен для получения содержимого загруженного документа.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://dss.cryptopro.ru/DocumentStore/rest/api/documents/{id}/content
Параметры	id (обязательный), тип string - ID файла в хранилище документов
Возвращаемое значение	тип Stream - Содержимое документа

Пример запроса

```
GET /documentstore/api/documents/1671c6bb-7c16-47e2-9c26-7a3c9e388f05/content HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJ....c0XtQ25ZEsw
Accept: */*
Cache-Control: no-cache
Host: simdss.cryptopro.ru
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

Пример ответа

```
HTTP/1.1 200
status: 200
Cache-Control: private
Content-Length: 2367
Content-Type: application/octet-stream
Server: Microsoft-IIS/10.0
X-AspNet-Version: 4.0.30319
Date: Tue, 27 Aug 2019 11:02:37 GMT

documentData
```

Типовые ошибки

HTTP КОД	КОД ОШИБКИ	ОПИСАНИЕ
404	-	Документ не найден
500	-	Внутренняя ошибка сервера

Изменение информации о документе

Предназначен для изменения информации о документе.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	PATCH
Путь	https://dss.cryptopro.ru/DocumentStore/rest/api/documents/{id}/info
Параметры	id (обязательный), тип string - ID файла в хранилище документов
Тело запроса	DocumentInfo - Информация о документе

Пример запроса

```
PATCH /documentstore/api/documents/1671c6bb-7c16-47e2-9c26-7a3c9e388f05 HTTP/1.1
Content-Type: application/json
Authorization: Bearer eyJ0eXAiOiJ....c0XtQ25ZEsw
Accept: */*
Cache-Control: no-cache
Host: simdss.cryptopro.ru
Accept-Encoding: gzip, deflate
Content-Length: 30
Connection: keep-alive
{
  "IsTemporary": false
}
```

Пример ответа

HTTP/1.1 200
status: 200
Server: Microsoft-IIS/10.0
Date: Tue, 27 Aug 2019 11:04:01 GMT
Content-Length: 0

Типовые ошибки

HTTP КОД	КОД ОШИБКИ	ОПИСАНИЕ
404	-	Документ не найден
500	-	Внутренняя ошибка сервера

Связывание документа

Предназначен для привязки другого документа к текущему

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://dss.cryptopro.ru/DocumentStore/rest/api/documents/{id}/link
Параметры	id (обязательный), тип string - ID файла в хранилище документов
Тело запроса	LinkedDocument - Информация о привязываемом документе

Пример запроса

```
POST /documentstore/api/documents/1671c6bb-7c16-47e2-9c26-7a3c9e388f05/link HTTP/1.1
Content-Type: application/json
Authorization: Bearer eyJ0eXAiOiJ...c0XtQ25ZEsw
Accept: */*
Cache-Control: no-cache
Host: simdss.cryptopro.ru
Accept-Encoding: gzip, deflate
Content-Length: 38
Connection: keep-alive
{
  "ParentDocumentId": "dcadd4f3-2126-4efb-99e6-121078b153e1"
}
```

Пример ответа

HTTP/1.1 200
status: 200
Server: Microsoft-IIS/10.0
Date: Tue, 27 Aug 2019 11:04:01 GMT
Content-Length: 0

Типовые ошибки

HTTP КОД	КОД ОШИБКИ	ОПИСАНИЕ
400	invalid_request	Не найден привязываемый документ

HTTP КОД	КОД ОШИБКИ	ОПИСАНИЕ
404	-	Документ не найден
500	-	Внутренняя ошибка сервера

Получение привязанного документа

Предназначен для получения идентификатора привязанного документа.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://dss.cryptopro.ru/DocumentStore/rest/api/documents/{id}/link
Параметры	id (обязательный), тип string - ID файла в хранилище документов
Тело запроса	LinkedDocument - Информация о привязываемом документе

Пример запроса

```
GET /documentstore/api/documents/1671c6bb-7c16-47e2-9c26-7a3c9e388f05/link HTTP/1.1
Content-Type: application/json
Authorization: Bearer eyJ0eXAiOiJ...c0XtQ25ZEsw
Accept: */*
Cache-Control: no-cache
Host: simdss.cryptopro.ru
Accept-Encoding: gzip, deflate
Content-Length: 0
Connection: keep-alive
```

Пример ответа

```
HTTP/1.1 200
status: 200
Server: Microsoft-IIS/10.0
Date: Tue, 27 Aug 2019 11:04:01 GMT
Content-Length: 53

{
  "DocumentId": "dcadd4f3-2126-4efb-99e6-121078b153e1"
}
```

Типовые ошибки

HTTP КОД	КОД ОШИБКИ	ОПИСАНИЕ
404	-	Документ не найден
500	-	Внутренняя ошибка сервера

Запрос на конвертацию без возвращения значения

Предназначен для инициирования конвертации предварительно загруженного документа.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://dss.cryptopro.ru/DocumentStore/rest/api/documents/{id}/converted_content/
Параметры	id (обязательный), тип string - ID файла в хранилище документов
Возвращаемое значение	ConvertedDocumentInfo - Информация о сконвертированном документе

Пример запроса

```
POST /documentstore/api/documents/1671c6bb-7c16-47e2-9c26-7a3c9e388f05/converted_content HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJ....c0XtQ25ZEsw
Accept: */*
Cache-Control: no-cache
Host: simdss.cryptopro.ru
Accept-Encoding: gzip, deflate
Content-Length:
Connection: keep-alive
```

Пример ответа

```
HTTP/1.1 202
status: 202
Server: Microsoft-IIS/10.0
Date: Tue, 27 Aug 2019 11:05:10 GMT
Content-Length: 0
```

Типовые ошибки

HTTP КОД	КОД ОШИБКИ	ОПИСАНИЕ
202	-	Запрос принят
200	-	Запрос обработан
404	-	Документ не найден
500	-	Внутренняя ошибка сервера

Получение информации о сконвертированном документе

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://dss.cryptopro.ru/DocumentStore/rest/api/documents/{id}/converted_content/info
Параметры	id (обязательный), тип string - ID файла в хранилище документов
Возвращаемое значение	ConvertedDocumentInfo - Информация о сконвертированном документе

Пример запроса

```
``` http
HTTP/1.1 200
status: 202
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/10.0
Date: Tue, 27 Aug 2019 11:11:45 GMT
Content-Length: 116

{
 "OriginalDocument": "1fdb3f8d-fc7f-4a19-b2f8-3eb6861e944f",
 "ConvertedFormat": "Pdf",
 "ConvertationStatus": "InProgress"
}
```

#### Пример ответа

```
HTTP/1.1 200
status: 200
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/10.0
Date: Tue, 27 Aug 2019 11:11:45 GMT
Content-Length: 116

{
 "OriginalDocument": "1671c6bb-7c16-47e2-9c26-7a3c9e388f05",
 "ConvertedFormat": "Pdf",
 "ConvertationStatus": "Completed"
}
```

#### Типовые ошибки

HTTP КОД	КОД ОШИБКИ	ОПИСАНИЕ
404	-	Документ не найден
500	-	Внутренняя ошибка сервера

### Получение сконвертированного документа

Предназначен для получения сконвертированного документа для отображения пользователю в веб-интерфейсе. Если в пути запроса передан номер страницы, то возвращаем указанную страницу сконвертированного документа. Если документ не был сконвертирован - запрос запускает процедуру конвертации. Если конвертация документа не успевает завершиться к моменту таймаута ответа - возвращает в ответе код 202 без тела.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://dss.cryptopro.ru/DocumentStore/rest/api/documents/{id}/converted_content/{page_id}?range={range}
Параметры	id (обязательный), тип string - ID файла в хранилище документов
Параметры	page_id (не обязательный), тип string - Номер страницы сконвертированного документа.
Параметры	range (не обязательный), тип string - Число страниц для выборки.
Возвращаемое значение	тип Stream - Содержимое сконвертированного документа

ПАРАМЕТР	ЗНАЧЕНИЕ
Возвращаемое значение, заголовок TotalPageCount	тип int - общее число страниц в документе, если в запросе был указан параметр page_id
Возвращаемое значение, заголовок ContentPageCount	тип int - число страниц в возвращаемом документе, если в запросе был указан параметр page_id

Пример запроса

```
GET /documentstore/api/documents/1671c6bb-7c16-47e2-9c26-7a3c9e388f05/converted_content HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJ....c0XtQ25ZEsw
Accept: */*
Cache-Control: no-cache
Host: simdss.cryptopro.ru
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

Пример ответа

```
HTTP/1.1 200
status: 200
Cache-Control: private
Content-Length: 42737
Content-Type: application/octet-stream
Server: Microsoft-IIS/10.0
X-AspNet-Version: 4.0.30319
Date: Tue, 27 Aug 2019 11:08:11 GMT

%PDF-1.4 ... %EOF
```

Типовые ошибки

HTTP КОД	КОД ОШИБКИ	ОПИСАНИЕ
400	invalid_object_data	Ошибка при конвертации документа
404	-	Документ не найден
500	-	Внутренняя ошибка сервера

Получение информации о доступных преобразованиях для документа

Предназначен для получения информации о доступных способах конвертации для документа.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://dss.cryptopro.ru/DocumentStore/rest/api/documents/{id}/supported_convertations
Параметры	id (обязательный), тип string - ID файла в хранилище документов.
Возвращаемое значение	тип List<DocumentConvertations> - информация о доступных способах конвертации для документа.

Пример запроса

```
GET /documentstore/api/documents/1671c6bb-7c16-47e2-9c26-7a3c9e388f05/supported_convertations HTTP/1.1
Accept: */*
Cache-Control: no-cache
Host: simdss.cryptopro.ru
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

#### Пример ответа

```
HTTP/1.1 200
status: 200
Server: Microsoft-IIS/10.0
Date: Tue, 27 Aug 2019 11:14:28 GMT
Content-Length: 202

[{"ConvertationType": "Snippet", "IsConvertationTypeSupported": "True"},
{"ConvertationType": "Document", "IsConvertationTypeSupported": "True"},
{"ConvertationType": "Raw", "IsConvertationTypeSupported": "True"}]
```

### Получение сконвертированного документа для отображения

Предназначен для получения сконвертированного документа для отображения пользователю в веб-интерфейсе.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://dss.cryptopro.ru/DocumentStore/rest/api/documents/{id}/preview?type={format}
Параметры	id (обязательный), тип string - ID файла в хранилище документов.
Параметры	format (обязательный), тип string - Допустимые значения: Snippet, Document.
Возвращаемое значение	тип Stream - Содержимое сконвертированного документа.
Возвращаемое значение, заголовок ContentHash	тип string - hex строка, содержащая хэш значение от содержимого ответа.

#### Пример запроса

```
GET /documentstore/api/documents/1671c6bb-7c16-47e2-9c26-7a3c9e388f05/preview?format=Snippet HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJ....c0XtQ25ZEsw
Accept: */*
Cache-Control: no-cache
Host: simdss.cryptopro.ru
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

#### Пример ответа

```
HTTP/1.1 200
status: 200
Cache-Control: private
Content-Length: 693
Content-Type: application/octet-stream
Server: Microsoft-IIS/10.0
ContentHash: F1EE8A63868F1DFB58A147C351EFB6A9BD8E5A8F3847B9133AF53B68C431A08C
X-AspNet-Version: 4.0.30319
Date: Tue, 27 Aug 2019 11:13:26 GMT

<div style="border-radius: 7px; box-shadow: 0 0 7px rgba(0,0,0,0.5); padding: 15px; margin: 10px;
line-height: 1.3; font-family: Tahoma; ">
 <div style="font-size: 16px; font-weight: bold; padding-bottom: 5px;">
 Полная оплата
 </div>
 <div style="font-size: 12px; color: gray;">
 US
 </div>
 <div style="font-size: 16px; font-weight: bold; padding-bottom: 5px; padding-top: 18px;">
 без оплаты BEN
 </div>
 <div style="font-size: 12px; color: gray;">
 В том числе НДС 15% - 4.60 примерно 891 тысяч рублей
 </div>
 <div style="font-size: 18px; padding-top: 13px;">
 643 Р
 </div>
</div>
```

Типовые ошибки

HTTP КОД	КОД ОШИБКИ	ОПИСАНИЕ
400	invalid_object_data	Ошибка при конвертации документа; Не найден ожидаемый шаблон.
400	invalid_request	Неизвестный формат.
400	forbidden_object_operation	Невозможно использовать переданный шаблон, так как использование внешних шаблонов запрещено.
404	-	Документ не найден
500	-	Внутренняя ошибка сервера

Удаление документа

Предназначен для удаления документа и связанных с ним сконвертированных документов из Хранилища документов.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	DELETE
Путь	https://dss.cryptopro.ru/DocumentStore/rest/api/documents/{id}
Параметры	id (обязательный), тип string - ID файла в хранилище документов

Пример запроса

DELETE /documentstore/api/documents/1671c6bb-7c16-47e2-9c26-7a3c9e388f05 HTTP/1.1  
Authorization: Bearer eyJ0eXAiOiJ....c0XtQ25ZEsw  
Accept: \*/\*  
Cache-Control: no-cache  
Host: simdss.cryptopro.ru  
Accept-Encoding: gzip, deflate  
Content-Length:  
Connection: keep-alive

Пример ответа

HTTP/1.1 200  
status: 200  
Server: Microsoft-IIS/10.0  
Date: Tue, 27 Aug 2019 11:14:28 GMT  
Content-Length: 0

Типовые ошибки

HTTP КОД	КОД ОШИБКИ	ОПИСАНИЕ
404	-	Документ не найден
500	-	Внутренняя ошибка сервера

## Конечная точка Service

Конечная точка /service предоставляет следующие методы:

- [Получение информации о загруженном документе](#)
- [Получение хранящегося документа](#)
- [Получение информации о доступных преобразованиях для документа](#)
- [Получение сконвертированного документа для отображения](#)
- [Получение сконвертированного документа](#)
- [Удаление документа](#)

Аутентификация производится по двухстороннему TLS. Для добавления нового сертификата необходимо воспользоваться командлетом Add-DssDocumentStoreClaimsProviderTrust.

## Загрузка документа

Предназначен потоковой загрузки документа в хранилище. При загрузке считается хэш документа. Содержимое документа передается в теле запроса с Content-Type: application/octet-stream, а дополнительные параметры – в отдельном HTTP заголовке, закодированном в формат Base64.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	POST
Путь	https://dss.cryptopro.ru/DocumentStore/rest/api/documents?UserId={userId}
Тело запроса	Content byte[] - Содержимое документа
Заголовок CPDSS-POSTDOC	<a href="#">PostDocumentInput</a> - Параметры загружаемого документа, в формате Base64 URL
userId	Идентификатор пользователя
Возвращаемое значение	<a href="#">UploadResult</a> - Результат загрузки документа

### Пример

Содержимое документа test.txt:

```
ATTRIBUTES:
GlobalID=fe16c231-804d-4681-ac1b-a7612bd55cf8
DocType=PayDocCur
DocTypeVersion=2.0
CreateDate=2019-05-16 14:28:08.58
CreateOrg=348409
FIELDS:
DocInfo.DocAccount=40702840838050000545
DocInfo.DocDate=2019-05-16
DocInfo.DocNumber=104
DocInfo.DocSum=7.00
DocInfo.DocSumCurrency=840
PayDocCur.AdditionalInfo=/FULLPAY/
PayDocCur.AmountTransfer=7.00
PayDocCur.BenefAccountCurISOCode=USD
PayDocCur.BenefBankBIC=CHASUS31HPQ
PayDocCur.BenefBankBICType=SWIFT
PayDocCur.BenefBankCountryCode=840
PayDocCur.BenefBankCountryISOCode=US
PayDocCur.BenefBankName=JPMORGAN CHASE BANK, N.A.
PayDocCur.BenefBankPlace=NEW YORK,NY
PayDocCur.BenefCountryCode=044
```

```
PayDocCur.BenefCountryISOCode=BS
PayDocCur.Beneficiary=ASKOT INVESTMENTS LTD
PayDocCur.BeneficiaryAccount=40807840955130000002
PayDocCur.BeneficiaryAddress=NASSAU, EST HILL STREET, 3944
PayDocCur.BeneficiaryCountry=БАГАМЫ
PayDocCur.BeneficiaryPlace=NASSAU
PayDocCur.BranchShortName=0038-7981-01835
PayDocCur.ChargeBEN=false
PayDocCur.ChargeOUR=true
PayDocCur.ChargeSHA=false
PayDocCur.ChargesType=OUR
PayDocCur.CurrCodeISO=USD
PayDocCur.CurrCodeISOTransfer=USD
PayDocCur.CurrCodeTransfer=840
PayDocCur.Flag_TargetAssignment=false
PayDocCur.IMediaBankAddress=2, KRASNOGO TEKSTILSHCHIKA STREET
PayDocCur.IMediaBankBIC=SABRRU2P
PayDocCur.IMediaBankBICType=SWIFT
PayDocCur.IMediaBankCountryCode=643
PayDocCur.IMediaBankCountryISOCode=RU
PayDocCur.IMediaBankName=SBERBANK (SEVERO-ZAPADNY HEAD OFFICE)
PayDocCur.IMediaBankPlace=ST. PETERSBURG
PayDocCur.IMediaClirCodeSymbol=RU
PayDocCur.IMediaClirCountryCode=RU
PayDocCur.IMediaClirShortName=Банковский идентификационный код (БИК)
PayDocCur.IsMultiCurr=false
PayDocCur.OfficialsPhone=76502587430
PayDocCur.Option50a=F
PayDocCur.Option56a=A
PayDocCur.Option57a=A
PayDocCur.Payer=JSC 'SRI PI.'
PayDocCur.PayerAddress=LENINA
PayDocCur.PayerBIC=044525225
PayDocCur.PayerBankBIC=SABRRUMM
PayDocCur.PayerBankName=SBERBANK (HEAD OFFICE - ALL BRANCHE
PayDocCur.PayerBankPlace=Москва
PayDocCur.PayerCountryCode=643
PayDocCur.PayerCountryISOCode=RU
PayDocCur.PayerCountryNameRu=РОССИЯ
PayDocCur.PayerFiscalCode=7715784155
PayDocCur.PayerName=АО "НИИ ТП"
PayDocCur.PayerPlace=PERM
PayDocCur.PaymentDirection=0
PayDocCur.PaymentsDetails=FLOWING EXPENSES
PayDocCur.SenderOfficials=niitp_rut niitp_rut niitp_rut
PayDocCur.Urgent=false
```

Объект `PostDocumentInput` в формате JSON:



[illegible]

Значение заголовка CPDSS-POSTDOC (переносы строк добавлены для удобства чтения):

ewogICJGaWxlbmFtZSI6ICJ0ZXN0LnR4dCIsCiAgIkFkZG10aW9uYXwJbmVzIiA6ICIiLAogICJJCjE1R1bXBvcnFyeSII0IAiVHJlZSI6ICAgIkFkZG10aW9uYXwJbmVzIiA6IAogIHsKICAgICJTBmlwcGV0VGVTcGxhdGUiIDogIjxkaXYgc3R5bGU9XCJib3JkZXItcmFkaXVzOia3Chg7XHJcb1x0XHRcdGJveCIzaGFkb3c6IDAgaMCA3chggcmdiYSgwLDA5MCwwLjUpO1xyXG5cdFx0XHRwYWRRkaw5nOiaXNXB401xyXG5cdFx0XHRtYXJnaW46IDEwecHg7XHJcb1x0XHRcdGxpbnUtaGVPZ2h0OiaXLjM7XHJcb1x0XHRcdGZvbnQtZmFtaWw5OiaBUyWhvbWE7XHJcb1x0XHRcdFwiPlxyXG5cdDxaXG5c3R5bGU9XCJmb250LXNpemU6IDE2cHg7IGZvbnQtd2VpZ2h0OiaBib2xkOyBwYWRkaw5nLWJvdHRvbTogNXB401wiPlxyXG5cdFx0ezA6RklFTERTLlBheURvY0N1ci5BZGRpdGlubmFsSW5mb31cc1xuXHQ8XC9kaXY+XHJcb1x0PGRpdibzdHlsZT1cImZvbnQtcl2ToGMTJweDsGY29sb3I6IGdyYXk7XCI+XHJcb1x0XHR7MDpGSUVMRFMuUGF5RG9jq3VyLkJl1bmVmQmFuaONvdW50cn1JU09Db2RlfVxyXG5cdDxcL2Rpdj5cc1xuXHRcc1xuXHQ8ZG12IHN0eww1PVwiZm9udC1zaXplOiaAxNnB40ybmb250LXdldawdodDogYm9sZDsgcGFkZGluZy1ib3R0b206IDVweDsGCfKZGluZy10b3A6IDE4cHg7XCI+XHJcb1x0XHR7MDpGSUVMRFMuUGF5RG9jq3VyLkNoYXJnZUJFTn1cc1xuXHQ8XC9kaXY+XHJcb1x0PGRpdibzdHlsZT1cImZvbnQtcl2ToGMTJweDsGY29sb3I6IGdyYXk7XCI+XHJcb1x0XHRcdTA0MTIgXHUwNDQyXHUwNDNFxHUwNDNDIFx1MDQ0N1x1MDQzOFx1MDQ0MVx1MDQzQ1x1MDQzNSbcdTA0MURcdTA0MTRcdTA0MjEgMTULIC0gNC42MCRcdTA0M0ZcdTA0NDBcdTA0MzhcdTA0M0NcdTA0MzVcdTA0NDBcdTA0M0RcdTA0M0UgdOkxIFx1MDQ0M1x1MDQ0Q1x1MDQ0MVx1MDQ0R1x1MDQ0NyBcdTA0NDBcdTA0NDNcdTA0MzfcdTA0M0jcdTA0MzVcdTA0Mzlcc1xuXHQ8XC9kaXY+XHJcb1x0XHJcb1x0PGRpdibzdHlsZT1cImZvbnQtcl2ZToGMThweDsGCfKZGluZy10b3A6IDE4cHg7XCI+XHJcb1x0XHQ8Yj57MDpGSUVMRFMuUGF5RG9jq3VyLklnZWRpYUJhbmtDb3VudHJ5J29kZX08XC9iPiBcdTIwQkRcc1xuXHQ8XC9kaXY+XHJcb1x0PCFbQ0RBVEFbe1xyXG5cdFwiRklFTERTLlBheURvY0N1ci5BZGRpdGlubmFsSW5mb1wiOiBcc1xuXHR7XHJcb1x0XHRcIlwvRlVMTFBBWvwvXCI6IFwiXHUwNDFGXHUwNDNFxHUwNDNCXHUwNDNEHXHUwNDMwXHUwNDRGIFx1MDQzRVx1MDQzRlx1MDQzQ1x1MDQzMfF1MDQ0M1x1MDQzMfFiLFxyXG5cdH0sXHJcb1x0XCJGSUVMRFMuUGF5RG9jq3VyLkNoYXJnZUJFTlwiOlxyXG5cdFx0e1xyXG5cdFx0XCJmYWxzZVwiOiBcIlx1MDQzMVx1MDQzNVx1MDQzNyBcdTA0M0VcdTA0M0ZcdTA0M0jcdTA0MzBcdTA0NDJcdTA0NEIgQkVOXCI6XHJcb1x0XHRcInRydWVcIjogXCJjc2dTA0NDEGXHUwNDNFxHUwNDNGXHUwNDNCXHUwNDMwXHUwNDQyXHUwNDNFxHUwNDM5IEJFTlwiXHJcb1x0fvxyXG59XV0+XHJcbjxcL2Rpdj4iLAogICAgaIkRvY3VtZW50VGVTcGxhdGUiIDogIiIKICB9Cn0=

## Пример запроса

### Пример ответа

## Типовые ошибки

HTTP КОД	КОД ОШИБКИ	ОПИСАНИЕ
400	invalid_object_state	Превышена квота для пользователя
400	invalid_request	Не найден связанный документ для загружаемого файла; Передан пустой файл.
500	-	Внутренняя ошибка сервера

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://dss.cryptopro.ru/DocumentStore/rest/api/service/documents/{id}/info
Параметры	id (обязательный), тип string - ID файла в хранилище документов



## Получение хранящегося документа

Предназначен для получения содержимого загруженного документа.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://dss.cryptopro.ru/DocumentStore/rest/api/service/documents/{id}/content
Параметры	id (обязательный), тип string - ID файла в хранилище документов
Возвращаемое значение	тип Stream - Содержимое документа

### Пример запроса

```
GET /documentstore/api/service/documents/1671c6bb-7c16-47e2-9c26-7a3c9e388f05/content HTTP/1.1
Accept: */*
Cache-Control: no-cache
Host: simdss.cryptopro.ru
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

### Пример ответа

```
HTTP/1.1 200
status: 200
Cache-Control: private
Content-Length: 2367
Content-Type: application/octet-stream
Server: Microsoft-IIS/10.0
X-AspNet-Version: 4.0.30319
Date: Tue, 27 Aug 2019 11:02:37 GMT

documentData
```

### Типовые ошибки

HTTP КОД	КОД ОШИБКИ	ОПИСАНИЕ
404	-	Документ не найден
500	-	Внутренняя ошибка сервера

## Получение информации о доступных преобразованиях для документа

Предназначен для получения информации о доступных способах конвертации для документа.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://dss.cryptopro.ru/DocumentStore/rest/api/service/documents/{id}/supported_convertations
Параметры	id (обязательный), тип string - ID файла в хранилище документов.
Возвращаемое значение	тип List< <a href="#">DocumentConvertations</a> > - информация о доступных способах конвертации для документа.

#### Пример запроса

```
GET /documentstore/api/service/documents/1671c6bb-7c16-47e2-9c26-7a3c9e388f05/supported_convertations
HTTP/1.1
Accept: */*
Cache-Control: no-cache
Host: simdss.cryptopro.ru
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

#### Пример ответа

```
HTTP/1.1 200
status: 200
Server: Microsoft-IIS/10.0
Date: Tue, 27 Aug 2019 11:14:28 GMT
Content-Length: 202

[{"ConvertationType":"Snippet","IsConvertationTypeSupported":"True"},
{"ConvertationType":"Document","IsConvertationTypeSupported":"True"},
{"ConvertationType":"Raw","IsConvertationTypeSupported":"True"}]
```

### Получение сконвертированного документа для отображения

Предназначен для получения сконвертированного документа для отображения пользователю в веб-интерфейсе.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://dss.cryptopro.ru/DocumentStore/rest/api/service/documents/{id}/preview?type={format}
Параметры	id (обязательный), тип string - ID файла в хранилище документов.
Параметры	format (обязательный), тип string - Допустимые значения: Snippet, Document.
Возвращаемое значение	тип Stream - Содержимое сконвертированного документа.
Возвращаемое значение, заголовок ContentHash	тип string - hex строка, содержащая хэш значение от содержимого ответа.

#### Пример запроса

```
GET /documentstore/api/service/documents/1671c6bb-7c16-47e2-9c26-7a3c9e388f05/preview?format=Snippet HTTP/1.1
Accept: */*
Cache-Control: no-cache
Host: simdss.cryptopro.ru
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

#### Пример ответа

```
HTTP/1.1 200
status: 200
Cache-Control: private
Content-Length: 693
Content-Type: application/octet-stream
Server: Microsoft-IIS/10.0
ContentHash: F1EE8A63868F1DFB58A147C351EFB6A9BD8E5A8F3847B9133AF53B68C431A08C
X-AspNet-Version: 4.0.30319
Date: Tue, 27 Aug 2019 11:13:26 GMT

<div style="border-radius: 7px; box-shadow: 0 0 7px rgba(0,0,0,0.5); padding: 15px; margin: 10px;
line-height: 1.3; font-family: Tahoma; ">
 <div style="font-size: 16px; font-weight: bold; padding-bottom: 5px;">
 Полная оплата
 </div>
 <div style="font-size: 12px; color: gray;">
 US
 </div>
 <div style="font-size: 16px; font-weight: bold; padding-bottom: 5px; padding-top: 18px;">
 без оплаты BEN
 </div>
 <div style="font-size: 12px; color: gray;">
 В том числе НДС 15% - 4.60 примерно 891 тысяч рублей
 </div>
 <div style="font-size: 18px; padding-top: 13px;">
 643 Р
 </div>
</div>
```

Типовые ошибки

HTTP КОД	КОД ОШИБКИ	ОПИСАНИЕ
400	invalid_object_data	Ошибка при конвертации документа; Не найден ожидаемый шаблон.
400	invalid_request	Неизвестный формат.
400	forbidden_object_operation	Невозможно использовать переданный шаблон, так как использование внешних шаблонов запрещено.
404	-	Документ не найден
500	-	Внутренняя ошибка сервера

Получение сконвертированного документа

Предназначен для получения сконвертированного документа для отображения пользователю в веб-интерфейсе. Если в пути запроса передан номер страницы, то возвращаем указанную страницу сконвертированного документа. Если документ не был сконвертирован - запрос запускает процедуру конвертации. Если конвертация документа не успевает завершиться к моменту таймаута ответа - возвращает в ответе код 202 без тела.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	GET
Путь	https://dss.cryptopro.ru/DocumentStore/rest/api/service/documents/{id}/converted_content/{page_id}?range={range}

ПАРАМЕТР	ЗНАЧЕНИЕ
Параметры	id (обязательный), тип string - ID файла в хранилище документов
Параметры	page_id (не обязательный), тип string - Номер страницы сконвертированного документа.
Параметры	range (не обязательный), тип string - Число страниц для выборки.
Возвращаемое значение	тип Stream - Содержимое сконвертированного документа
Возвращаемое значение, заголовок TotalPageCount	тип int - общее число страниц в документе, если в запросе был указан параметр page_id
Возвращаемое значение, заголовок ContentPageCount	тип int - число страниц в возвращаемом документе, если в запросе был указан параметр page_id

Пример запроса

```
GET /documentstore/api/documents/1671c6bb-7c16-47e2-9c26-7a3c9e388f05/converted_content HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJ....c0XtQ25ZEsw
Accept: */*
Cache-Control: no-cache
Host: simdss.cryptopro.ru
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

Пример ответа

```
HTTP/1.1 200
status: 200
Cache-Control: private
Content-Length: 42737
Content-Type: application/octet-stream
Server: Microsoft-IIS/10.0
X-AspNet-Version: 4.0.30319
Date: Tue, 27 Aug 2019 11:08:11 GMT

%PDF-1.4 ... %EOF
```

Типовые ошибки

HTTP КОД	КОД ОШИБКИ	ОПИСАНИЕ
400	invalid_object_data	Ошибка при конвертации документа
404	-	Документ не найден
500	-	Внутренняя ошибка сервера

Удаление документа

Предназначен для удаления документа и связанных с ним сконвертированных документов из Хранилища документов.

ПАРАМЕТР	ЗНАЧЕНИЕ
HTTP-метод	DELETE

ПАРАМЕТР	ЗНАЧЕНИЕ
Путь	https://dss.cryptopro.ru/DocumentStore/rest/api/service/documents/{id}
Параметры	id (обязательный), тип string - ID файла в хранилище документов

Пример запроса

```
DELETE /documentstore/api/service/documents/1671c6bb-7c16-47e2-9c26-7a3c9e388f05 HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJ....c0XtQ25ZEsw
Accept: */*
Cache-Control: no-cache
Host: simdss.cryptopro.ru
Accept-Encoding: gzip, deflate
Content-Length:
Connection: keep-alive
```

Пример ответа

```
HTTP/1.1 200
status: 200
Server: Microsoft-IIS/10.0
Date: Tue, 27 Aug 2019 11:14:28 GMT
Content-Length: 0
```

Типовые ошибки

НТТР КОД	КОД ОШИБКИ	ОПИСАНИЕ
404	-	Документ не найден
500	-	Внутренняя ошибка сервера



## Типы данных

- [Информация о конвертации документа \(ConvertedDocumentInfo\)](#)
- [Информация о документе \(DocumentInfo\)](#)
- [Политика Сервиса Обработки Документов \(Policy\)](#)
- [Параметры загружаемого документа \(PostDocumentInput\)](#)
- [Результат загрузки документа \(UploadResult\)](#)
- [Дополнительные \(опциональные\) параметры документа \(AdditionalDocumentInfo\)](#)
- [Информация о поддерживаемом типе конвертации \(DocumentConvertations\)](#)
- [Информация о родительском документе \(LinkedDocument\)](#)
- [Результат загрузки пакета документов на Сервис Обработки Документов \(PackUploadResult\)](#)
- [Параметры загружаемого пакета документов \(PostDocumentPack\)](#)
- [Параметры загружаемого документа \(DocumentItem\)](#)

Тип данных

ConvertedDocumentInfo

Информация о сконвертированном документе.

поле	тип	описание
OriginalDocument	string	ID оригинального файла в Хранилище Документов.
ConvertedFormat	string	ID сконвертированного файла в Хранилище Документов.
ConvertationStatus	string	Статус сконвертированного документа. Допустимые значения: InProcess, Complited, Error.
ConvertationStatusReason	string	Дополнительная информация о статусе конвертации.

Тип данных

DocumentInfo

Информация о документе, находящемся в Сервисе Обработки Документов.

поле	тип	описание
Id	string	ID файла в Хранилище Документов.
UserId	string	ID пользователя, имеющего права доступа к файлу.
Name	string	Название документа.
FileType	string	Расширение документа.
UploadTime	DateTime	Дата и время загрузки документа.
LastAccessTime	DateTime	Дата и время последнего обращения к документу.
IsTemporary	bool	Временный ли документ.
FileSize	integer	Размер документа.
Hash	string	Хэш файла
HashAlgorithm	string	Алгоритм хэширования файла
AdditionalInfo	<a href="#">AdditionalDocumentInfo</a>	Дополнительные сведения о документе.

Тип данных

Policy

Политики Сервиса Обработки Документов.

поле	тип	описание
DocumentStorageTime	int	Срок хранения документов.
TempDocumentStorageTime	int	Срок хранения временных документов.
MemoryLimitPerUser	int	Лимит памяти на пользователя
DocumentFormats	List<enum>	Форматы документов

Тип данных

PostDocumentInput

Параметры документа, загружаемого в Сервис Обработки Документов.

поле	тип	описание
Filename	string	Имя файла.
IsTemporary	bool	Является ли файл временным.
AdditionalInfo	AdditionalDocumentInfo	Дополнительные параметры документа.
ParentDocumentId	string	Идентификатор родительского документа.
FileType	string	Расширение файла (тип).

# Тип данных UploadResult

Результат загрузки документа Сервиса Обработки Документов.

поле	тип	описание
DocumentId	<div>string</div>	ID документа.

# Тип данных AdditionalDocumentInfo

Дополнительные (опциональные) параметры документа.

ПОЛЕ	ТИП	ОПИСАНИЕ
DocumentTemplate	string	Шаблон для конвертации.
SnippetTemplate	string	Шаблон для конвертации.
SmallFile	bool	Метка "маленький файл". При значении False файл будет сохранён с использованием FileStream, если DocumentStore настроен на работу с ним.

Тип данных

DocumentConversations

Информация о поддерживаемом типе конвертации

поле	тип	описание
ConverstationType	string	Тип конвертации. Принимает значения Snippet, Document, Raw.
IsConverstationTypeSupported	bool	Данный тип конвертации поддерживается для данного документа.
ConverstationStatusReason	string	Причина неподдерживания данного типа конвертации.



# Тип данных LinkedDocument

Информация о родительском документе.

поле	тип	описание
DocumentId	<div>string</div>	ID родительского документа.

# Тип данных PackUploadResult

Результат загрузки пакета документов на Сервис Обработки Документов.

поле	тип	описание
DocumentIds	List<string>	Список ID документов.

Тип данных

PostDocumentPack

Параметры пакета документов, загружаемых в Сервис Обработки Документов.

поле	тип	описание
AdditionalInfo	List< <a href="#">DocumentItem</a> >	Дополнительные параметры документа.

Тип данных

DocumentItem

Параметры документа, загружаемого в Сервис Обработки Документов.

ПОЛЕ	ТИП	ОПИСАНИЕ
Filename	string	Имя файла.
IsTemporary	bool	Является ли файл временным.
AdditionalInfo	AdditionalDocumentInfo	Дополнительные параметры документа.
ParentDocumentId	string	Идентификатор родительского документа.
FileType	string	Расширение файла (тип).
Content	string	Содержимое документа в виде BASE64 строки

# REST API

В разделе приведено описание методов Сервиса Управления Пользователями для работы с SIM-картами. Базовый адрес Сервиса Управления Пользователями:

```
https://<hostname>/<StsAppName>/ums
```

Сервис Управления Пользователями предназначен для использования Операторами DSS. Аутентификация Операторов DSS на Сервисе Управления Пользователями осуществляется по сертификату (двустороннее TLS-соединение).

## Назначение пользователю аутентификации на SIM-карте

```
POST user/{id}/simauth
```

## Получение сведений о назначенной пользователю SIM-карте

```
GET user/{id}/simauth
```

## Удаление SIM-карты, назначенной пользователю

```
DELETE user/{id}/simauth
```

## Смена ключа аутентификации на SIM-карте

```
POST user/{id}/simauth/changekeymessage
```

### Параметры

- `id` - идентификатор пользователя.

Ответ сервиса будет содержать:

- `TransactionId` - идентификатор транзакции смены ключа аутентификации;
- `ActivationCode2` - код смены ключа.

Код смены ключа аутентификации представляет собой список из 10 блоков цифр, которые необходимо ввести пользователю на мобильном устройстве.

Результат смены ключа можно узнать по `TransactionId`, вызвав [метод получения результата транзакции](#).

**Внимание!**  
Получение результата выполнения запроса на апплете возможно либо через CallBack-сервис, либо периодическим опросом [метода получения результата транзакции](#)

## Отправка запроса на апплет

```
POST user/{id}/simauth/message/{messageType}
```

ИМЯ СООБЩЕНИЯ	КОД СООБЩЕНИЯ	ОПИСАНИЕ
Activate	0xAA	Запрос на активацию апплета.
GetStatus	0xAB	Запрос на получение статуса апплета.
ChangePin	0xAC	Запрос изменения ПИН-кода.

Ответ сервиса будет содержать:

- `TransactionId` - идентификатор транзакции смены ключа аутентификации

Результат запроса к апплету можно узнать по `TransactionId`, вызвав [метод получения результата транзакции](#).

## Внимание!

Получение результата выполнения запроса на апплете возможно либо через CallBack-сервис, либо периодическим опросом [метода получения результата транзакции](#).

## Получение результата транзакции

```
POST user/{id}/simauth/message/{transactionId}
```

## Параметры

- `id` - идентификатор пользователя.

Ответ сервиса будет содержать:

- `AppletResult` - результат выполнения операции;
- `IsCompleted` - флаг, показывающее, завершена ли обработка запроса.

Значения поле `AppletResult`

ИМЯ	КОД	ОПИСАНИЕ
OperationInProgress	0x00	Операция выполняется (результат будет отправлен в исходящем сообщении).
AppletActivated	0x01	Апплет активирован.
AppletInactive	0x02	Апплет не активирован.
AppletBlocked	0x03	Апплет заблокирован (превышено количество попыток ввода кода активации/PIN).
InvalidMessage	0x04	Неправильный формат сообщения.
AppletIsBusy	0x05	Апплет занят.
AppletAlreadyActivated	0x06	Апплет уже активирован.
AppletActivated2	0x07	Апплет активирован (ключ 2).
OperationConfirmed	0x10	Операция одобрена пользователем.
OperationCancelled	0x11	Операция отменена пользователем.
OperationTimeout	0x12	Операция не выполнена из-за превышения времени ожидания.
DstkError	0x13	Ошибка обработки DSTK.
PinNotChanged	0x14	ПИН-код не изменён.
PinChanged	0x15	ПИН-код изменён.
TextLengthExceeded	0x17	Превышена длина текста.

ИМЯ	КОД	ОПИСАНИЕ
KeyAlreadyChanged	0x18	Ключ уже был обновлён.
InvalidKey	0x19	Неверный ключ.
KeyChangeSucceed	0x20	Ключ успешно сменён.

# Аутентификация с помощью мобильного приложения

**КриптоПро myDSS** — это приложение для смартфона, которое позволяет аутентифицировать действия пользователя в системах дистанционного банкинга, порталах госуслуг, системах ЭДО, электронных торговых площадках и т.д.

## Пользователям

- [Руководство пользователя myDSS](#) — документ, включающий в себя инструкции по созданию и настройке аутентификации учетной записи, выпуску сертификата, формированию электронной подписи документа и подтверждению данной операции при помощи мобильного приложения myDSS.

## Разработчикам

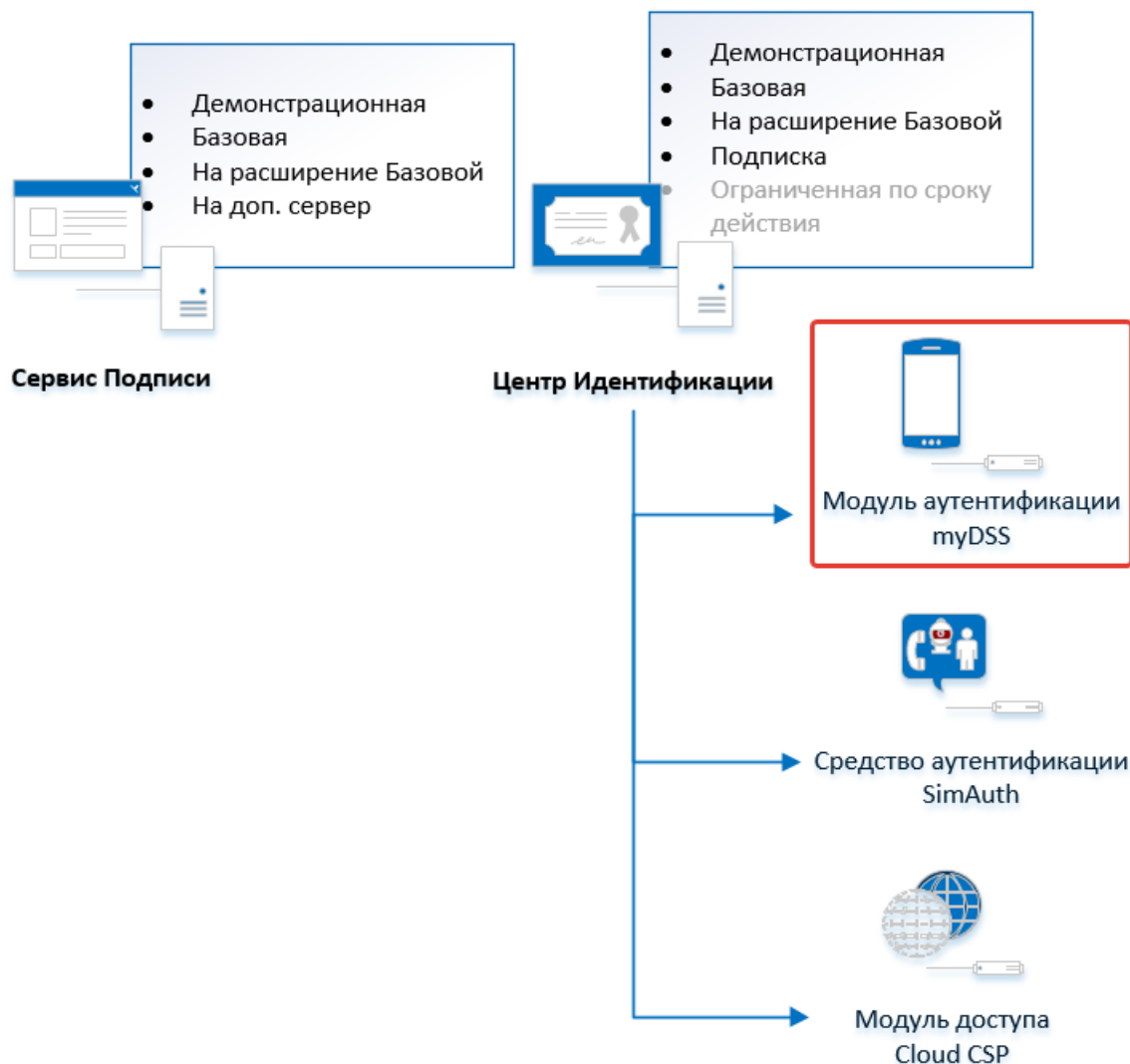
Раздел содержит руководство разработчика по интеграции с myDSS:

- [Назначение и управление аутентификацией через myDSS](#) — руководство разработчика по работе через Оператора DSS.
- [Подтверждение операций через myDSS](#) — руководство разработчика по работе через пользователя DSS.
- [Выпуск сертификата пользователя](#) — руководство разработчика по выпуску сертификата пользователя.



# Лицензирование метода аутентификации myDSS

Метод аутентификации myDSS является одним из средств аутентификации КриптоПро DSS:



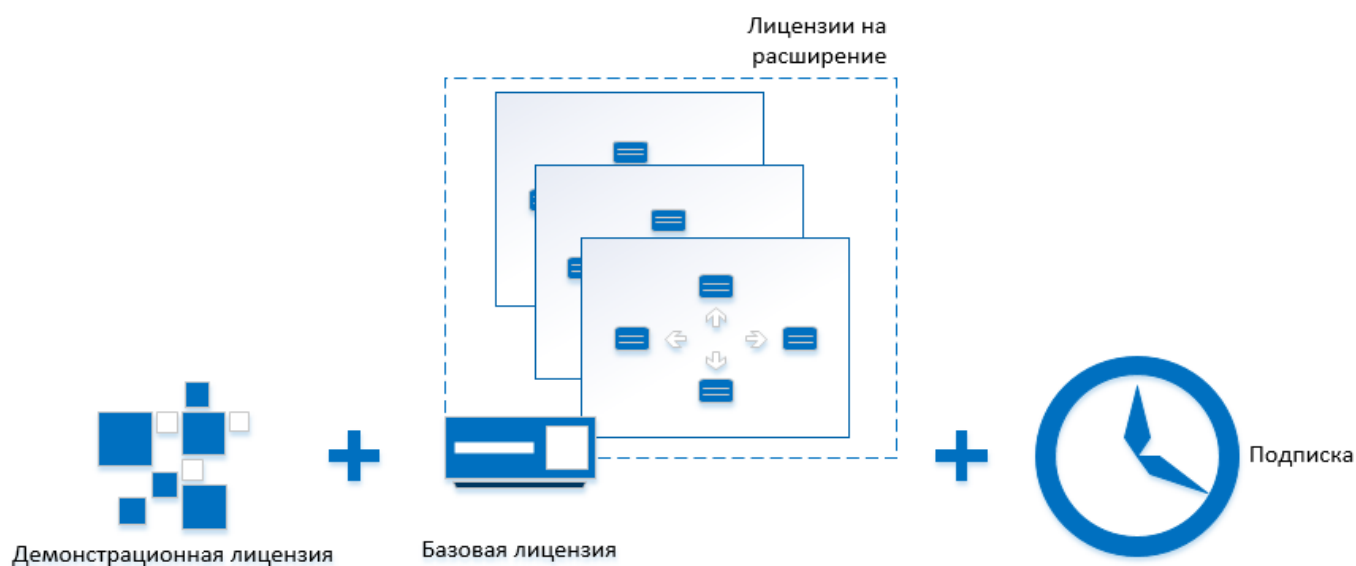
Существуют следующие типы лицензий на модуль аутентификации myDSS:

- **Демонстрационная.** Активируется автоматически при создании экземпляра ЦИ, рассчитана на 10 Пользователей и не ограничена по сроку действия. Не удаляется после ввода лицензии другого типа.
- **Базовая.** Лицензия выдается на ограниченное количество Пользователей и не ограничена по времени. Для одного экземпляра ЦИ может быть введена только одна Базовая лицензия. При вводе новой Базовой лицензии старая уничтожается, а переназначение лицензии Пользователям происходит автоматически при попытке подтверждения операции или вручную при назначении метода аутентификации myDSS.
- **На расширение Базовой лицензии.** Лицензия дополняет Базовую (не может быть введена без нее) и выдается на ограниченное количество Пользователей. Лицензий На расширение может быть сколько угодно.
- **Подписка.** В лицензии типа Подписка ограничен как срок действия лицензии, так и количество Пользователей (кол-во активаций). Особенности лицензии:
  - Срок действия лицензии отсчитывается индивидуально с момента активации каждого Пользователя. Активацией Пользователя называется включение данному Пользователю метода аутентификации myDSS.
  - При включении Пользователю метода аутентификации myDSS тратится одно использование лицензии (одна активация). Если срок действия лицензии для данного Пользователя истек и у него по-прежнему включён метод аутентификации myDSS, тратится еще одно использование лицензии (одна активация). Данный процесс происходит автоматически и повторяется, пока метод аутентификации myDSS остаётся включённым для

данного Пользователя.

- Активация лицензии закрепляется за Пользователем. В течение срока действия активированной лицензии включение/отключение для данного Пользователя метода аутентификации myDSS не тратит новую активацию.
- Лицензия может быть введена, даже если Базовая лицензия отсутствует.

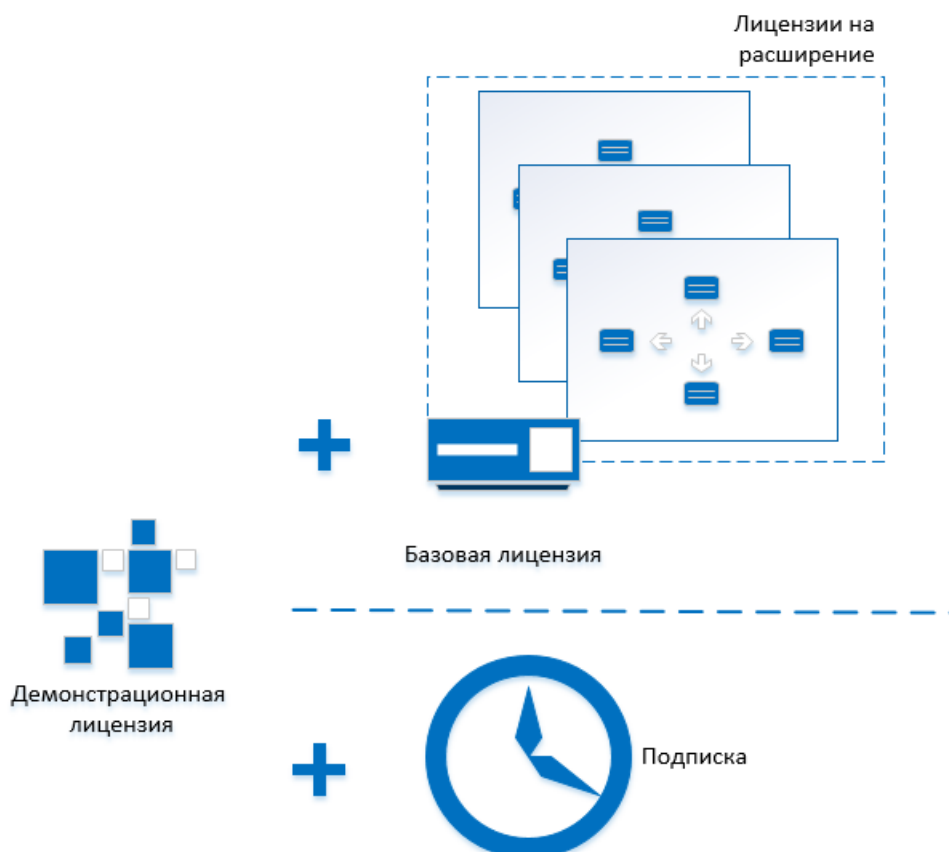
### Лицензии могут быть введены как последовательно :



### Так и параллельно:

#### Примечание

Лицензии типа Подписка не требуются наличие лицензий других типов.



Некоторые Пользователи, использующие метод аутентификации myDSS, могли быть присоединены к **Ограниченной по сроку действия** лицензии. В данном типе лицензии прописан явно срок окончания действия (общий для всех пользователей, начинает отсчитываться с момента ввода лицензии), а также может быть ограничено количество Пользователей. Данная лицензия не требует обязательного наличия Базовой и/или лицензии На расширение, но может их дополнять. Ограниченная по сроку действия лицензия может быть введена только одна.

### Примечание

Ограниченная по сроку лицензия перестает полноценно действовать в версиях DSS 2.0.2849 и выше. При этом невозможны:

- Ввод и удаление лицензии данного типа.
- Присоединение к действующей лицензии новых Пользователей.

### Примечание

Особенности использования ограниченной по сроку лицензии в версиях DSS 2.0.2849 и выше:

- Прикрепленные к действующей лицензии пользователи могут пользоваться ей до истечения срока действия при условии неотключения метода аутентификации myDSS.
- Если отключить пользователю, прикрепленному к ограниченной по сроку лицензии, метод аутентификации myDSS, следующее назначение ему этого метода прикрепляет его к имеющейся свободной лицензии (Приоритет: Базовая, На расширение, Подписка).

### Соответствие типов лицензий форме заказов

Перечисленные в данной статье названия типов лицензий используются для краткости. Точное соответствие типов лицензий на КриптоПро myDSS версии 2.0 названиям из [формы заказов](#) представлено в таблице ниже.

### Примечание

Соответствие Демонстрационной лицензии в данной таблице не приводится, т.к. данный тип лицензии доступен сразу после установки и настройки КриптоПро DSS вне зависимости от других приобретенных лицензий.

ТИП ЛИЦЕНЗИИ	НАИМЕНОВАНИЕ ПОЗИЦИИ В ФОРМЕ ЗАКАЗОВ
Базовая	Лицензия на право использования ПО "Модуль аутентификации myDSS для ПАК "КриптоПро DSS" версии 2.0 до <количество> пользователей
На расширение Базовой лицензии	Лицензия на расширение права использования ПО "Модуль аутентификации myDSS для ПАК "КриптоПро DSS" версии 2.0 на <количество> пользователей
Подписка (ранее - Ограниченная по времени)	Лицензия на право использования ПО "Модуль аутентификации myDSS для ПАК "КриптоПро DSS" версии 2.0 до <количество> пользователей (<срок, например "годовая">)

# Подключение метода аутентификации myDSS

Раздел содержит руководство разработчика по работе с myDSS на Сервисе Управления Пользователями. В разделе приведены основные сценарии использования, примеры HTTP-запросов и ответов REST Сервиса Управления Пользователями.

Так же в разделе приведены рекомендации Администраторам по настройке DSS для реализации различных сценариев работы с myDSS.

Перед началом интеграции с Сервисом Управления Пользователями Администратору DSS необходимо:

- Выпустить и зарегистрировать на DSS сертификат аутентификации Оператора DSS
- Настроить сервер myDSS
- Включить метод аутентификации myDSS на Центре Идентификации
- Ввести лицензию на модуль аутентификации myDSS на Центре Идентификации

Сценарии должны выполняться учётной записью с ролью Оператора DSS.

Аутентификация Операторов DSS на Сервисе Управления Пользователями осуществляется по сертификату (двухстороннее TLS-соединение).

Последовательность шагов по регистрации пользователя:

1. [Регистрация логина пользователя.](#)
2. [Назначение метода первичной аутентификации.](#)
3. [Получение QR-кода с ключом аутентификации myDSS.](#)
4. [Назначение myDSS в качестве второго фактора аутентификации.](#)
5. [Назначение операций, требующие подтверждения через myDSS.](#)

Вспомогательные действия:

1. [Обновление ключа аутентификации пользователя](#)
2. [Поиск пользователя](#)
3. [Получение сведений о myDSS](#)
4. [Отключение аутентификации через myDSS](#)
5. [Задание отпечатка устройства](#)

## Регистрация логина пользователя

В качестве идентификатора (логина) пользователя могут выступать:

- логин
- адрес электронной почты
- номер телефона

### Внимание!

По умолчанию на DSS в качестве идентификатора разрешён только Логин.

Разрешить/запретить другие идентификаторы пользователя может Администратор DSS выполнив команду в консоли PowerShell:

```
Set-DssStsProperties -AvailableIdentifiers Login,Email,PhoneNumber
```

## Примеры запросов

- Регистрация пользователя по логину

```
POST https://host/STS/ums/user HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 28
Expect: 100-continue

{"Login":"DssTest-6f956360"}
```

- Регистрация пользователя по логину, email и номеру телефона

```
POST https://host/STS/ums/user HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 92
Expect: 100-continue
Connection: Keep-Alive

{"Login":"DssTest-05e789e7","PhoneNumber":"+70004064846","Email":"DssTest-0678acd4@dss.com"}
```

### Пример ответа

В ответ DSS вернёт идентификатор созданного пользователя (DssUserId). DssUserId используется при вызове любых методов Сервиса Управления Пользователями:

- возвращающих сведения об учётной записи пользователя
- изменяющих учётную запись пользователя.

Вызывающая система может сохранить DssUserId. Это позволит ускорить последующие обращения к Сервису Управления Пользователями, так как не потребуется получать DssUserId повторно.

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Date: Thu, 23 Aug 2018 14:59:09 GMT
Content-Length: 38

"264caee9-b77c-4b8c-b52a-ef9dd502f959"
```

### Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_identifiers	Переданный идентификатор запрещённый на DSS.
400	invalid_phone	Пользователь с указанным номер телефона уже зарегистрирован.
400	invalid_email	Пользователь с указанным email уже зарегистрирован.
400	invalid_login	Пользователь с указанным логином уже зарегистрирован.
500	An error has occurred	1. В поле Login указан номер телефона или email. 2. Неверно сформирован email. 3. Неверно сформирован номер телефона.

Назначение метода первичной аутентификации

После регистрации логина пользователя необходимо назначить метод первичной аутентификации. Пользователю может быть назначен один или несколько методов первичной аутентификации:

МЕТОД	ОПИСАНИЕ
/user/{DssUserId}/authmethod/idonly	Только идентификация
/user/{DssUserId}/authmethod/password	Аутентификация по паролю
/user/{DssUserId}/authmethod/cert	Аутентификация по сертификату
/user/{DssUserId}/authmethod/external	Аутентификация через сторонний Центр Идентификации

Чаще всего при использовании myDSS в качестве метода первичной аутентификации назначают "Только идентификация".

**Внимание!**

Назначаемый метод аутентификации должен быть разрешён на DSS. Включить или отключить метод аутентификации должен Администратор на сервере DSS.

Разрешить/запретить метод аутентификации можно на Сервере DSS командами:

```
Enable-DssAuthenticationMethod -Id <method_ID>
```

```
Disable-DssAuthenticationMethod -Id <method_ID>
```

**Внимание!**

Совместное включение методов idonly и password допустимо, но использоваться будет метод "Только идентификация".

Примеры запросов

Назначение метода первичной аутентификации "Только идентификация"

```
POST https://host/STS/ums/user/d0bfa4e1-808d-4c28-b3cb-0dc5a591d300/authmethod/idonly HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 2
Expect: 100-continue

{}
```

Пример ответа

Назначение метода аутентификации не имеет возвращаемого значения.

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Date: Thu, 23 Aug 2018 16:35:38 GMT
Content-Length: 0
```

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	wrong_operation	Метод аутентификации уже назначен.
400	invalid_authn_method	Метод аутентификации запрещён на сервере DSS.
404	user_not_found	Пользователь не найден.

## Получение QR-кода с ключом аутентификации myDSS

Перед назначением пользователю метода аутентификации myDSS необходимо получить QR-код, содержащий ключ аутентификации пользователя. QR-код должен быть передан пользователю. Отсканировав QR-код пользователь загрузит ключ аутентификации в мобильное приложение myDSS.

Ключ аутентификации, передаваемый в QR-коде, может быть защищён на коде активации. Код активации передаётся пользователю в SMS или email сообщении.

Требование защиты ключа аутентификации на коде активации настраивается Администратором на сервере DSS и распространяется на всех пользователей.

```
Set-DssMobileAuthProperties -KeyInfoDivideRequired 1
```

Код активации состоит из цифр. Минимальная длина кода - 6 цифр.

Изменить длину кода активации может Администратор DSS выполнив команду в консоли PowerShell:

```
Set-DssMobileAuthProperties -SecondKeyPartLength 8
```

## Примечание

Для отправки кода активации в SMS или Email Администратору необходимо подключить и настроить соответствующий модуль оповещения на сервере DSS.

## Примеры запросов

- Пример получения QR-кода без кода активации

```
POST https://host/STS/ums/user/29665c07-e0e1-496a-b8f6-96ac5f59501b/mydss/offline HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 50
Expect: 100-continue

{ }
```

- Пример получения QR-кода с кодом активации, доставляемом в SMS или Email

Для передачи кода активации в SMS или Email необходимо указать параметр `UserContactInfoType`, который может принимать значение

- EmailAddress
- PhoneNumber

В параметре `UserContactInfo` передаётся адрес электронной почты или номер телефона.

```
POST https://host/STS/ums/user/3901bd80-0cc4-4011-b197-3064dc41d626/mydss/offline HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 119
Expect: 100-continue

{"UserContactInfo":"+70007321826","UserContactInfoType":"PhoneNumber"}
```

Пример ответа

Сервер возвращает следующие данные:

- `QrCode` - изображение в формате gif в кодировке Base64
- `KeyInfo` - информация о созданном ключе, в том числе blobs ключей, срок действия и др.

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Content-Length: 7711

{
 "QrCode": "R01GOD1hLAEsAfcAAAA...AAA/+loREAA7",
 "KeyInfo": {
 "AuthKeyBlob":
"ASAAAB5mAAD9UUo3HmYAAAWWSA1THGa+jZ6uZjL7YB4wEhrGYKZfY9MdVnVtWydBV1QbPsYcQ2N4Iv0MAsgCSqFAwcBAGUBAQEAAACT1932",
 "ConfirmKeyBlob":
"ASAAAB5mAAD9UUo3HmYAAFegRNck3J5zTrGiM9wFYvrNyQ9Yi0b3tjTV1eVhvLMD2VYX1iTKLLigDzhFMAsGCSqFAwcBAGUBAQEAAADADIFK",
 "Seed": "k2ms7QY3QLw7tPvV/9RJBldzc/Yt2cFmk9R7lMQjDZU=",
 "ActivationRequired": false,
 "ServiceUrl": "https://simdss.cryptopro.ru:4430/mydss",
 "Uid": "6d961728-8039-4f34-95f7-725e4988acd8",
 "Kid": "68278185",
 "DeviceName": null,
 "NotBefore": 1567336799,
 "NotAfter": 1606821599,
 "State": "Created"
 }
}
```

Типовые ошибки

HTTP-код	ошибка	описание
400	invalid_contact_info	1. Требуется предоставить номер телефона или email для отправки кода активации. 2. Указан неверный тип данных для отправки кода активации.
404	user_not_found	Пользователь не найден.
400	wrong_operation	Попытка повторно получить QR-код. Для обновления ключа пользователя необходимо отправить PATCH запрос.
500	An error has occurred	1. Метод аутентификации myDSS запрещен на сервере DSS. 2. Истекла лицензия на myDSS на сервере DSS.



После того как пользователю создан ключ аутентификации необходимо назначить метод аутентификации myDSS.

Пример запроса

```
POST https://host/STS/ums/user/44cdb94b-7168-4b26-b02a-a734510873e8/authmethod/mydss?level=1 HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 2
Expect: 100-continue

{}
```

**Внимание!**  
Значение параметра `level` должно быть равно 1

Пример ответа

Метод не имеет возвращаемого значения.

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Date: Thu, 23 Aug 2018 21:02:26 GMT
Content-Length: 0
```

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_authentication_scheme	Указан неверный уровень метода аутентификации.
404	user_not_found	Пользователь не найден.
400	authn_method_not_confirmed	Попытка назначить метод аутентификации не получив QR-код.

Назначение операций, требующие подтверждения через myDSS

После получения QR-кода и назначения пользователю myDSS необходимо задать список операций требующий подтверждения с помощью myDSS.

Про типы операций, для которых можно настроить подтверждение, и их идентифкаторы можно прочитать [на странице Типы Операций](#). В запросе необходимо перечислить [коды операций](#), которые будет подтверждать пользователь.

Пример запроса

```
POST https://host/STS/ums/user/b33f2148-70de-4e2b-b508-db3f8bf72a7e/operationpolicy HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 5
Expect: 100-continue

[2, 16, 1024]
```

Пример ответа

Метод не имеет возвращаемого значения

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Content-Length: 0
```

### Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_authentication_scheme	Указан неверный уровень метода аутентификации.
404	user_not_found	Пользователь не найден.
400	wrong_operation	Оператору запрещено изменять список операций, требующих подтверждения.

### Обновление ключа аутентификации пользователя

Ключ аутентификации пользователя имеет ограниченный срок действия. Ключ аутентификации необходимо периодически обновлять. Процедура смены ключа аналогична получению [первого ключа аутентификации](#)

### Примечание

Отличие в используемом HTTP-методе - для обновления ключа используется метод **PATCH**.

Флаг `DelayedActivation` отвечает за режим активации нового ключа пользователя:

- Если флаг отсутствует в запросе или равен **false**, то новый ключ пользователя вступает в действие немедленно. Пользователь не сможет подтверждать операции пока не отсканирует QR-код с новым ключом.
- Если флаг указан в запросе со значением **true**, то новый ключ пользователя вступит в действие в момент, когда пользователь отсканирует QR-код с новым ключом. Пользователь сможет продолжить использование текущего ключа до истечения срока его действия.

### Пример запроса

```
PATCH https://host/STS/ums/user/d217ee28-0f04-45d2-b71a-0ebb50ee7573/mobileauth HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 119
Expect: 100-continue

{"UserContactInfo":"+70004297594","UserContactInfoType":"PhoneNumber"}
```

### Пример запроса с отложенной активацией ключа

```
PATCH https://host/STS/ums/user/6b33998b-ae10-4132-9291-9fbc97aa0942/mobileauth HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 118
Expect: 100-continue

{"UserContactInfo":"+70005471676","UserContactInfoType":"PhoneNumber","DelayedActivation":true}
```

### Пример ответа

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Content-Length: 7687

{"XmlKeyInfo":"","QRCode":"R0lGODlhL ... oA7pAQEA0w==","KeyExpirationTime":"2018-09-23T00:00:00"}
```

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_authentication_scheme	Указан неверный уровень метода аутентификации.
404	user_not_found	Пользователь не найден.
400	wrong_operation	Оператора запрещено изменять список операций, требующий подтверждения.

Повторная отправка кода активации пользователю

Если ключ аутентификации уже назначен пользователю и защищён на коде активации, то можно сделать повторную от отправку кода активации ключа.

Требование защиты ключа аутентификации на коде активации настраивается Администратором на сервере DSS и распространяется на всех пользователей.

```
Set-DssMobileAuthProperties -KeyInfoDivideRequired 1
```

Для передачи кода активации в СМС или по адресу электронной почты необходимо указать параметр `UserContactInfoType`, который может принимать значение

- EmailAddress
- PhoneNumber

В параметре `UserContactInfo` передаётся адрес электронной почты или номер телефона пользователя.

Пример запроса

Метод не имеет возвращаемого значения.

```
POST /STS/ums/user/0a0eaf08-665e-452c-a5b0-e342b3c43a3c/mobileauth/activationcode HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host

{"UserContactInfo":"+70007321826","UserContactInfoType":"PhoneNumber"}
```

Пример ответа

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Content-Length: 0
```

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_contact_info	1. Нет возможности отправить вторую часть ключевой информации: не задана контактная информация пользователя. 2. Неизвестный тип контактной информации: "EmailAddress".
400	wrong_operation	Код активации не требуется в соответствии с настройками сервиса.
404	user_not_found	Пользователь не найден.

## Поиск пользователя

Сервис Управления пользователями предоставляет несколько возможностей поиска пользователя:

- По логину, номеру телефона или email
- По идентификатору DssUserId
- Расширенный поиск

По логину, номеру телефона или email

### Пример запроса

Тип ключа поиска может принимать значения (значение параметра `type`):

- Login
- PhoneNumber
- Email

```
GET https://host/STS/ums/user?type=Login&value=DssTest-dc3bf3f5 HTTP/1.1
Accept: application/json
Host: host
```

### Пример ответа

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Content-Length: 334

{"UserId":"d1831dea-985f-4df1-a54b-2497eeace2f2","Login":"DssTest-dc3bf3f5","PhoneNumber":null,"Email":null,"PhoneConfirmed":false,"EmailConfirmed":false,"DisplayName":null,"DistinguishName":"","AccountLocked":false,"Group":"Default","CreationDate":"2018-08-24T14:36:33.02","LockoutDate":null,"LastLoginDate":"2018-08-24T14:36:33.02"}
```

По идентификатору DssUserId

### Пример запроса

```
GET https://host/STS/ums/user/d1831dea-985f-4df1-a54b-2497eeace2f2 HTTP/1.1
Accept: application/json
Host: host
```

### Пример ответа

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Content-Length: 334

{"UserId": "d1831dea-985f-4df1-a54b-2497eeace2f2", "Login": "DssTest-
dc3bf3f5", "PhoneNumber": null, "Email": null, "PhoneConfirmed": false, "EmailConfirmed": false, "DisplayName": null, "D
istinguishName": "", "AccountLocked": false, "Group": "Default", "CreationDate": "2018-08-
24T14:36:33.02", "LockoutDate": null, "LastLoginDate": "2018-08-24T14:36:33.02"}
```

Расширенный поиск

Расширенный поиск позволяет применять различные фильтры для поиска пользователей. Результатом выполнения метода может быть группа пользователей, отвечающая параметрам фильтра.

Поиск пользователей можно выполнить по одному или нескольким параметрам:

ПАРАМЕТР	КОД	ОПИСАНИЕ
Login	0	Логин пользователя
PhoneNumber	1	Номер телефона
Email	2	Адрес электронной почты
CreateDate	3	Дата создания учётной записи
GroupId	4	Идентификаторы группы пользователя

Код параметра указывается в поле

Операции сравнения могут быть следующих типов:

ТИП	КОД	ОПИСАНИЕ
Equal	0	Строгое равенство
NotEqual	1	Не равно
Like	2	Содержит
Greater	3	Больше
Less	4	Меньше

Код операции указывается в поле

Тип сравнения Like определяет, совпадает ли указанная символьная строка с заданным шаблоном. Шаблон может включать обычные символы и символы-шаблоны. Во время сравнения с шаблоном необходимо, чтобы его обычные символы в точности совпадали с символами, указанными в строке. Символы-шаблоны могут совпадать с произвольными элементами символьной строки.

Поддерживаются следующие символы шаблоны:

СИМВОЛ-ШАБЛОН	ОПИСАНИЕ	ПРИМЕР
%	Любая строка, содержащая ноль или более символов.	%вано%
(подчеркивание)	Любой одиночный символ.	_етров
[ ]	Любой одиночный символ, содержащийся в диапазоне ([a-f]) или наборе ([abcdef]).	[Л-С]омов
[^]	Любой одиночный символ, не содержащийся в диапазоне ([^a-f]) или наборе ([^abcdef]).	'ив[^a]%

Параметры `StartPosition` и `EndPosition` определяют начальную и конечную позицию из итоговой выборки. Данные параметры могут быть использованы для страничной выборки пользователей

При поиске пользователей по времени создания значение фильтра должно иметь следующий формат: **yyyy-MM-ddThh:mm:ss**

Общее количество элементов подпадающих под критерии фильтра возвращается в параметре `TotalCount`. Количество элементов отданных методом возвращается в параметре `AffectedCount`: `AffectedCount <= EndPosition - StartPosition`

Примеры запросов

Получить пользователя с заданным логином:

```
POST https://host/STS/ums/users HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 101
Expect: 100-continue

{"StartPosition":1,"EndPosition":1,"Filters":[{"Column":0,"Operation":0,"Value":"DssTest-dc3bf3f5"}]}
```

Проверка были ли создан пользователь с заданным логином в указанном промежуток времени:

```
POST https://host/STS/ums/users HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 172
Expect: 100-continue

{"StartPosition":1,"EndPosition":1,"Filters":[{"Column":0,"Operation":0,"Value":"DssTest-2fa204c5"}, {"Column":3,"Operation":3,"Value":"2018-08-24T15:12:12.4683672+03:00"}]}
```

Получить пользователей созданных в указанный промежуток времени:

```
POST https://host/STS/ums/users HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 161
Expect: 100-continue

{"StartPosition":1,"EndPosition":10,"Filters":[{"Column":3,"Operation":4,"Value":"2018-08-25T04:24:50"}, {"Column":3,"Operation":3,"Value":"2018-08-23T04:24:50"}]}
```

Пример ответа

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Content-Length: 386
```

```
{"UserInfos":[{"UserId":"d1831dea-985f-4df1-a54b-2497eeace2f2","Login":"DssTest-
dc3bf3f5","PhoneNumber":null,"Email":null,"PhoneConfirmed":false,"EmailConfirmed":false,"DisplayName":null,"D
istinguishName":"","AccountLocked":false,"Group":"Default","CreationDate":"2018-08-
24T14:36:33.02","LockoutDate":null,"LastLoginDate":"2018-08-
24T14:36:33.02"}],"TotalCount":2047,"AffectedCount":1}
```

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404	user_not_found	Пользователь не найден.
500	An error has occurred	Неверно указано значение или тип фильтра.

Получение сведений о myDSS

Ниже приведён список методов, которые позволяют проверить:

- назначен ли пользователю ключ аутентификации myDSS
- срок действия ключа аутентификации myDSS
- назначен ли пользователю метод аутентификации myDSS
- список действий требующих подтверждения

Проверка назначен ли ключ аутентификации myDSS пользователю

Пример запроса

```
GET https://host/STS/ums/user/c9b4b217-edc1-4329-be4d-1cddaecdea4d/mobileauth HTTP/1.1
Accept: application/json
Host:host
```

Примеры ответов

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Content-Length: 91

{"UserId":"9912c464-8d2e-4145-bcee-587f10dad61b","KeyExpirationTime":"2018-09-23T00:00:00"}
```

Если ключ аутентификации не назначен пользователю ответ сервиса будет содержать **null**:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Content-Length: 4

null
```

Получение схемы аутентификации пользователя

Пример запроса

```
GET https://host/STS/ums/user/01df2040-3e16-4405-9947-fc4152448c13/authmethod HTTP/1.1
Accept: application/json
Host: host
```

Пример ответа

Сервис возвращает список элементов содержащих:

- Идентификатор метода аутентификации
- Уровень метода аутентификации

Методы первичной аутентификации имеют уровень 0.

Методы вторичной аутентификации имеют уровень 1.

Список идентификаторов методов первичной аутентификации:

ИДЕНТИФИКАТОР	ОПИСАНИЕ
<a href="http://schemas.microsoft.com/ws/2012/09/identity/authenticationmethod/none">http://schemas.microsoft.com/ws/2012/09/identity/authenticationmethod/none</a>	Только идентификация
<a href="http://dss.cryptopro.ru/identity/authenticationmethod/password">http://dss.cryptopro.ru/identity/authenticationmethod/password</a>	По паролю
<a href="http://dss.cryptopro.ru/identity/authenticationmethod/certificate">http://dss.cryptopro.ru/identity/authenticationmethod/certificate</a>	По сертификату
<a href="http://dss.cryptopro.ru/identity/authenticationmethod/saml">http://dss.cryptopro.ru/identity/authenticationmethod/saml</a>	Через сторонний Центр Идентификации

Список идентификаторов методов вторичной аутентификации:

ИДЕНТИФИКАТОР	ОПИСАНИЕ
<a href="http://dss.cryptopro.ru/identity/authenticationmethod/mobile">http://dss.cryptopro.ru/identity/authenticationmethod/mobile</a>	Аутентификация через мобильное приложениеуDSS
<a href="http://dss.cryptopro.ru/identity/authenticationmethod/otpviasms">http://dss.cryptopro.ru/identity/authenticationmethod/otpviasms</a>	Одноразовые пароли по SMS
<a href="http://dss.cryptopro.ru/identity/authenticationmethod/otpviaemail">http://dss.cryptopro.ru/identity/authenticationmethod/otpviaemail</a>	Одноразовые пароли по Email
<a href="http://dss.cryptopro.ru/identity/authenticationmethod/oath">http://dss.cryptopro.ru/identity/authenticationmethod/oath</a>	Аутентификация по протоколу Oath
<a href="http://dss.cryptopro.ru/identity/authenticationmethod/simauth">http://dss.cryptopro.ru/identity/authenticationmethod/simauth</a>	Аутентификация на SIM-карте
<a href="http://dss.cryptopro.ru/identity/authenticationmethod/airkey">http://dss.cryptopro.ru/identity/authenticationmethod/airkey</a>	Аутентификация через мобильное приложение Indeed AirKey

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Content-Length: 189

[{"MethodUri": "http://schemas.microsoft.com/ws/2012/09/identity/authenticationmethod/none", "Level": 0}, {"MethodUri": "http://dss.cryptopro.ru/identity/authenticationmethod/mobile", "Level": 1}]
```

Типовые ошибки



HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404	user_not_found	Пользователь не найден.

Получение списка операций, требующих подтверждения

### Пример запроса

```
GET https://host/STS/ums/user/8386abb0-b3d0-44c0-96a7-90d635e45d21/operationpolicy HTTP/1.1
Accept: application/json
Host: 192.168.109.149
```

### Пример ответа

Описание операций приведено в разделе [выше](#)

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Content-Length: 679

[{"Action": "Issue", "ConfirmationRequired": true}, {"Action": "SignDocument", "ConfirmationRequired": true}, {"Action": "SignDocuments", "ConfirmationRequired": false}, {"Action": "DecryptDocument", "ConfirmationRequired": false}, {"Action": "CreateRequest", "ConfirmationRequired": false}, {"Action": "ChangePin", "ConfirmationRequired": false}, {"Action": "RenewCertificate", "ConfirmationRequired": true}, {"Action": "RevokeCertificate", "ConfirmationRequired": false}, {"Action": "HoldCertificate", "ConfirmationRequired": false}, {"Action": "UnholdCertificate", "ConfirmationRequired": false}, {"Action": "DeleteCertificate", "ConfirmationRequired": false}, {"Action": "PrivateKeyAccess", "ConfirmationRequired": false}]
```

### Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404	user_not_found	Пользователь не найден.

### Отключение myDSS

Отключение аутентификации через myDSS состоит из последовательности шагов:

1. Отключить требования подтверждения операций
2. Отключить метода аутентификации myDSS
3. Удалить ключ аутентификации пользователя

### Отключение требований подтверждения операций

#### Примечание

Отключение методов аутентификации требуется, если myDSS является единственным способом вторичной аутентификации. Если пользователю назначены другие способы аутентификации (например, одноразовые пароли по SMS), то отключение методов не требуется.

#### Примечание

Если отключить метод аутентификации myDSS, не отключив требование подтверждения операций операций, то

пользователь не сможет выполнить данные операции.

Пример запроса

```
POST https://192.168.109.149/STS/ums/user/206adc4f-3262-469c-9871-b7a7cabaa979/operationpolicy HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: 192.168.109.149
Content-Length: 2
Expect: 100-continue

[]
```

Пример ответа

Метод не имеет возвращаемого значения

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Content-Length: 0
```

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404	user_not_found	Пользователь не найден.

Отключение метода аутентификации

Пример запроса

```
DELETE https://host/STS/ums/user/e06a4bc1-4c31-4f8d-a7a7-920b66ca4ad6/authmethod/mobileauth HTTP/1.1
Accept: application/json
Host: host
Content-Length: 0
```

Пример ответа

Метод не имеет возвращаемого значения

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Content-Length: 0
```

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404	user_not_found	Пользователь не найден.

Удаление ключа аутентификации

Пример запроса

```
DELETE https://host/STS/ums/user/a17efd43-181a-45b7-8d60-088b6889480c/mobileauth HTTP/1.1
Accept: application/json
Host: host
Content-Length: 0
```

**Пример ответа**

Методу не имеет возвращаемого значения.

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Content-Length: 0
```

**Типовые ошибки**

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404	user_not_found	Пользователь не найден.
400	wrong_operation	Нельзя удалить ключ аутентификации пользователя не отключив метод аутентификации myDSS

**Задание отпечатка устройства**

Отпечаток устройство однозначно связывает ключ аутентификации и мобильное устройство пользователя. В типовом сценарии отпечаток устройство регистрируется автоматически в момент сканирования QR-кода с ключом аутентификации.

Требуется ли привязка ключа аутентификации к устройству пользователя задаётся Администратором на сервере DSS:

```
Set-DssMobileAuthProperties -DeviceFingerprintRequired 1
```

**Пример запроса**

```
POST https://host/STS/ums/user/78d9d77b-6b76-4045-a8e0-579e70fba468/mobileauth/thumbprint HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 8
Expect: 100-continue

"abcdef"
```

**Пример ответа**

Методу не имеет возвращаемого значения.

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Content-Length: 0
```

**Типовые ошибки**

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404	user_not_found	Пользователь не найден.

# Выпуск сертификата пользователя

Раздел содержит руководство разработчика по выпуску сертификата пользователя. В разделе приведены основные сценарии использования, примеры HTTP-запросов и ответов сервисов DSS.

В разделе приведены сценарии выпуска сертификата:

- [Оператором DSS](#)
- [Пользователем DSS](#)

Перед началом интеграции Администратору DSS необходимо:

- Выпустить и зарегистрировать на DSS сертификат аутентификации Оператора DSS
- Зарегистрировать OAuth-клиента
- Подключить к Сервису Подписи модуль взаимодействия с УЦ

Общий подход для Пользователей и Операторов при выпуске сертификата:

1. Аутентификация на Центре Идентификации
2. Получение параметров выпуска запроса на сертификат
3. Создание запроса на сертификат
4. Подтверждение создания запроса на сертификат (для пользователей)
5. Установка сертификата

## Примечание

В примере рассматривается выпуск сертификата пользователя через Сторонний УЦ. Созданный на шаге 3 запрос на сертификат (PKCS#10) Пользователь или Оператор самостоятельно передать в УЦ для выпуска Сертификата. Выпущенный сертификат должен быть установлен на Сервисе Подписи через API.

## Создание запроса на сертификат

Параметры выпуска запроса на сертификат можно получить из Политики Сервиса Подписи (метод [/policy](#)). Политика Сервиса Подписи содержит:

- Список параметров Удостоверяющих Центров, подключенных к DSS
- Список криптопровайдеров, подключенных к DSS

Каждый элемент списка [параметров УЦ](#) содержит:

- Идентификатор Удостоверяющего Центра
- Тип Удостоверяющего Центра
- Шаблон различительного имени (Distinguished Name)
- Список шаблонов сертификатов
- Отображаемое имя

В интерфейсе интегрируемой системы должна быть возможность выбора Удостоверяющего Центра, для которого будет создан запрос на сертификат. Для каждого Удостоверяющего Центра Сервис Подписи передаёт отображаемое имя (DSSCAPolicy -> `Name`), которое может быть показано пользователю.

Для выбранного пользователем Удостоверяющего Центра в интерфейсе интегрируемой системы должна отображаться форма для заполнения Идентифицирующих данных. Форма составляется в соответствии с шаблоном имени (DSSCAPolicy -> `NamePolicy`). У каждого компонента имени в шаблоне есть отображаемое имя (`Name`), строковый идентификатор (`StringIdentifier`) и требование к заполнению (`IsRequired`).

Так же на форме создания запроса должен быть отображен список шаблонов сертификатов (`EkuTemplates`). Каждый шаблон сертификата имеет отображаемое имя.

Если Политика Сервиса Подписи содержит более одного криптопровайдера, то необходимо предоставить пользователю возможность выбора.

Данные с формы передаются в метод [/requests](#) для создания запроса на сертификат:

- Идентификатор Удостоверяющего Центра
- Различительное имя
- Шаблон сертификата
- ПИН-код на закрытый ключ (опционально)
- Идентификатор криптопровайдера (опционально)

Данные передаются в структуре [CertificateRequest](#).

**Идентификатор Удостоверяющего Центра** (`AuthorityId`) является константой. Он может быть получен от Администратора DSS и зафиксирован в настройках интегрируемой системы.

### Примечание

Если Удостоверяющий Центр с заданным идентификатором отсутствует в Политике Сервиса Подписи, то либо он недоступен в данный момент, либо был отключен Администратором DSS. Для выяснения причин недоступности Удостоверяющего Центра следует обратиться к Администратору DSS.

**Различительное имя** может быть передано в двух форматах:

- Список пар `oid:value` (`DistinguishedName`)
- Строковое представление (`RawDistinguishedName`)

Объектные идентификаторы (**OID**) компонентов имени указаны в шаблоне имени.

### Примечание

Строковое представление различительного имени кодируется согласно [RFC 1779](#).

**Шаблон сертификата** представляет собой набор объектных идентификаторов, которые попадут в расширение Enhanced Key Usage (EKU) запроса на сертификат, или идентификатор шаблона сертификата КриптоПро УЦ 2.0, который попадёт в расширение Certificate Template (1.3.6.1.4.1.311.21.7).

Шаблон передаётся через разные поля запроса на сертификат в зависимости от типа:

- Enhanced key usage - передаётся в дополнительных параметрах запроса `Parameters` в ключе `EkuString` в формате `oid1,oid2,...,oidN`.

### Примечание

Данный шаблон используется при создании запроса на сертификат к Удостоверяющему Центру типа 0 (КриптоПро УЦ 1.5) и 2 (Сторонний УЦ).

- Certificate Template - передаётся в параметре `Template` запроса на сертификат.

### Примечание

Данный шаблон используется при создании запроса на сертификат к Удостоверяющему Центру типа 1 (КриптоПро УЦ 2.0) и 2 (Сторонний УЦ).

**Идентификатор криптопровайдера** должен быть задан, если в Политике Сервиса Подписи доступно более одного криптопровайдера. Идентификатор криптопровайдера (DSSCSPPolicy -> `GroupId`) передаётся в дополнительных параметрах запроса в ключе `GroupId`

Создание запроса на сертификат с подтверждением при помощи вторичной аутентификации

При создании запроса на сертификат с подтверждением при помощи вторичной аутентификации требуется выполнить следующую последовательность действий (шагов):

- [Создание транзакции на Сервисе Подписи](#)
- [Подтверждение транзакции на Сервисе Подтверждения Операций](#)
- [Получение результата операции на Сервисе Подписи](#)

При этом в массив параметров транзакции метода [/transactions](#) должны быть отображены следующие поля [запроса на сертификат](#):

CERTIFICATE REQUEST	ПАРАМЕТРЫ ТРАНЗАКЦИИ
AuthorityId	CAId
PinCode	Не используется
Template	CertTemplateOid
DistinguishedName	Не используется
RawDistinguishedName	CertSubjectName
Parameters -> EkuString	EkuString
Parameters -> GroupId	GroupId

**Примечание**

При создании запроса на сертификат с подтверждением с подтверждением при помощи вторичной аутентификации различительное имя может быть передано только в строковом представлении.

Примеры запросов

Пример запроса с указанием различительного имени в строковом представлении:

```
POST https://host/SignServer/rest/api/requests HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cGEiOiJ1bmRlciIsImVudCI6IjE5MDYyMjE0In0=
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 153
Expect: 100-continue

{
 "AuthorityId":11,
 "PinCode":"","
 "RawDistinguishedName":"CN=dssUser,C=RU",
 "Parameters":
 {"EkuString":"1.2.643.2.2.34.2,1.2.643.2.2.34.4,1.3.6.1.5.5.7.3.2"}
}
```

Пример запроса с указанием различительного имени в виде набора компонентов:

```

POST https://host/SignServer/rest/api/requests HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJ... PhYmXscTmwGkD8b1SWy0nYQ
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 169
Expect: 100-continue

{
 "AuthorityId":11,
 "PinCode":"","
 "DistinguishedName":{"2.5.4.3":"dssUser","2.5.4.6":"RU"},
 "Parameters":
 {
 "EkuString":"1.2.643.2.2.34.2,1.2.643.2.2.34.4,1.3.6.1.5.5.7.3.2"
 }
}

```

Пример запроса с указанием шаблона сертификата:

```

POST https://host/SignServer/rest/api/requests HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV... Ysj1GpIVmR2hw
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 135
Expect: 100-continue

{
 "AuthorityId":11,
 "PinCode":"","
 "Template":"1.3.6.1.5.5.7.3.2",
 "DistinguishedName":{"2.5.4.3":"dssUser","2.5.4.6":"RU"},
 "Parameters":{}}

```

Пример ответа:

```

HTTP/1.1 200 OK
Content-Length: 723
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
Date: Tue, 04 Sep 2018 13:35:02 GMT

{
 "CertificateType":"ServerSide",
 "Base64Request":"MIIBQDCB8AIBADAFMQswCQYD... iOibLabDHZ2VY1G8CsaxjE",
 "CertificateAuthorityID":11,
 "CADisplayName":null,
 "DistName":"CN=dssUser, C=RU",
 "Subject":"dssUser",
 "Status":"PENDING",
 "ID":22,
 "CARequestID":null,
 "CertificateID":0,
 "RequestType":"Certificate",
 "GroupID":"e8e67f9e-7eed-4116-ad98-20582e4d766e"}

```

Запрос на сертификат **с подтверждением** с подтверждением при помощи вторичной аутентификации:



```
POST https://host/SignServer/rest/api/transactions HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cGEiOiJ1bmRlciJ9.pz4erYJpgoN_RgQLA
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 200
Expect: 100-continue

{
 "OperationCode":16,
 "Parameters":
 [
 { "Name":"CertSubjectName", "Value":"CN=dssUser,C=RU"},
 { "Name":"CAId", "Value":"11"},
 { "Name":"EkuString", "Value":"1.2.643.2.2.34.2,1.2.643.2.2.34.4,1.3.6.1.5.5.7.3.2"}
]
}
```

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	pending_requests_exist	У пользователя есть необработанный запрос на сертификат (статус PENDING).

Обработка ответа Сервиса Подписи

При успешном создании запроса на сертификат Сервис Подписи в ответе вернёт структуру [DSSCertRequest](#).

Дальнейшее поведение пользователя зависит от значения поля `Status` в структуре [DSSCertRequest](#) и типа УЦ, на котором создавался запрос на сертификат.

**ACCEPTED** - запрос на сертификат принят и обработан УЦ. В данном случае в поле `CertificateID` будет записан идентификатор выпущенного сертификата.

**REGISTRATION** - запрос на сертификат принят в КриптоПро УЦ 2.0 и находится на этапе регистрации пользователя УЦ. В зависимости от настроек подключения DSS к КриптоПро УЦ 2.0, необходимо:

- ожидать одобрения запроса на сертификат Администратором УЦ;
- одобрить запрос Оператором DSS.

**PENDING** - запрос на сертификат находится в обработке. Если запрос отправлен на КриптоПро УЦ 2.0, то в зависимости от настроек подключения DSS к КриптоПро УЦ 2.0 необходимо:

- ожидать одобрения запроса на сертификат Администратором УЦ;
- одобрить запрос Оператором DSS.

Если запрос создавался через "Сторонний Удостоверяющий Центр", необходимо:

- скачать запрос на сертификат по идентификатору [/requests](#);
- передать запроса на сертификат в УЦ;
- выпущенный сертификат установить в DSS.

Запрос на сертификат (PKCS#10) в формате Base64 содержится в поле `Base64Request` структуры [DSSCertRequest](#).

**REJECTED** - запрос отклонён. Дальнейшая обработка запроса невозможна. Для выяснения причин отклонения запроса необходимо обратиться к Администратору УЦ.

Выпуск сертификата Оператором DSS

Аутентификация Оператора DSS на Центре Идентификации

Аутентификация Оператора DSS производится по сертификату (HTTPS с аутентификацией клиента).

Для получения AccessToken используется OAuth сценарий с использованием кода авторизации. Подробная информация по протоколу аутентификации: [The OAuth 2.0 Authorization Framework](#)

Администратор DSS должен предварительно настроить OAuth клиента на сервере DSS:

- создав нового клиента:

```
Add-DssClient -Identifier testClient -Name testClient -Description "Test Client Description" -RedirectUri urn:ietf:wg:oauth:2.0:oob:auto -AllowedFlow ResourceOwner,AuthorizationCode
```

- изменив настройки существующего клиента:

```
Set-DssClient -ClientId testClient -RedirectUri urn:ietf:wg:oauth:2.0:oob:auto -AllowedFlow ResourceOwner,AuthorizationCode
```

## Примечание

Значение `RedirectUri` **urn:ietf:wg:oauth:2.0:oob:auto** говорит серверу DSS о том, что AccessToken необходимо вернуть непосредственно в ответе на запрос клиента. Данное значение используется в тех случаях, когда для клиента трудозатратно открыть слушателя на другом URL.

Последовательность шагов:

1. [Инициация аутентификации](#), путём отправки запроса на конечную точку /authorize/certificate по HTTPS с аутентификацией по сертификату.
2. [Получение кода авторизации](#).
3. [Получение AccessToken по коду авторизации](#).
4. [Получение делегирующего AccessToken](#).

AccessToken, полученный на шаге 3, позволит Оператору DSS получить [Политику Сервиса Подписи](#).

Для выполнения действий от имени пользователя на Сервисе Подписи необходимо получить [делегирющий AccessToken](#). Делегирующий AccessToken позволит Оператору DSS выпустить сертификат пользователя, просмотреть список сертификатов и запросов пользователя и т.п.

## Инициация аутентификации

Конечная точка для аутентификации Оператора DSS: `/authorize/certificate`

Параметры запроса:

- `redirect_uri` – зарегистрированный на Центре Идентификации адрес возврата (по этому адресу будет возвращён запрошенный код авторизации). Допустимые значения данного параметра сохраняются в ЦИ на этапе регистрации клиента.
- `response_type` – в данном сценарии всегда должен иметь значение **code**.
- `scope` – области использования маркера. Должен содержать значение **dss**.
- `resource` – идентификатор Сервиса Подписи.

## Примеры запросов

```
GET https://host/STS/oauth/authorize/certificate?
client_id=testClient&response_type=code&scope=dss&redirect_uri=urn:ietf:wg:oauth:2.0:oob:auto&resource=urn:cr
uptopro:dss:signserver:signserver
Host: host
Connection: Keep-Alive
```

## Получение кода авторизации

В случае успешной аутентификации

- ответ сервера будет иметь статус HTTP 302
- В заголовке `Location` будет содержаться адрес получения AccessToken.

Т.е. в примере используется специальное значение `redirect_uri`, то клиенту необходимо из заголовка `Location` извлечь значение параметра **code**. Значение параметра **code** будет использовано для получения AccessToken на следующем шаге.

### Пример ответа

```
HTTP/1.1 302 Found
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Location: urn:ietf:wg:oauth:2.0:oob:auto?code=65e4322a9751cf9ba43012692ce02ec1
Date: Fri, 07 Sep 2018 10:30:24 GMT
Content-Length: 0
```

### Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_client	OAuth-клиент не зарегистрирован или неверно указан clientID
400	unauthorized_client	OAuth-клиент использует незарегистрированный сценарий аутентификации (Flow)
400	invalid_request	Неверно сформирован параметр resource
500	An error has occurred	1. Проверяющая сторона с идентификатором resource не зарегистрирована.

### Получение AccessToken

Для получения маркера доступа используется конечная точка `oauth/token`.

Параметры запроса:

- `grant_type` - тип разрешения, в данном сценарии равен **authorization\_code**.
- `code` – код авторизации, полученный на предыдущем шаге.
- `resource` – идентификатор Сервиса Подписи.
- `redirect_uri` – зарегистрированный на Центре Идентификации адрес возврата.

В заголовке Authorization HTTP-запроса клиент должен передать идентификатор OAuth-клиента и секрет (если используется): Authorization: Basic Base64(<client\_id>:<secret>)

### Пример запроса

```
POST https://host/STS/oauth/token HTTP/1.1
Authorization: Basic dGVzdENSaWVudDo=
Content-Type: application/x-www-form-urlencoded
Host: host
Content-Length: 180
Expect: 100-continue

grant_type=authorization_code&code=65e4322a9751cf9ba43012692ce02ec1&redirect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aoob%3Aauto&resource=urn%3Acryptopro%3Adss%3Asignserver%3Asignserver
```

В случае успешной аутентификации ответ будет содержать:

- `access_token` - AccessToken, выпущенный Центром Идентификации DSS
- `token_type` - Тип токена

- `expires_in` - Время жизни токена в секундах

Значение параметра `access_token` необходимо будет использовать при обращениях к Сервису Подписи.

## Примечание

Данный `access_token` не даёт право Оператору DSS выполнять операции на Сервисе Подписи от имени пользователей.  
`access_token` может быть использован для получения [Политики Сервиса Подписи](#).

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 2268
Content-Type: application/json; charset=utf-8
Expires: -1
```

```
{"access_token":"eyJ0eXAiOiJKV1Q...LnS1sAundSE1hh3A5n8W7lhPSM4z_VA","expires_in":300,"token_type":"Bearer"}
```

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_client	OAuth-клиент не зарегистрирован или неверно указан clientID
400	unauthorized_client	OAuth-клиент использует незарегистрированный сценарий аутентификации (Flow)
400	invalid_request	Неверно сформирован параметр resource
400	invalid_grant	Невалидный код авторизации.
500	An error has occurred	1. Проверяющая сторона с идентификатором resource не зарегистрирована.

## Получение делегирующего AccessToken

Для получения AccessToken для делегирования используется конечная точка `oauth/token`. Подробная информация по протоколу получения AccessToken для делегирования: [OAuth 2.0 Token Exchange](#).

Параметры запроса:

- `grant_type` - тип разрешения, в данном сценарии равен **urn:ietf:params:oauth:grant-type:token-exchange**.
- `resource` – идентификатор Сервиса Подписи.
- `actor_token` - AccessToken, полученный на предыдущем шаге
- `actor_token_type` – тип маркера доступа, должен иметь значение **urn:ietf:params:oauth:token-type:jwt**.
- `subject_token_type` – тип маркера доступа, должен иметь значение **urn:ietf:params:oauth:token-type:jwt**.
- `subject_token` – неподписанный JWT-токен, содержащий логин управляемого пользователя.

В декодированном виде `subject_token` имеет вид:

```
{
 "alg": "none",
 "typ": "JWT"
}.
{
 "unique_name": "mydss",
 "nbf": 1488312889,
 "exp": 1488316489,
 "iat": 1488312889
}
.
```

Пример кодирования JWT-токена можно посмотреть по [ссылке](#).

Первая часть (до точки) называется header, вторая – payload. Для получения закодированного значения необходимо выполнить следующее преобразование:

```
Base64UrlEncode(UTF8GetBytes(header)) + "." + Base64UrlEncode(UTF8GetBytes(payload)) + "."
```

### Внимание!

Символ "." в конце получившегося значения является обязательным.

В заголовке Authorization HTTP-запроса клиент должен передать идентификатор OAuth-клиента и секрет (если используется): Authorization: Basic Base64(<client\_id>:<secret>)

### Пример запроса

```
POST https://host/STS/oauth/token HTTP/1.1
Authorization: Basic dGVzdENSaWVudDo=
Content-Type: application/x-www-form-urlencoded
Host: host
Content-Length: 2529
Expect: 100-continue

grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Atoken-exchange&actor_token=eyJ0eXAiOiJKV1QiLCJhbGc ...
E1hh3A5n8W7lhPSM4z_VA&actor_token_type=urn%3Aietf%3Aparams%3Aoauth%3Atoken-
type%3Ajwt&subject_token=e30.eyJ1bm1xdWVfbmFtZSI6Im15ZHNzIn0.&subject_token_type=urn%3Aietf%3Aparams%3Aoauth%
3Atoken-type%3Ajwt&resource=urn%3Acryptopro%3Adss%3Asignserver%3Asignserver
```

В случае успешной аутентификации ответ будет содержать:

- `access_token` - делегирующий AccessToken, выпущенный Центром Идентификации DSS
- `token_type` - Тип токена
- `expires_in` - Время жизни токена в секундах

Значение параметра `access_token` необходимо будет использовать при обращениях к Сервису Подписи.

### Пример ответа

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 2475
Content-Type: application/json; charset=utf-8
Expires: -1

{"access_token":"eyJ0eXAiOiJKV1QiLCJhbGc ... h0X-
7aUneD_po8p5uD3nJGQ5VlNHlw4vA", "expires_in":300, "token_type":"Bearer"}
```

### Получение Политики Сервиса Подписи

#### Пример запроса

```
GET https://host/SignServer/rest/api/policy HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJK ... K0ePGIpg
Host: host
```

#### Пример ответа

```
HTTP/1.1 200 OK
Content-Length: 4821
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
```

Date: Mon, 03 Sep 2018 09:21:40 GMT

```
{
 "CAPolicy":
 [{
 "ID":11,
 "Name":"Out of Band",
 "Active":true,
 "AllowUserMode":false,
 "SNChangesEnable":true,
 "NamePolicy":
 [
 {"IsRequired":false,"Order":2,"OID":"1.2.840.113549.1.9.1","Name":"E-Mail","Value":null,"StringIdentifier":"E"},
 {"IsRequired":false,"Order":8,"OID":"1.2.643.3.131.1.1","Name":"ИНН","Value":null,"StringIdentifier":"INN"},
 {"IsRequired":false,"Order":4,"OID":"2.5.4.7","Name":"Населенный пункт","Value":null,"StringIdentifier":"L"},
 {"IsRequired":false,"Order":7,"OID":"1.2.643.100.1","Name":"ОГРН","Value":null,"StringIdentifier":"OGRN"},
 {"IsRequired":false,"Order":5,"OID":"2.5.4.10","Name":"Организация","Value":null,"StringIdentifier":"O"},
 {"IsRequired":false,"Order":6,"OID":"2.5.4.11","Name":"Подразделение","Value":null,"StringIdentifier":"OU"},
 {"IsRequired":false,"Order":3,"OID":"2.5.4.8","Name":"Регион","Value":null,"StringIdentifier":"S"},
 {"IsRequired":false,"Order":9,"OID":"2.5.4.6","Name":"Страна","Value":null,"StringIdentifier":"C"},
 {"IsRequired":true,"Order":1,"OID":"2.5.4.3","Name":"Общее имя","Value":null,"StringIdentifier":"CN"}
],
 "EKUTemplates":
 {
 "Временный сертификат администратора УЦ":["1.2.643.2.2.34.2","1.2.643.2.2.34.4","1.3.6.1.5.5.7.3.2"],
 "Временный сертификат оператора УЦ":["1.2.643.2.2.34.2","1.2.643.2.2.34.5","1.3.6.1.5.5.7.3.2"],
 "Временный сертификат пользователя УЦ":["1.2.643.2.2.34.2","1.3.6.1.5.5.7.3.2"],
 "Временный сертификат пользователя УЦ1":["1.2.643.2.2.34.6","1.3.6.1.5.5.7.3.2"],
 "Сертификат пользователя УЦ":["1.2.643.2.2.34.6","1.3.6.1.5.5.7.3.2"]
 },
 "CAType":"DSSOutOfBandEnroll",
 "ValidationMode":"CertificateAuthority"
 }],
 "CSPsPolicy":
 [
 {
 "ID":"67e6f39d-c6c5-4ce6-9535-4e22bae84786",
 "GroupID":"e8e67f9e-7eed-4116-ad98-20582e4d766e",
 "TypeID":2,
 "ProviderName":"Crypto-Pro GOST R 34.10-2001 Cryptographic Service Provider",
 "ProviderType":75,"KeyLength":512,"HashAlgorithms":["GOST R 34.11-94"],
 "Description":"GOST 2001"
 },
 {
 "ID":"60e9912c-68a8-4608-b1c2-0c6a074456e8",
 "GroupID":"648092d3-46a9-422a-9240-d32c58cc498b",
 "TypeID":2,
 "ProviderName":"Crypto-Pro GOST R 34.10-2012 Cryptographic Service Provider",
 "ProviderType":80,
 "KeyLength":512,
 "HashAlgorithms":["GR 34.11-2012 256"],
 "Description":"GOST 2012"
 }
],
 "ActionPolicy":[{"DisplayName":"Выпуск маркера (вход в ЦИ)","Uri":"http://dss.cryptopro.ru/identity/claims/action/Issue","Action":"Issue","MfaRequired":false},
```

```
{
 "DisplayName": "Подпись документа",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/SignDocument",
 "Action": "SignDocument",
 "MfaRequired": true
}, {
 "DisplayName": "Подпись пакета документов",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/SignDocuments",
 "Action": "SignDocuments",
 "MfaRequired": false
}, {
 "DisplayName": "Расшифрование документа",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/DecryptDocument",
 "Action": "DecryptDocument",
 "MfaRequired": false
}, {
 "DisplayName": "Создание запроса на сертификат",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/CreateRequest",
 "Action": "CreateRequest",
 "MfaRequired": false
}, {
 "DisplayName": "Смена пин-кода закрытого ключа",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/ChangePin",
 "Action": "ChangePin",
 "MfaRequired": false
}, {
 "DisplayName": "Обновление сертификата",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/RenewCertificate",
 "Action": "RenewCertificate",
 "MfaRequired": false
}, {
 "DisplayName": "Отзыв сертификата",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/RevokeCertificate",
 "Action": "RevokeCertificate",
 "MfaRequired": false
}, {
 "DisplayName": "Приостановление действия сертификата",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/HoldCertificate",
 "Action": "HoldCertificate",
 "MfaRequired": false
}, {
 "DisplayName": "Возобновление действия сертификата",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/UnholdCertificate",
 "Action": "UnholdCertificate",
 "MfaRequired": false
}, {
 "DisplayName": "Удаление сертификата",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/DeleteCertificate",
 "Action": "DeleteCertificate",
 "MfaRequired": false
}, {
 "DisplayName": "Доступ к закрытому ключу",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/PrivateKeyAccess",
 "Action": "PrivateKeyAccess",
 "MfaRequired": false
}],
"PinCodeMode": "Allow",
"TspServices": [
 {
 "Name": "TestTSP",
 "Title": "TestTSP",
 "Url": "http://TEST-DSS-W8R2/TSP/tsp.srf"
 }
],
"TransactionConfirmation": "NotSet",
"AllowedSignatureTypes": ["GOST3410", "CMS", "CAES", "XMLDSig", "MSOffice", "PDF"]
}
```

## Создание запроса на сертификат

### Пример запроса

Согласно параметрам УЦ из Политики Сервиса Подписи Оператор формирует запроса на сертификат.

```
POST https://host/SignServer/rest/api/requests HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1 .. so0AiQOhtllgg
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 167
Expect: 100-continue

{"AuthorityId":11,"PinCode":"","DistinguishedName":{"2.5.4.3":"mydss","2.5.4.6":"RU"},"Parameters":{"EkuString":"1.2.643.2.2.34.2,1.2.643.2.2.34.4,1.3.6.1.5.5.7.3.2"}}
```

### Пример ответа

```
HTTP/1.1 200 OK
Content-Length: 719
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
```

```
{
 "CertificateType": "ServerSide",
 "Base64Request": "MIIIBPjCB7gIBADAdMQswCQYD ... 9MmKj3pHKCuhwuZCfzU+gKLuzWrQ==",
 "CertificateAuthorityID": 11,
 "CADisplayName": null,
 "DistName": "CN=mydss, C=RU",
 "Subject": "mydss",
 "Status": "PENDING",
 "ID": 23,
 "CARequestID": null,
 "CertificateID": 0,
 "RequestType": "Certificate",
 "GroupID": "e8e67f9e-7eed-4116-ad98-20582e4d766e"
}
```

## Установка сертификата

Сервер вернул запрос на сертификат в поле `Base64Request`. Запрос на сертификат необходимо передать в Удостоверяющий Центр для выпуска сертификата.

## Пример запроса

```
POST https://host/SignServer/rest/api/certificates HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIU2 ... PYErWVsOP9IM7oahdJ3iRg
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 1154
Expect: 100-continue

{"Certificate": "MIIDHTCCAsyAwIBAgITEgAs1guTuGxFdbofF ... qJNHeSr2QvryU0IPn1jRE/VZnuEMf80PZ1\r\n"}
```

## Пример ответа



HTTP/1.1 200 OK  
Content-Length: 1491  
Content-Type: application/json; charset=utf-8  
Server: Microsoft-IIS/7.5

```
{
 "CertificateType": "ServerSide",
 "ID": 14,
 "DName": "CN=mydss, C=RU",
 "CertificateBase64": "MIIDHTCCAsygAwIBAgITEgAs ... NHeSr2QvryU0IPn1jRE/VZnuEMf80PZ1",
 "Status": {
 "Value": "ACTIVE",
 "RevocationInfo": null,
 "PinCode": null,
 "ActiveCertId": 0
 },
 "IsDefault": false,
 "CertificateAuthorityID": 11,
 "CspID": "e8e67f9e-7eed-4116-ad98-20582e4d766e",
 "HashAlgorithms": ["GOST R 34.11-94"],
 "ProviderName": null,
 "ProviderType": 0,
 "PrivateKeyNotBefore": null,
 "PrivateKeyNotAfter": null,
 "HasPin": false,
 "FriendlyName": ""
}
```

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_certificate_format	Сертификат имеет неверный формат. Например, сертификат передан с заголовками.

Выпуск сертификата Пользователем DSS

Аутентификация пользователя на Центре Идентификации

В примере рассматривается авторизация с использованием учётных данных пользователя (логин/пароль). Подробная информация по протоколу аутентификации: [The OAuth 2.0 Authorization Framework](#)

Параметры запроса:

- grant\_type - тип разрешения, в данном сценарии равен **password**.
- password – пароль пользователя.
- resource – идентификатор Сервиса Подписи.

В заголовке Authorization HTTP-запроса клиент должен передать идентификатор OAuth-клиента и секрет (если используется): Authorization: Basic Base64(<client\_id>:<secret>)

Примечание

В примере значение параметр **password** оставлено пустым, так как пользователю в качестве первичной аутентификации назначен метод "Только Идентификация"

Примеры запросов

```
POST https://host/STS/oauth/token HTTP/1.1
Authorization: Basic dGVzdENSawVudDo=
Content-Type: application/x-www-form-urlencoded
Host: host
Content-Length: 101
Expect: 100-continue
Connection: Keep-Alive

grant_type=password&username=mydss&password=&resource=urn%3Acryptopro%3Adss%3Asignserver%3Asignserver
```

В случае успешной аутентификации ответ будет содержать:

- `access_token` - AccessToken, выпущенный Центром Идентификации DSS
- `token_type` - Тип токена
- `expires_in` - Время жизни токена в секундах

Значение параметра `access_token` необходимо будет использовать при обращениях к Сервису Подписи и Сервису Подтверждения Операций.

**Пример ответа**

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 2017
Content-Type: application/json; charset=utf-8
Expires: -1

{
 "access_token": "eyJ0eXAiOiJKV...5Wti-H8CeXycwB6A",
 "expires_in": 300,
 "token_type": "Bearer"
}
```

**Типовые ошибки**

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_client	OAuth-клиент не зарегистрирован или неверно указан clientID
400	unauthorized_client	OAuth-клиент использует незарегистрированный сценарий аутентификации (Flow)
400	invalid_request	Неверно сформирован параметр resource
500	An error has occurred	1. Проверяющая сторона с идентификатором resource не зарегистрирована.

**Получение Политики Сервиса Подписи**

**Пример запроса**

```
GET https://host/SignServer/rest/api/policy HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJK...K0ePGIpg
Host: host
```

**Пример ответа**

```
HTTP/1.1 200 OK
Content-Length: 4821
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
```

Date: Mon, 03 Sep 2018 09:21:40 GMT

```
{
 "CAPolicy":
 [{
 "ID":11,
 "Name":"Out of Band",
 "Active":true,
 "AllowUserMode":false,
 "SNChangesEnable":true,
 "NamePolicy":
 [
 {"IsRequired":false,"Order":2,"OID":"1.2.840.113549.1.9.1","Name":"E-Mail","Value":null,"StringIdentifier":"E"},
 {"IsRequired":false,"Order":8,"OID":"1.2.643.3.131.1.1","Name":"ИНН","Value":null,"StringIdentifier":"INN"},
 {"IsRequired":false,"Order":4,"OID":"2.5.4.7","Name":"Населенный пункт","Value":null,"StringIdentifier":"L"},
 {"IsRequired":false,"Order":7,"OID":"1.2.643.100.1","Name":"ОГРН","Value":null,"StringIdentifier":"OGRN"},
 {"IsRequired":false,"Order":5,"OID":"2.5.4.10","Name":"Организация","Value":null,"StringIdentifier":"O"},
 {"IsRequired":false,"Order":6,"OID":"2.5.4.11","Name":"Подразделение","Value":null,"StringIdentifier":"OU"},
 {"IsRequired":false,"Order":3,"OID":"2.5.4.8","Name":"Регион","Value":null,"StringIdentifier":"S"},
 {"IsRequired":false,"Order":9,"OID":"2.5.4.6","Name":"Страна","Value":null,"StringIdentifier":"C"},
 {"IsRequired":true,"Order":1,"OID":"2.5.4.3","Name":"Общее имя","Value":null,"StringIdentifier":"CN"}
],
 "EKUTemplates":
 {
 "Временный сертификат администратора УЦ":["1.2.643.2.2.34.2","1.2.643.2.2.34.4","1.3.6.1.5.5.7.3.2"],
 "Временный сертификат оператора УЦ":["1.2.643.2.2.34.2","1.2.643.2.2.34.5","1.3.6.1.5.5.7.3.2"],
 "Временный сертификат пользователя УЦ":["1.2.643.2.2.34.2","1.3.6.1.5.5.7.3.2"],
 "Временный сертификат пользователя УЦ1":["1.2.643.2.2.34.6","1.3.6.1.5.5.7.3.2"],
 "Сертификат пользователя УЦ":["1.2.643.2.2.34.6","1.3.6.1.5.5.7.3.2"]
 },
 "CAType":"DSSOutOfBandEnroll",
 "ValidationMode":"CertificateAuthority"
 }],
 "CSPsPolicy":
 [
 {
 "ID":"67e6f39d-c6c5-4ce6-9535-4e22bae84786",
 "GroupID":"e8e67f9e-7eed-4116-ad98-20582e4d766e",
 "TypeID":2,
 "ProviderName":"Crypto-Pro GOST R 34.10-2001 Cryptographic Service Provider",
 "ProviderType":75,"KeyLength":512,"HashAlgorithms":["GOST R 34.11-94"],
 "Description":"GOST 2001"
 },
 {
 "ID":"60e9912c-68a8-4608-b1c2-0c6a074456e8",
 "GroupID":"648092d3-46a9-422a-9240-d32c58cc498b",
 "TypeID":2,
 "ProviderName":"Crypto-Pro GOST R 34.10-2012 Cryptographic Service Provider",
 "ProviderType":80,
 "KeyLength":512,
 "HashAlgorithms":["GR 34.11-2012 256"],
 "Description":"GOST 2012"
 }
],
 "ActionPolicy":[{"DisplayName":"Выпуск маркера (вход в ЦИ)","Uri":"http://dss.cryptopro.ru/identity/claims/action/Issue","Action":"Issue","MfaRequired":false},
```

```
{
 "DisplayName": "Подпись документа",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/SignDocument",
 "Action": "SignDocument",
 "MfaRequired": true
}, {
 "DisplayName": "Подпись пакета документов",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/SignDocuments",
 "Action": "SignDocuments",
 "MfaRequired": false
}, {
 "DisplayName": "Расшифрование документа",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/DecryptDocument",
 "Action": "DecryptDocument",
 "MfaRequired": false
}, {
 "DisplayName": "Создание запроса на сертификат",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/CreateRequest",
 "Action": "CreateRequest",
 "MfaRequired": false
}, {
 "DisplayName": "Смена пин-кода закрытого ключа",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/ChangePin",
 "Action": "ChangePin",
 "MfaRequired": false
}, {
 "DisplayName": "Обновление сертификата",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/RenewCertificate",
 "Action": "RenewCertificate",
 "MfaRequired": false
}, {
 "DisplayName": "Отзыв сертификата",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/RevokeCertificate",
 "Action": "RevokeCertificate",
 "MfaRequired": false
}, {
 "DisplayName": "Приостановление действия сертификата",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/HoldCertificate",
 "Action": "HoldCertificate",
 "MfaRequired": false
}, {
 "DisplayName": "Возобновление действия сертификата",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/UnholdCertificate",
 "Action": "UnholdCertificate",
 "MfaRequired": false
}, {
 "DisplayName": "Удаление сертификата",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/DeleteCertificate",
 "Action": "DeleteCertificate",
 "MfaRequired": false
}, {
 "DisplayName": "Доступ к закрытому ключу",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/PrivateKeyAccess",
 "Action": "PrivateKeyAccess",
 "MfaRequired": false
}],
"PinCodeMode": "Allow",
"TspServices": [
 {
 "Name": "TestTSP",
 "Title": "TestTSP",
 "Url": "http://TEST-DSS-W8R2/TSP/tsp.srf"
 }
],
"TransactionConfirmation": "NotSet",
"AllowedSignatureTypes": [
 "GOST3410",
 "CMS",
 "CADES",
 "XMLDSig",
 "MSOffice",
 "PDF"
]
}
```

# Подтверждение операций с помощью myDSS

Раздел содержит руководство разработчика по подтверждению (отклонению) операций с помощью myDSS на примере подтверждения операции подписи. В разделе приведены основные сценарии использования, примеры HTTP-запросов и ответов REST-сервисов DSS.

Сценарии должны выполняться Пользователем DSS.

myDSS поддерживает два сценария подтверждения (отклонения) операций:

**Online** - мобильное устройство пользователя имеет выход в интернет. Пользователю придёт Push-уведомление о необходимости подтвердить операцию. Мобильное приложение myDSS загрузит с сервера сведения об операции (сопровождающий текст и подписываемый документ). Пользователю необходимо ознакомиться с подписываемыми данными и выразить своё согласие (отказ) на подписание документа, нажав кнопку "Подтвердить" ("Отказаться") в мобильном приложении.

**Offline** - мобильное устройство пользователя **не имеет** выхода в Интернет. В данном сценарии пользователю необходимо отобразить QR-код, содержащие сведения о подтверждаемой операции. После считывания QR-кода, пользователю в мобильном приложении отобразится код подтверждения (отмены), который необходимо будет ввести в интерфейс DSS вручную.

## Внимание!

В Offline сценарии на мобильном устройстве пользователя не может быть отображён подписываемый документ. Отобразить возможно только сопровождающий операцию текст.

Последовательность шагов при подтверждении операции подписи:

1. [Аутентификация пользователя на Центре Идентификации](#)
2. [Создание транзакции подписи на Сервисе Подписи](#)
3. [Подтверждение транзакции подписи на Сервисе Подтверждения Операций](#)
4. [Получение подписанного документа на Сервисе Подписи](#)

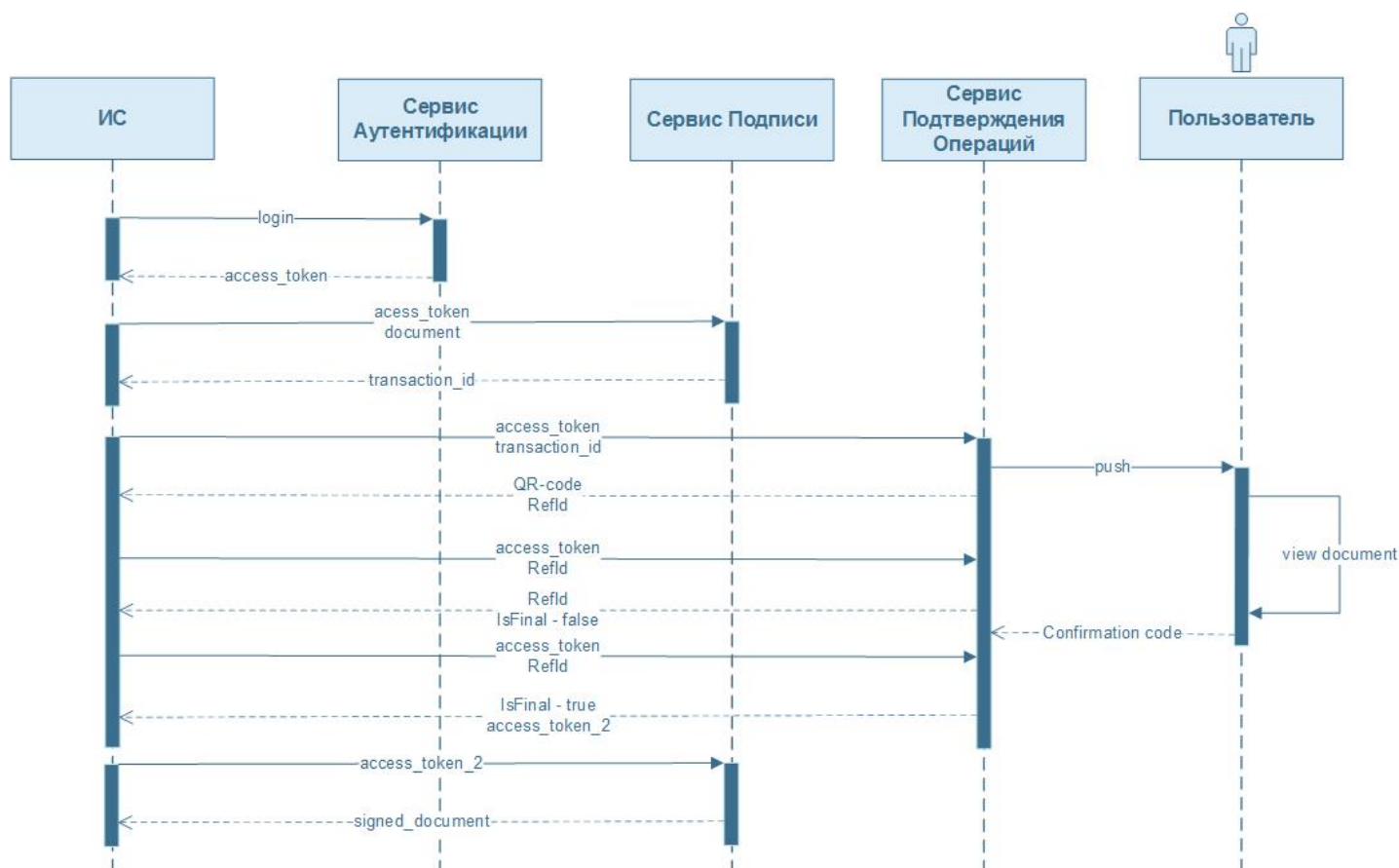
## Примечание

Подтверждение других операций на Сервисе Подписи (создание запроса на сертификат, отзыв сертификата, подпись пакета документов и т.п.) состоит из аналогичной последовательности шагов - отличие в типе транзакции, создаваемой на Сервисе Подписи.

Результатом подтверждения транзакции на Сервисе Подтверждения Операций является AccessToken, содержащий идентификатор подтверждённой транзакции. При подтверждении транзакции на Центре Идентификации у пользователя есть две стратегии поведения:

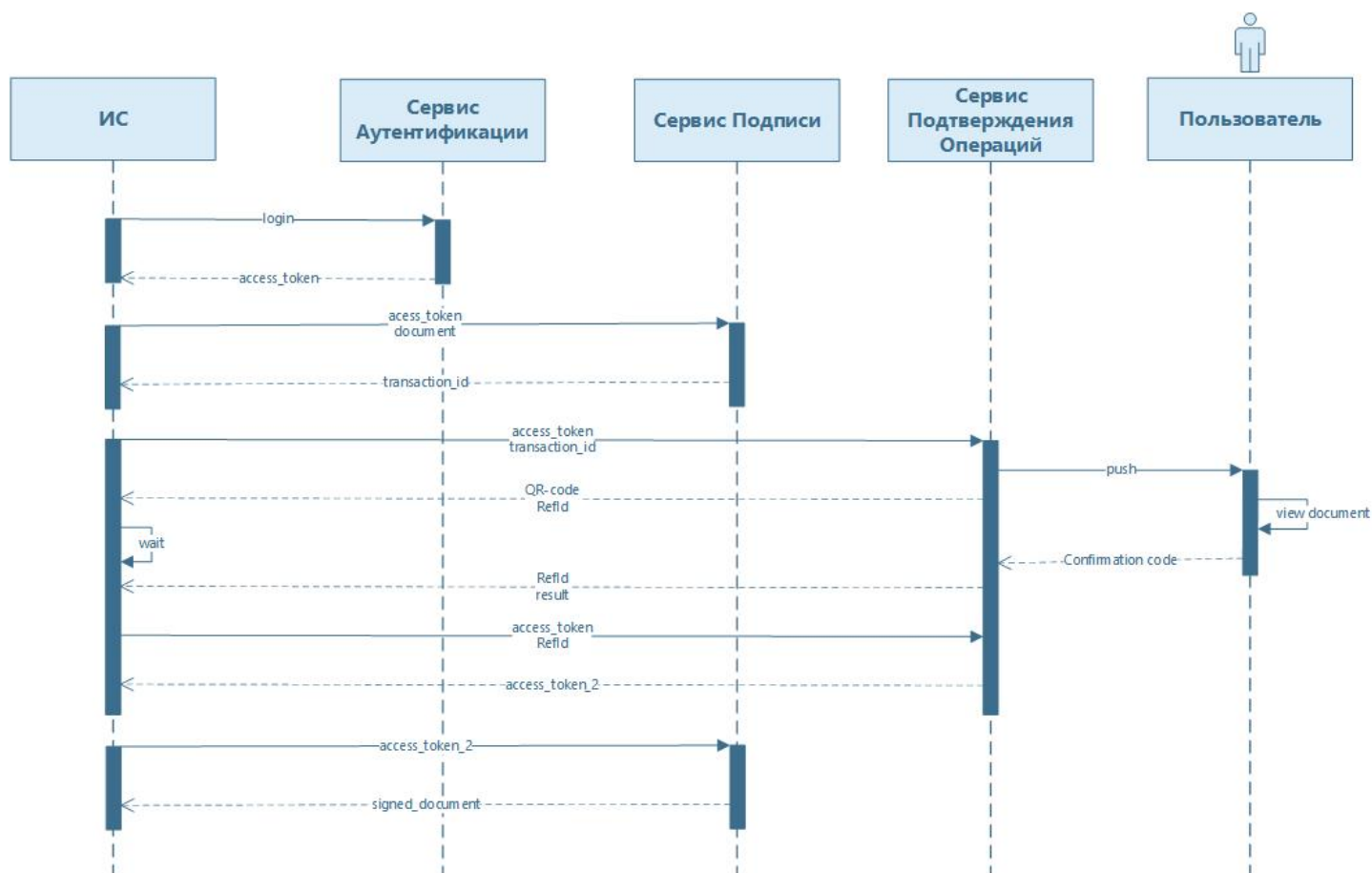
**Синхронная** - пользователь периодически опрашивает конечную точку /confirmation. Если в ответе Сервиса Подтверждения Операций флаг `IsFinal` выставлен в true, то ответ будет содержать перевыпущенный AccessToken. С данным AccessToken пользователь обратится к Сервису Подписи для получения подписанного документа.

Последовательность действий при синхронном-online подтверждении

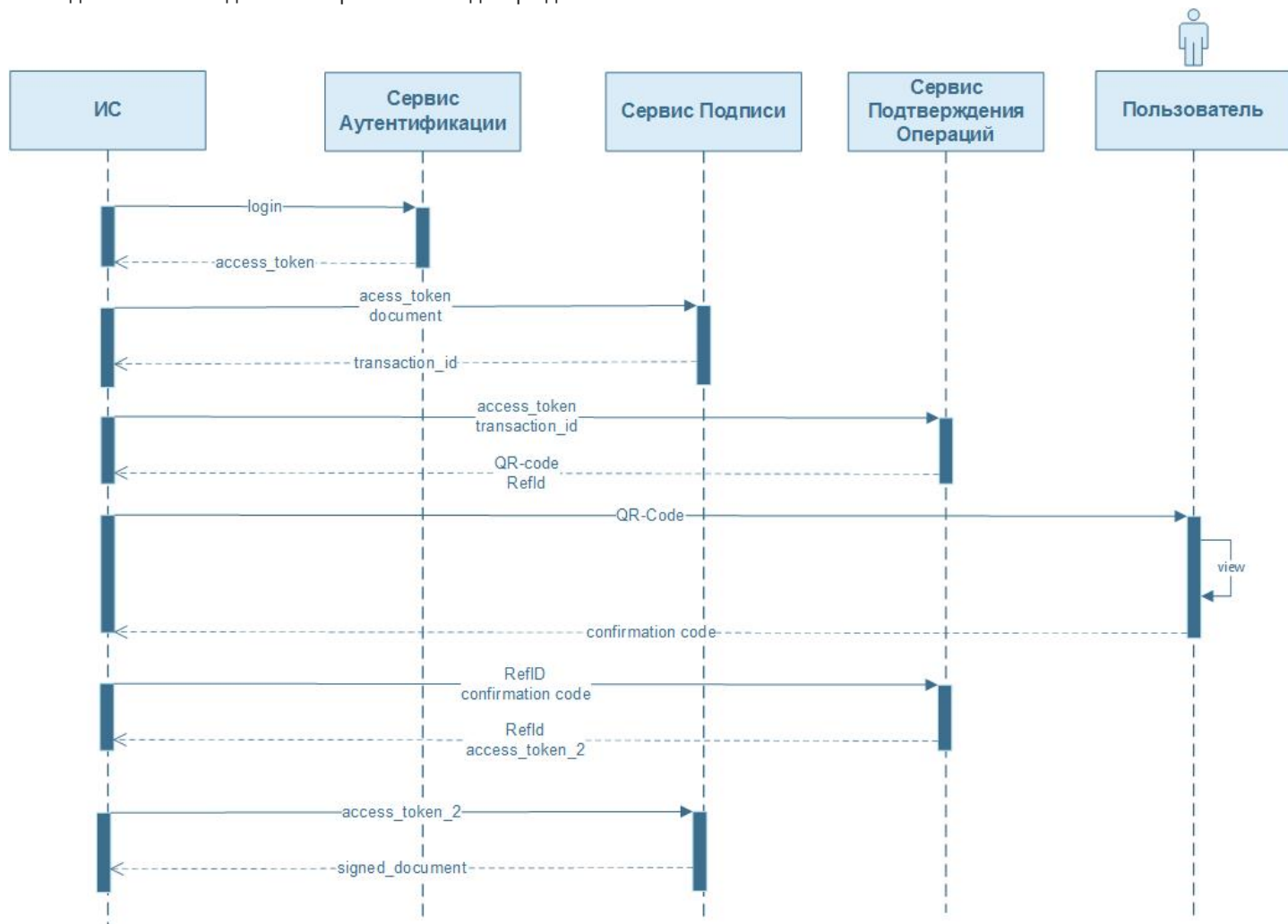


**Асинхронная** - пользователь ожидает оповещения о завершении транзакции на адрес `CallbackUri`, переданный в первом запросе на конечную точку `/confirmation`. После подтверждения (отклонения) транзакции на мобильном устройстве на адрес `CallbackUri`, придет оповещение о завершении транзакции. В случае успешного завершения транзакции пользователь должен повторно обратиться на конечную точку `/confirmation` для получения нового `AccessToken`.

Последовательность действий при асинхронном-online подтверждении



Последовательность действий при Offline подтверждении



Подтверждение операции на Сервисе Подписи

## Предварительные условия

- Пользователю создана учётная запись в DSS;
- [Пользователю назначена аутентификация через myDSS](#);
- Пользователю выпущен сертификат электронной подписи;
- На сервере DSS зарегистрирован OAuth20 клиент.

В подтверждении транзакции задействованы следующие сервисы DSS:

КОНЕЧНАЯ ТОЧКА	СЕРВИС	ОПИСАНИЕ
<code>https://&lt;host&gt;/&lt;StsAppName&gt;/oauth</code>	Сервис Аутентификации.	Аутентификация пользователей для возможности обращений к Сервису Подписи
<code>https://&lt;host&gt;/&lt;SignServerAppName&gt;/rest/api</code>	Сервис Подписи	Создание транзакций и получение результатов, подтвержденной операции
<code>https://&lt;host&gt;/&lt;StsAppName&gt;/confirmation</code>	Сервис Подтверждения Операций	Подтверждение транзакций

### Примечание

У Администратора DSS необходимо получить значение параметров `client_id` и `resource`. `resource` - идентификатор Сервиса Подписи, имеет вид: `urn:cryptopro:dss:signserver:<SignServerAppName>`

### Примечание

Для отображения подписываемого документа в мобильном приложении на Центре Идентификации должны быть настроены плагины преобразования документов - см. раздел [Отображение документов](#)

## Аутентификация пользователя на Центре Идентификации

В примере рассматривается авторизация с использованием учётных данных пользователя (логин/пароль). Подробная информация по протоколу аутентификации: [The OAuth 2.0 Authorization Framework](#)

Параметры запроса:

- `grant_type` - тип разрешения, в данном сценарии равен **password**.
- `password` – пароль пользователя.
- `resource` – идентификатор Сервиса Подписи.

В заголовке Authorization HTTP-запроса клиент должен передать идентификатор OAuth-клиента и секрет (если используется): `Authorization: Basic Base64(<client_id>:<secret>)`

### Примечание

В примере значение параметр **password** оставлено пустым, так как пользователю в качестве первичной аутентификации назначен метод "Только Идентификация"

## Пример запроса



```
POST https://host/STS/oauth/token HTTP/1.1
Authorization: Basic dGVzdENSawVudDo=
Content-Type: application/x-www-form-urlencoded
Host: host
Content-Length: 101
Expect: 100-continue
Connection: Keep-Alive

grant_type=password&username=mydss&password=&resource=urn%3Acryptopro%3Adss%3Asignserver%3Asignserver
```

В случае успешной аутентификации ответ будет содержать:

- `access_token` - AccessToken, выпущенный Центром Идентификации DSS
- `token_type` - Тип токена
- `expires_in` - Время жизни токена в секундах

Значение параметра `access_token` необходимо будет использовать при обращениях к Сервису Подписи и Сервису Подтверждения Операций.

### Пример ответа

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 2017
Content-Type: application/json; charset=utf-8
Expires: -1

{
 "access_token": "eyJ0eXAiOiJKV... 5Wti-H8CeXycwB6A",
 "expires_in": 300,
 "token_type": "Bearer"
}
```

### Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_client	OAuth-клиент не зарегистрирован или неверно указан clientID
400	unauthorized_client	OAuth-клиент использует незарегистрированный сценарий аутентификации (Flow)
400	invalid_request	Неверно сформирован параметр resource
500	An error has occurred	1. Проверяющая сторона с идентификатором resource не зарегистрирована.

### Создание транзакции подписи на Сервисе Подписи

После прохождения аутентификации пользователь инициирует подписание документа. Для подтверждения любых операций на Сервисе Подписи используется метод [/transactions](#) В запросе необходимо указать:

- [OperationCode](#) - тип создаваемой транзакции.
- [Parameters](#) - параметры транзакции.
- Document - подписываемый документ.

В заголовке Authorization HTTP-запроса клиент должен указать AccessToken полученный при аутентификации:  
Authorization: Bearer <access\_token>.

Идентификатор сертификата подписи `CertificateID` можно получить запросив список сертификатов пользователя, обратившись на конечную точку `\certificates`

Параметры создания транзакций других типов приведены [здесь](#)

**Пример запроса**

В примере создаётся прикреплённая CAdES-BES подпись.

```
POST https://host/SignServer/rest/api/transactions HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 355049
Expect: 100-continue

{
 "OperationCode":2,
 "Parameters":
 [
 {"Name":"SignatureType","Value":"CMS"},
 {"Name":"CertificateID","Value":"13"},
 {"Name":"DocumentInfo","Value":"testPdf.pdf"},
 {"Name":"DocumentType","Value":"pdf"},
 {"Name":"IsDetached","Value":"false"},
 {"Name":"CADESType","Value":"BES"}
],
 "Document":"JVBERi0xLjUNCiW1tbW14Kfu"
}
```

**Пример ответа**

Сервис Подписи вернёт идентификатор созданной транзакции. Далее пользователю требуется подтвердить транзакцию на Сервисе Подтверждения Операций.

```
HTTP/1.1 200 OK
Content-Length: 38
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5

"d5ebd393-e093-4aa8-bdcf-f5e497dc6b4d"
```

**Типовые ошибки**

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_certificate	Неверный идентификатор сертификата
400	invalid_request	Неверно указаны параметры подписи

**Подтверждение транзакции подписи на Сервисе Подтверждения Операций**

Для подтверждения транзакции, созданной на Сервисе Подписи, пользователь отправляет запрос содержащий:

- `CallbackUri` - адрес для оповещения о завершении транзакции (опционально).
- `TransactionTokenId` – идентификатор транзакции, созданной на сервисе подписи.
- `Resource` – идентификатор Сервиса Подписи.
- `ClientId` - идентификатор OAuth клиента.
- `ClientSecret` - пароль OAuth клиента (для неконфиденциальных клиентов данный параметр не указывается).

В заголовке Authorization HTTP-запроса клиент должен передать токен, полученный на первом шаге: Authorization: Bearer <access\_token>.

Параметр CallbackUri - опциональный, используется в асинхронном сценарии подтверждения транзакции.

Примеры запросов

```
POST https://host/STS/confirmation HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIHM ... 5aPB98A3NAVduJbtz5Wti-H8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 246
Expect: 100-continue

{
 "Resource": "urn:cryptopro:dss:signserver:signserver",
 "ClientId": "oauth-client-id",
 "ClientSecret": "oauth-client-secret",
 "TransactionTokenId": "d5ebd393-e093-4aa8-bdcf-f5e497dc6b4d",
 "CallbackUri": "http://clienthost/callback/"
}
```

При получении запроса Сервис Подтверждения Операций и сервис myDSS начнут процедуру подтверждения операции в мобильном приложении. В частности отправят Push-уведомление пользователю.

Пример ответа

Ответ Сервиса Подтверждения Операций содержит:

ПОЛЕ	ОПИСАНИЕ
Challenge	Запрос на выполнение аутентификационного испытания
AccessToken	Маркер доступа. Заполняется при IsFinal - true
ExpiresIn	Время жизни AccessToken в секундах. Заполняется при IsFinal - true
IsFinal	Является ли данный ответ последним в процессе подтверждения.
IsError	Содержит ли данный ответ ошибку обработки запроса. Заполняется при IsFinal - false
Error	Ошибка обработки запроса. Заполняется при IsFinal - false
ErrorDescription	Подробное описание ошибки обработки запроса

Поле Challenge содержит:

ПОЛЕ	ОПИСАНИЕ
Title	Текст, который вызывающая система может отобразить пользователю в своём интерфейсе
TextChallenge	Дополнительные данные для подтверждения операции

В поле `TextChallenge` содержится:

ПОЛЕ	ОПИСАНИЕ
Image	QR-код для Offline подтверждения операции
RefID	Идентификатор транзакции, созданной на Сервисе Подтверждения Операций
ExpiresIn	Срок действия транзакции, созданной на Сервисе Подтверждения Операций
AuthnMethod	Идентификатор метода используемый для подтверждения транзакции

**Примечание**

`RefId` - Идентификатор транзакции, созданной на Сервисе Подтверждения Операций. Идентификатор необходимо будет использовать при последующих обращениях на конечную точку `/confirmation`.

**Примечание**

При обработке ответа Сервиса Подтверждения Операций вызывающее приложение должно смотреть на значение двух флагов: `IsFinal` и `IsError`.

Если получен ответ с `IsError` - true, то дальнейшее подтверждение транзакции не возможно.

Если получен ответ с `IsFinal` - false, то подтверждение транзакции ещё не завершено.

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 6736
Content-Type: application/json; charset=utf-8
Expires: -1

{
 "Challenge":
 {
 "Title":
 {
 "Value": "Подтвердите операцию на устройстве с помощью приложения."
 },
 "TextChallenge":
 [
 {
 "Image":
 {
 "MimeType": "image/gif",
 "Value": "R0lGODlhLAESAfcaAAAAAAAAAAmw ... AAziO77kw77me77om77qGxcBAQA7"
 },
 "AuthnMethod": "http://dss.cryptopro.ru/identity/authenticationmethod/mobile",
 "RefID": "e7207ff7-5456-4943-bebf-a7cc624aadaa",
 "ExpiresIn": 300,
 "ExpiresInSpecified": true
 }
],
 "ContextData":
 { "RefID": "e7207ff7-5456-4943-bebf-a7cc624aadaa" }
 },
 "IsFinal": false,
 "IsError": false
}
```

Дальнейшее взаимодействие с Сервисом Подтверждения Операций зависит от выбранного сценария:

- [синхронный](#)
- [асинхронный](#)
- [Offline](#)

#### Асинхронное подтверждение транзакции

Если в первом запросе к Сервису Подтверждения Операций пользователь указал `CallbackUri`, то после подтверждения операции на мобильном устройстве пользователя придёт оповещение о завершении транзакции.

Сообщение о завершении транзакции содержит:

- `Result` - результат подтверждения транзакции (success или failed)
- `TransactionId` - идентификатор транзакции на Сервисе Подтверждения операций (`RefId`)
- `Error` - код ошибки
- `ErrorDescription` - описание ошибки

#### Примеры ответа на `CallbackUri`

Оповещение о подтверждении операции:

```
{
 "Result": "success",
 "TransactionId": "aa1a4a5d-bb4d-456b-87da-31818604fcd8",
 "Error": "",
 "ErrorDescription": null}
```

Оповещение об отказе (пользователь в мобильном приложении Отказался от подтверждения операции):

```
{
 "Result": "failed",
 "TransactionId": "2fbd0a40-77be-4a40-a688-a0249bba16a6",
 "Error": null,
 "ErrorDescription": null}
```

Оповещение об истечении срока действия транзакции.

```
{
 "Result": "failed",
 "TransactionId": "bc0ffdee-7143-439f-bf6b-d1400725d8f1",
 "Error": "transaction_expired",
 "ErrorDescription": "Срок действия транзакции истёк"}
```

Если пользователь подтвердил операцию на мобильном устройстве, необходимо обратиться на Сервис Подтверждения Операций для получения нового `AccessToken`. В запросе передаётся идентификатор `RefId`.

#### Пример запроса

```
{
 "Resource" : "urn:cryptopro:dss:signserver:SignServer",
 "ClientId": "oauth-client-id",
 "ClientSecret": "oauth-client-secret",
 "ChallengeResponse":
 {
 "TextChallengeResponse":
 [{"RefId": "e7207ff7-5456-4943-bebf-a7cc624aadaa"}]
 }
}
```

Ответ Сервиса Подтверждения Операций будет содержать новый AccessToken, который необходимо использовать для получения подписанного документа.

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_transaction	1. Срок действия транзакции истёк 2. Передан неверный идентификатор транзакции (RefId)
400	transaction_pending	У пользователя есть неподтвержденная транзакция.

В синхронном режиме пользователь должен периодически опрашивать Сервис Подтверждения Операция, ожидая завершение подтверждения транзакции (флаг `IsFinal` = true).

## Пример запроса

```
POST https://host/STS/confirmation HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1 ... mXqvC5_3W244A
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 212
Expect: 100-continue

{
 "Resource" : "urn:cryptopro:dss:signserver:SignServer",
 "ClientId": "oauth-client-id",
 "ClientSecret": "oauth-client-secret",
 "ChallengeResponse":
 {
 "TextChallengeResponse":
 [{"RefId": "de34f120-55d5-4f3e-8e7a-b15c1444d747"}]}
}
```

## Примеры ответов

Если подтверждение не завершено, то `IsFinal` - false

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 352
Content-Type: application/json; charset=utf-8
Expires: -1

{
 "Challenge":
 {
 "Title": {"Value": ""},
 "TextChallenge":
 [{
 "AuthnMethod": "http://dss.cryptopro.ru/identity/authenticationmethod/mobile",
 "RefID": "de34f120-55d5-4f3e-8e7a-b15c1444d747"
 }],
 "ContextData": {"RefID": "de34f120-55d5-4f3e-8e7a-b15c1444d747"}},
 "IsFinal": false,
 "IsError": false
}
```

Если в ответе `IsFinal` - true, то Сервис вернул новый AccessToken.

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 2215
Content-Type: application/json; charset=utf-8
Expires: -1

{
 "AccessToken": "eyJ0eXAiOiJKV1QiLC ... 5b1T6H1ytuWztMPGfz-0w",
 "ExpiresIn": 600,
 "IsFinal": true,
 "IsError": false
}
```

## Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_transaction	1. Срок действия транзакции истёк 2. Передан неверный идентификатор транзакции (RefId)

#### Offline подтверждение транзакции

Offline сценарий может использоваться как альтернативный способ подтверждения или отклонения транзакции. Сценарий может использоваться когда мобильное приложение не имеет доступа в Интернет, либо по каким либо причинам не смогло загрузить с сервера данные транзакции (сопровождающий текст, подписываемый документ)

Интегрируемая система должна отобразить пользователю QR-код (Image), полученный при первом обращении к Сервису Подтверждения Операций, и предоставить пользователю интерфейс для ручного ввода кода подтверждения (отказа) транзакции.

#### Пример запроса

```
POST https://host/STS/confirmation HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1Q ... mlfprpmS79Xto3KEQ
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 229
Expect: 100-continue

{
 "Resource" : "urn:cryptopro:dss:signserver:SignServer",
 "ClientId": "oauth-client-id",
 "ClientSecret": "oauth-client-secret",
 "ChallengeResponse":
 {
 "TextChallengeResponse":
 [
 {
 "RefId": "ca6d568a-e81c-4a43-a3a2-65841f7213e3",
 "Value": "12..56"
 }
]
 }
}
```

Длина кода подтверждения (отмены) настраивается Администратором на сервере DSS. Минимальная длина кода подтверждения (отмены) - 6 цифр.

```
Set-DssMobileAuthProperties -ConfirmationCodeLength 8
```

#### Пример ответа

Если в ответе 'IsFinal' - true, то Сервис вернул новый AccessToken.

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 2215
Content-Type: application/json; charset=utf-8
Expires: -1

{
 "AccessToken": "eyJ0eXAiOiJKV1QicLC ... 5b1T6H1ytuWztMPGfz-0w",
 "ExpiresIn": 600,
 "IsFinal": true,
 "IsError": false
}
```



Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_transaction	1. Срок действия транзакции истёк 2. Передан неверный идентификатор транзакции (RefId)
400	authentication_failed	Передан неверный код подтверждения (отмены)

Получение подписанного документа на Сервисе Подписи

Для получения подписанного документа необходимо отправить запрос Сервису Подписи на конечную точку /documents.

Примечание

В заголовке Authorization HTTP-запроса клиент должен указать AccessToken полученный от **Сервиса Подтверждения Операций**: Authorization: Bearer <access\_token>.

Примеры запросов

```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiICKM81 ... jQIwoldWtB5_Gw
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 2
Expect: 100-continue

{}
```

Примечание

Если закрытый ключ сертификата защищён на ПИН-коде, то ПИН-код должен быть указан при обращении на конечную точку /documents

Пример запроса с указанием ПИН-кода:

```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiICKM81 ... xBT_myemDbgJoQ
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 97
Expect: 100-continue

{"Signature":{"PinCode":"1234"}}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Length: 356734
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5

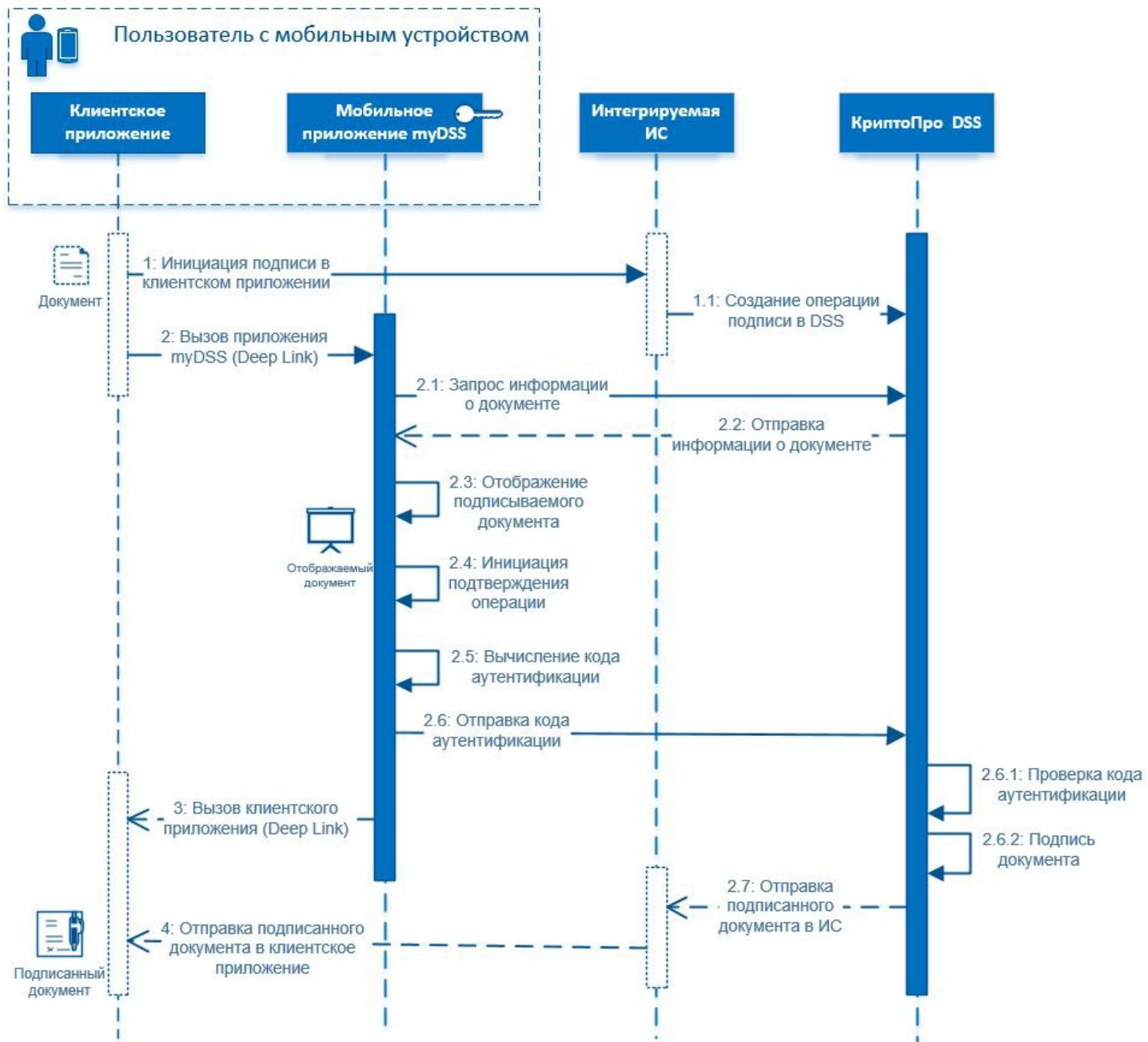
"MIMEFRcG ... gkRSA"
```

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_pin	Неверно указан ПИН-код на закрытый ключ

# Глубинные ссылки (Deep Links)

Мобильное приложение myDSS, позволяющее подтверждать операции подписи в КриптоПро DSS, может быть автоматически открыто (вызвано) непосредственно из клиентского приложения, установленного на мобильном устройстве пользователя. Для этого используются глубинные ссылки (Deep Links). В общем виде схема взаимодействия выглядит следующим образом:



## Примечание

Некоторые блоки ожидания ответа на схеме изображены прозрачными с пунктирным контуром. Эти блоки представлены в общем виде и разделены для примера. В зависимости от реализации взаимодействия клиентского приложения и интегрируемой ИС с myDSS эти блоки могут быть объединены.

1. Пользователь инициирует операцию подписи в клиентском приложении на своем мобильном устройстве, связываясь с серверной частью интегрируемой с КриптоПро DSS системы.

1.1. Интегрируемая система передает необходимые сведения в КриптоПро DSS, где начинается операция подписи.

## Примечание

В зависимости от того, где хранится подписываемый документ, он будет передан либо по цепочке *Клиентское приложение* → *Интегрируемая ИС* → *КриптоПро DSS*, либо *Интегрируемая ИС* → *КриптоПро DSS*. Изображение документа на схеме не устанавливает требований к его местонахождению.

2. Одновременно с п. 1 клиентское приложение при помощи Deep Link вызывает приложение myDSS.

2.1. Открывшееся на мобильном устройстве пользователя приложение myDSS запрашивает у КриптоПро DSS информацию о подписываемом документе.

2.2. КриптоПро DSS отправляет в myDSS указанную в п. 2.1. информацию.

2.3. Получив информацию от КриптоПро DSS, myDSS отображает подписываемый документ пользователю для проверки.

2.4. Пользователь просматривает сообщение и/или документ, убеждается, что хочет выполнить данную операцию, и инициирует подтверждение операции в myDSS.

2.5. myDSS вычисляет необходимый для подтверждения операции код аутентификации.

2.6. myDSS отправляет код аутентификации в КриптоПро DSS.

2.6.1. КриптоПро DSS вычисляет код аутентификации и проверяет полученный код аутентификации.

2.6.2. В случае совпадения полученных в п. 2.6.1. кодов аутентификации КриптоПро DSS успешно подписывает документ.

2.7. КриптоПро DSS отправляет подписанный документ в интегрируемую систему.

## Примечание

Перед вызовом myDSS клиентскому приложению рекомендуется удостовериться (получить ответ от КриптоПро DSS), что операция подписи запущена.

3. После отправки кода аутентификации в КриптоПро DSS приложение myDSS при помощи Deep Link вызывает клиентское приложение.

4. (Опционально) Интегрируемая система отправляет подписанный документ в клиентское приложение.

## Примечание

Описанная в п. 2.7–3 цепочка *КриптоПро DSS* → *Интегрируемая ИС* → *Клиентское приложение* приводится для примера. Изображение подписанного документа на схеме не устанавливает требований к его местонахождению.

# Префикс Deep Link для myDSS

mydss://

## Формат URL для myDSS

Клиентское приложение вызывает myDSS следующим образом:

mydss://[operation]?[params]

где

- operation = start\_confirmation
- params:

- `callback` - (обязательный параметр) url для возврата в вызывающее приложение, закодированный в соответствии с [RFC 3986](#) (urlencode);
- `user_id` - (необязательный параметр) идентификатор пользователя для подтверждения;
- `transaction_id_list` - (необязательный параметр) операции для подтверждения, comma-separated, применяется только вместе с `user_id`.

#### Пример:

```
mydss://start_confirmation?callback=testapp%3A%2F%2Fmydss_callback&user_id=test-ebaad8fd-cafe-43e6-9fed-33ffb31c65a7&transaction_id_list=7b3bf903-55bf-4867-b224-294fd223afbf,858dfa54-3295-4b1a-9388-ff5534bc9590
```

В callback отправляется результат подтверждения следующим образом:

```
[callback_url]?mydss_result=[result]&error=[error]
```

где

- `result`
  - `success` - все операции обработаны;
  - `processed_partially` - обработано операций больше 0, но меньше общего количества;
  - `user_canceled` - отменено пользователем;
  - `error` - ошибка.
- `error`
  - 0, если result не равно `error`;
  - код ошибки с сервера;
  - 17, если указанный в `user_id` идентификатор пользователя не зарегистрирован.

#### Пример:

```
testapp://mydss_callback?mydss_result=error&error=278
```

# Аутентификация с помощью мобильного приложения на базе DSS SDK

DSS SDK для встраивания в мобильное приложение представляет собой набор программных компонентов для использования в мобильных приложениях, который позволяет производить удаленное выполнение операций подписи и управление сертификатами, а также подтверждать операции Пользователя в КриптоПро DSS, инициированные другими способами.

DSS SDK предоставляет **программный (API) и графический интерфейсы**, позволяющие выполнять следующие действия.

- Управление сертификатами пользователя:
  - Создание запроса на сертификат;
  - Установка сертификата;
  - Просмотр списка сертификатов и запросов;
  - Удаление сертификатов и/или запросов;
  - Отзыв сертификата;
  - Назначения сертификата по умолчанию;
  - Назначения дружественного имени сертификата.
- Отправка документов на подпись.
- Подтверждение или отклонение подписи документов:
  - В том числе подтверждение или отклонение действий пользователя в ИС.
- Управление учетной записью:
  - Регистрация новой учётной записи с привязкой мобильного устройства;
  - Привязка устройства к существующей учетной записи;
  - Просмотр списка устройств пользователя;
  - Удаление устройств пользователя;
  - Смена ПИН-кода для доступа к ключу аутентификации на устройстве пользователя;
  - Назначение дружественного имени ключа аутентификации.
- Просмотр истории операций пользователя.

DSS SDK предоставляет **графический интерфейс (окна)**, позволяющий выполнять следующие действия.

- Ввод нового пароля для защиты устанавливаемых векторов аутентификации.

Ввод пароля с повторным вводом для подтверждения. Окно отображается, если на стороне сервера разрешен только пароль для защиты векторов.

- Ввод пароля для доступа к векторам аутентификации.

Окно отображается, если на сервере разрешен только пароль для защиты векторов.

- Подпись: сопровождающий текст и список документов.
- Просмотр документа.
- Просмотр «сырого» представления документа.

**Разработчикам мобильного приложения**

[Подключение к сервису взаимодействия с DSS SDK](#) - раздел содержит данные для подключения к сервису взаимодействия с DSS SDK (mDAG) и инструкции по созданию и настройке аутентификации учетной записи, выпуску сертификата через Веб-интерфейс.

**Backend-разработчикам**

Раздел содержит руководство разработчика по интеграции с КриптоПро DSS.

Сценарии:

- [Назначение и управление аутентификацией через DSS SDK](#) — руководство разработчика по работе через Оператора DSS.
- [Подтверждение операций с помощью DSS SDK](#) — руководство разработчика по созданию операций для подтверждения через DSS SDK.
- [Выпуск сертификата пользователя](#) — руководство разработчика по выпуску сертификата пользователя.

API сервисов DSS:

- [Сервис Управления Пользователями](#) — API Оператора DSS для управления аутентификацией myDSS.
- [Сервис Подписи](#) - API Сервиса Подписи
- [Сервис Управления Пользователями](#) - API Сервиса Управления пользователями
- [Аутентификация](#) - сценарии аутентификации

### Примечание

Перед началом интеграции необходимо обратиться к Администратором сервиса для:

- регистрации OAuth-клиента для интегрируемой системы
- получения сертификата Оператора DSS.

Адреса сервисов:

АДРЕС СЕРВИСА	ОПИСАНИЕ
<a href="https://dss-sdk.cryptopro.ru/SignServer/rest/api/v2">https://dss-sdk.cryptopro.ru/SignServer/rest/api/v2</a>	Сервис Подписи
<a href="https://dss-sdk.cryptopro.ru/STS/ums">https://dss-sdk.cryptopro.ru/STS/ums</a>	Сервис Управления Пользователями
<a href="https://dss-sdk.cryptopro.ru/DocumentStore">https://dss-sdk.cryptopro.ru/DocumentStore</a>	Сервис Обработки Документов
<a href="https://dss-sdk.cryptopro.ru/STS">https://dss-sdk.cryptopro.ru/STS</a>	Сервис Маркеров Безопасности
<a href="https://dss-sdk.cryptopro.ru/STS/v2.0/confirmation">https://dss-sdk.cryptopro.ru/STS/v2.0/confirmation</a>	Сервис Подтверждения Операций

### Примечание

Сервисы доступны по портам:

- 443 – RSA TLS
- 4430 – GOST TLS

Эмулятор мобильного приложения

Эмулятор можно загрузить по [ссылке](#)

Для запуска требуется Windows + .NET FW v4.7.1 Справка по эмулятору:

```
MyDssSdkClient.exe help
```

# Подключение метода аутентификации через DSS SDK

Раздел содержит руководство разработчика по работе с DSS SDK на Сервисе Управления Пользователями. В разделе приведены основные сценарии использования, примеры HTTP-запросов и ответов REST Сервиса Управления Пользователями.

Так же в разделе приведены рекомендации Администраторам по настройке DSS для реализации различных сценариев работы с DSS SDK.

Перед началом интеграции с Сервисом Управления Пользователями Администратору DSS необходимо:

- Выпустить и зарегистрировать на DSS сертификат аутентификации Оператора DSS
- Настроить сервер myDSS API Gateway
- Включить метод аутентификации DSS SDK на Центре Идентификации
- Ввести лицензию на модуль аутентификации myDSS на Центре Идентификации

Сценарии должны выполняться учётной записью с ролью Оператора DSS.

Аутентификация Операторов DSS на Сервисе Управления Пользователями осуществляется по сертификату (двухстороннее TLS-соединение).

Методы и структуры используемые в данном разделе приведены в разделе [REST API Сервиса Управления Пользователями \(UMS\)](#) Методы для работы с DSS SDK приведены в разделе [Настройка аутентификации DSS SDK](#)

Последовательность шагов по регистрации пользователя:

1. [Регистрация логина пользователя.](#)
2. [Назначение метода первичной аутентификации.](#)

Дальнейшие действия зависят от сценария инициализации устройства пользователя.

- Регистрация анонимного устройства
  1. [Поиск устройства пользователя](#)
  2. [Присоединение устройства к УЗ пользователя](#)
  3. [Назначение метода аутентификации DSS SDK](#)
  4. [Получение QR-код Nonce \(опционально\)](#)
- Регистрация устройства по QR-коду Klnit
- Создание QR-кода Klnit
- [Назначение метода аутентификации DSS SDK](#)

Вспомогательные действия:

1. [Поиск пользователя](#)
2. [Получение сведений о DSS SDK](#)
3. [Отключение аутентификации через myDSS](#)
4. [Заполнение профиля пользователя](#)

## Регистрация логина пользователя

В качестве идентификатора (логина) пользователя могут выступать:

- логин
- адрес электронной почты
- номер телефона



## Внимание!

По умолчанию на DSS в качестве идентификатора разрешён только Логин.

Разрешить/запретить другие идентификаторы пользователя может Администратор DSS выполнив команду в консоли PowerShell:

```
Set-DssStsProperties -AvailableIdentifiers Login,Email,PhoneNumber
```

### Примеры запросов

- Регистрация пользователя по логину

```
POST https://host/STS/ums/user HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 28
Expect: 100-continue

{"Login":"DssTest-6f956360"}
```

- Регистрация пользователя по логину, email и номеру телефона

```
POST https://host/STS/ums/user HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 92
Expect: 100-continue
Connection: Keep-Alive

{"Login":"DssTest-05e789e7","PhoneNumber":"+70004064846","Email":"DssTest-0678acd4@dss.com"}
```

### Пример ответа

В ответ DSS вернёт идентификатор созданного пользователя (DssUserId). DssUserId используется при вызове любых методов Сервиса Управления Пользователями:

- возвращающих сведения об учётной записи пользователя
- изменяющих учётную запись пользователя.

Вызывающая система может сохранить DssUserId. Это позволит ускорить последующие обращения к Сервису Управления Пользователями, так как не потребуется получать DssUserId повторно.

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Date: Thu, 23 Aug 2018 14:59:09 GMT
Content-Length: 38

"264caee9-b77c-4b8c-b52a-ef9dd502f959"
```

### Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_identifiers	Переданный идентификатор запрещённый на DSS.

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_phone	Пользователь с указанным номер телефона уже зарегистрирован.
400	invalid_email	Пользователь с указанным email уже зарегистрирован.
400	invalid_login	Пользователь с указанным логином уже зарегистрирован.
500	An error has occurred	1. В поле Login указан номер телефона или email. 2. Неверно сформирован email. 3. Неверно сформирован номер телефона.

## Назначение метода первичной аутентификации

После регистрации логина пользователя необходимо назначить метод первичной аутентификации. Пользователю может быть назначен один или несколько методов первичной аутентификации:

МЕТОД	ОПИСАНИЕ
/user/{DssUserId}/authmethod/identity	Только идентификация
/user/{DssUserId}/authmethod/password	Аутентификация по паролю
/user/{DssUserId}/authmethod/cert	Аутентификация по сертификату
/user/{DssUserId}/authmethod/external	Аутентификация через сторонний Центр Идентификации

Чаще всего при использовании DSS SDK в качестве метода первичной аутентификации назначают "Только идентификация".

### Внимание!

Назначаемый метод аутентификации должен быть разрешён на DSS. Включить или отключить метод аутентификации должен Администратор на сервере DSS.

Разрешить/запретить метод аутентификации можно на Сервере DSS командами:

```
Enable-DssAuthenticationMethod -Id <method_ID>
```

```
Disable-DssAuthenticationMethod -Id <method_ID>
```

### Внимание!

Совместное включение методов identity и password допустимо, но использоваться будет метод "Только идентификация".

## Примеры запросов

Назначение метода первичной аутентификации "Только идентификация"

```
POST https://host/STS/ums/user/d0bfa4e1-808d-4c28-b3cb-0dc5a591d300/authmethod/idonly HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 2
Expect: 100-continue

{}
```

## Пример ответа

Назначение метода аутентификации не имеет возвращаемого значения.

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Date: Thu, 23 Aug 2018 16:35:38 GMT
Content-Length: 0
```

## Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	wrong_operation	Метод аутентификации уже назначен.
400	invalid_authn_method	Метод аутентификации запрещён на сервере DSS.
404	user_not_found	Пользователь не найден.

## Поиск устройства пользователя

Поиск анонимного устройства пользователя может быть выполнен по следующим ключам поиска:

- kid - идентификатор устройства пользователя
- Alias - псевдоним устройства пользователя

Поиск осуществляется с помощью метода [authntokens](#)

## Пример запроса

```
GET https://host/STS/ums/authntokens HTTP/1.1

Content-Length: 94
Content-Type: application/json; charset=utf-8
Accept: application/json
Expect: 100-continue
Host: host

{"StartPosition":0,"EndPosition":10,"Filters":[{"Column":1,"Operation":0,"Value":"73128110"}]}
```

## Примеры ответов

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
 "TokenInfos":
 [
 {
 "Id":35742,
 "Serial":"73128110",
 "UserName":null,
 "TokenType":"MyDss",
 "Parameters":
 {
 "DeviceFingerprintRequired":"True",
 "ProfileId":"84873bac-3205-4030-944f-9eb571627ccc",
 "CreationType":"Anonymous",
 "DeviceFingerprint":"9af7e394-76d4-47f5-b0d2-d6217572c5c6",
 "DeviceName":"MyApple",
 "PushAddress":"push-address",
 "OsType":"2",
 "OsVersion":"4.0.0",
 "DeviceModel":"AppleiPhoneA",
 "Locale":"ru-RU",
 "TimeZoneUTCOffset":"3",
 "AppVersion":"1.0",
 "IMEI":null,
 "TempUserName":"MdagTestUser-D547B268",
 "NotBefore":"11/27/2019 18:37:45",
 "NotAfter":"02/27/2021 18:37:45",
 "VerificationNonce":"cKx+pdmejI/SRz30Qex32wd6vNF7oVni6LsAf7Sf0Zw="
 }
 }
],
 "TotalCount":4876,
 "AffectedCount":1
}
```

Присоединение устройства к УЗ пользователя

#### Пример запроса

```
POST https://host/STS/ums/user/8d16086d-676c-49d6-bf49-09e6f936a596/mydss/assign HTTP/1.1
Accept: application/json; charset=utf-8
Host:host

{"Kid":"73128110"}
```

#### Примеры ответов

```
HTTP/1.1 200 OK
Pragma: no-cache
Content-Type: application/json; charset=utf-8
```

```
{
 "Uid": "8d16086d-676c-49d6-bf49-09e6f936a596",
 "Kid": "73128110",
 "DeviceName": "MyApple",
 "NotBefore": 1574869065,
 "NotAfter": 1614440265,
 "State": "NotVerified",
 "UserName": "MdagTestUser-D547B268",
 "Profile": "{ \"Version\": 1, \"Keys\": {
 \"1\": \"%D0%9E%D0%93%D0%A0%D0%9D\\\", \"2\": \"%D0%9E%D0%93%D0%A0%D0%9D%D0%98%D0%9F\\\", \"3\": \"%D0%A1%D0%9D%D0%98%D0%9B%D0%A1\\\", \"4\": \"%D0%98%D0%9D%D0%9D\\\", \"5\": \"%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F%D0%BF%D0%BE%D1%87%D1%82%D0%B0\\\", \"6\": \"%D0%A1%D1%82%D1%80%D0%B0%D0%BD%D0%B0\\\", \"7\": \"%D0%9E%D0%B1%D0%BB%D0%B0%D1%81%D1%82%D1%8C\\\", \"8\": \"%D0%93%D0%BE%D1%80%D0%BE%D0%B4\\\", \"9\": \"%D0%9E%D1%80%D0%B3%D0%B0%D0%BD%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F\\\", \"10\": \"%D0%9F%D0%BE%D0%B4%D1%80%D0%B0%D0%B7%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5\\\", \"11\": \"%D0%9E%D0%B1%D1%89%D0%B5%D0%B5%D0%B8%D0%BC%D1%8F\\\", \"12\": \"%D0%90%D0%B4%D1%80%D0%B5%D1%81\\\", \"13\": \"%D0%94%D0%BE%D0%BB%D0%B6%D0%BD%D0%BE%D1%81%D1%82%D1%8C\\\", \"14\": \"%D0%98%D0%BD%D0%B8%D1%86%D0%B8%D0%B0%D0%BB%D1%8B\\\", \"15\": \"%D0%98%D0%BC%D1%8F\\\", \"16\": \"%D0%A4%D0%B0%D0%BC%D0%B8%D0%BB%D0%B8%D1%8F\\\", \"Values\": {}}\",
 \"NonceRequired\": true
 }
}
```

## Назначение метода аутентификации DSS SDK

Далее необходимо назначить DSS SDK как метод аутентификации.

```
POST https://host/STS/ums/user/8d16086d-676c-49d6-bf49-09e6f936a596/authmethod/mydss?level=0 HTTP/1.1
Accept: application/json; charset=utf-8
Host: host

{"Kid": "73128110"}
```

## Примеры ответов

```
HTTP/1.1 200 OK
```

## Получение QR-код Nonce

Необходимость выполнения данного шага зависит от флага `NonceRequired` в [сведениях об устройстве](#) пользователя. Если для завершения активации устройства требуется подтверждение через QR-код Nonce, то полученный QR-код необходимо передать пользователю. После того как пользователь отсканирует QR-код устройство пользователя перейдет в состояние Active.

`QrCode` - QR-код в кодировке base64

## Пример запрос

```
POST https://host/STS/ums/user/8d16086d-676c-49d6-bf49-09e6f936a596/mydss/verify/get HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 2
Expect: 100-continue

{"Kid": "73128110"}
```

## Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"QrCode": "R0lGODlh...zuI8zuRczuZ8zuisEgEBADs=",
"QrCodeData": {"type": "Verification", "version": 1, "data": {"kid": "73128110", "uid": "8d16086d-676c-49d6-bf49-09e6f936a596", "service_url": "https://simdss.cryptopro.ru:4430/mydss", "seed": "DQiMn1xAvdQFQlAR4ceB3NkWaeVdppM7klIV0ZssTc=", "nonce": "cKx+pdmejI/SRz30Qex32wd6vNF7oVni6LsAf7Sf0Zw="}}},
"Data": {"type": "Verification", "version": 1, "data": {"kid": "73128110", "uid": "8d16086d-676c-49d6-bf49-09e6f936a596", "service_url": "https://simdss.cryptopro.ru:4430/mydss", "seed": "DQiMn1xAvdQFQlAR4ceB3NkWaevDppM7klIV0ZssTc=", "nonce": "cKx+pdmejI/SRz30Qex32wd6vNF7oVni6LsAf7Sf0Zw="}}}
```

## Поиск пользователя

Сервис Управления пользователями предоставляет несколько возможностей поиска пользователя:

- По логину, номеру телефона или email
- По идентификатору DssUserId
- Расширенный поиск

По логину, номеру телефона или email

### Пример запроса

Тип ключа поиска может принимать значения (значение параметра `type`):

- Login
- PhoneNumber
- Email

```
GET https://host/STS/ums/user?type=Login&value=DssTest-dc3bf3f5 HTTP/1.1
Accept: application/json
Host: host
```

### Пример ответа

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Content-Length: 334

{"UserId": "d1831dea-985f-4df1-a54b-2497eeace2f2", "Login": "DssTest-dc3bf3f5", "PhoneNumber": null, "Email": null, "PhoneConfirmed": false, "EmailConfirmed": false, "DisplayName": null, "DistinguishName": "", "AccountLocked": false, "Group": "Default", "CreationDate": "2018-08-24T14:36:33.02", "LockoutDate": null, "LastLoginDate": "2018-08-24T14:36:33.02"}
```

По идентификатору DssUserId

### Пример запроса

```
GET https://host/STS/ums/user/d1831dea-985f-4df1-a54b-2497eeace2f2 HTTP/1.1
Accept: application/json
Host: host
```

### Пример ответа

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Content-Length: 334

{"UserId": "d1831dea-985f-4df1-a54b-2497eeace2f2", "Login": "DssTest-
dc3bf3f5", "PhoneNumber": null, "Email": null, "PhoneConfirmed": false, "EmailConfirmed": false, "DisplayName": null, "D
istinguishName": "", "AccountLocked": false, "Group": "Default", "CreationDate": "2018-08-
24T14:36:33.02", "LockoutDate": null, "LastLoginDate": "2018-08-24T14:36:33.02"}
```

Расширенный поиск

Расширенный поиск позволяет применять различные фильтры для поиска пользователей. Результатом выполнения метода может быть группа пользователей, отвечающая параметрам фильтра.

Поиск пользователей можно выполнить по одному или нескольким параметрам:

ПАРАМЕТР	КОД	ОПИСАНИЕ
Login	0	Логин пользователя
PhoneNumber	1	Номер телефона
Email	2	Адрес электронной почты
CreateDate	3	Дата создания учётной записи
GroupId	4	Идентификаторы группы пользователя

Код параметра указывается в поле

Операции сравнения могут быть следующих типов:

ТИП	КОД	ОПИСАНИЕ
Equal	0	Строгое равенство
NotEqual	1	Не равно
Like	2	Содержит
Greater	3	Больше
Less	4	Меньше

Код операции указывается в поле

Тип сравнения Like определяет, совпадает ли указанная символьная строка с заданным шаблоном. Шаблон может включать обычные символы и символы-шаблоны. Во время сравнения с шаблоном необходимо, чтобы его обычные символы в точности совпадали с символами, указанными в строке. Символы-шаблоны могут совпадать с произвольными элементами символьной строки.

Поддерживаются следующие символы шаблоны:

СИМВОЛ-ШАБЛОН	ОПИСАНИЕ	ПРИМЕР
%	Любая строка, содержащая ноль или более символов.	%вано%
(подчеркивание)	Любой одиночный символ.	_етров
[ ]	Любой одиночный символ, содержащийся в диапазоне ([a-f]) или наборе ([abcdef]).	[Л-С]омов
[^]	Любой одиночный символ, не содержащийся в диапазоне ([^a-f]) или наборе ([^abcdef]).	'ив[^a]%

Параметры `StartPosition` и `EndPosition` определяют начальную и конечную позицию из итоговой выборки. Данные параметры могут быть использованы для страничной выборки пользователей

При поиске пользователей по времени создания значение фильтра должно иметь следующий формат: **yyyy-MM-ddThh:mm:ss**

Общее количество элементов подпадающих под критерии фильтра возвращается в параметре `TotalCount`. Количество элементов отданных методом возвращается в параметре `AffectedCount`: `AffectedCount <= EndPosition - StartPosition`

Примеры запросов

Получить пользователя с заданным логином:

```
POST https://host/STS/ums/users HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 101
Expect: 100-continue

{"StartPosition":1,"EndPosition":1,"Filters":[{"Column":0,"Operation":0,"Value":"DssTest-dc3bf3f5"}]}
```

Проверка были ли создан пользователь с заданным логином в указанном промежуток времени:

```
POST https://host/STS/ums/users HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 172
Expect: 100-continue

{"StartPosition":1,"EndPosition":1,"Filters":[{"Column":0,"Operation":0,"Value":"DssTest-2fa204c5"}, {"Column":3,"Operation":3,"Value":"2018-08-24T15:12:12.4683672+03:00"}]}
```

Получить пользователей созданных в указанный промежуток времени:

```
POST https://host/STS/ums/users HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 161
Expect: 100-continue

{"StartPosition":1,"EndPosition":10,"Filters":[{"Column":3,"Operation":4,"Value":"2018-08-25T04:24:50"}, {"Column":3,"Operation":3,"Value":"2018-08-23T04:24:50"}]}
```

Пример ответа



```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Content-Length: 386

{"UserInfos":[{"UserId":"d1831dea-985f-4df1-a54b-2497eeace2f2","Login":"DssTest-dc3bf3f5","PhoneNumber":null,"Email":null,"PhoneConfirmed":false,"EmailConfirmed":false,"DisplayName":null,"DistinguishName":"","AccountLocked":false,"Group":"Default","CreationDate":"2018-08-24T14:36:33.02","LockoutDate":null,"LastLoginDate":"2018-08-24T14:36:33.02"}],"TotalCount":2047,"AffectedCount":1}
```

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404	user_not_found	Пользователь не найден.
500	An error has occurred	Неверно указано значение или тип фильтра.

Получение сведений о myDSS

Ниже приведён список методов, которые позволяют проверить:

- назначен ли пользователю ключ аутентификации myDSS
- срок действия ключа аутентификации myDSS
- назначен ли пользователю метод аутентификации myDSS
- список действий требующих подтверждения

Проверка назначен ли ключ аутентификации DSS SDK пользователю

Пример запроса

```
GET https://host/STS/ums/user/c9b4b217-edc1-4329-be4d-1cddaecdea4d/mydss HTTP/1.1
Accept: application/json
Host:host
```

Примеры ответов

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 91

{"UserId":"9912c464-8d2e-4145-bcee-587f10dad61b","KeyExpirationTime":"2018-09-23T00:00:00"}
```

Если ключ аутентификации не назначен пользователю ответ сервиса будет содержать пустой список Keys:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Content-Length: 4

{
 "UserId": "3b104793-1796-4c90-b3ec-09bbbfca6ced",
 "Keys": [],
 "InitializationToken": null,
 "Blocked": false
}
```

Получение схемы аутентификации пользователя

Пример запроса

```
GET https://host/STS/ums/user/01df2040-3e16-4405-9947-fc4152448c13/authmethod HTTP/1.1
Accept: application/json
Host: host
```

Пример ответа

Сервис возвращает список элементов содержащих:

- Идентификатор метода аутентификации
- Уровень метода аутентификации

Методы первичной аутентификации имеют уровень 0.

Методы вторичной аутентификации имеют уровень 1.

Список идентификаторов методов первичной аутентификации:

ИДЕНТИФИКАТОР	ОПИСАНИЕ
<a href="http://schemas.microsoft.com/ws/2012/09/identity/authenticationmethod/none">http://schemas.microsoft.com/ws/2012/09/identity/authenticationmethod/none</a>	Только идентификация
<a href="http://dss.cryptopro.ru/identity/authenticationmethod/password">http://dss.cryptopro.ru/identity/authenticationmethod/password</a>	По паролю
<a href="http://dss.cryptopro.ru/identity/authenticationmethod/certificate">http://dss.cryptopro.ru/identity/authenticationmethod/certificate</a>	По сертификату
<a href="http://dss.cryptopro.ru/identity/authenticationmethod/saml">http://dss.cryptopro.ru/identity/authenticationmethod/saml</a>	Через сторонний Центр Идентификации

Список идентификаторов методов вторичной аутентификации:

ИДЕНТИФИКАТОР	ОПИСАНИЕ
<a href="http://dss.cryptopro.ru/identity/authenticationmethod/mobile">http://dss.cryptopro.ru/identity/authenticationmethod/mobile</a>	Аутентификация через мобильное приложение DSS
<a href="http://dss.cryptopro.ru/identity/authenticationmethod/otpviasms">http://dss.cryptopro.ru/identity/authenticationmethod/otpviasms</a>	Одноразовые пароли по SMS
<a href="http://dss.cryptopro.ru/identity/authenticationmethod/otpviaemail">http://dss.cryptopro.ru/identity/authenticationmethod/otpviaemail</a>	Одноразовые пароли по Email
<a href="http://dss.cryptopro.ru/identity/authenticationmethod/oath">http://dss.cryptopro.ru/identity/authenticationmethod/oath</a>	Аутентификация по протоколу Oath

ИДЕНТИФИКАТОР	ОПИСАНИЕ
<a href="http://dss.cryptopro.ru/identity/authenticationmethod/simauth">http://dss.cryptopro.ru/identity/authenticationmethod/simauth</a>	Аутентификация на SIM-карте
<a href="http://dss.cryptopro.ru/identity/authenticationmethod/airkey">http://dss.cryptopro.ru/identity/authenticationmethod/airkey</a>	Аутентификация через мобильное приложение Indeed AirKey

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Content-Length: 189

[{"MethodUri":"http://schemas.microsoft.com/ws/2012/09/identity/authenticationmethod/none","Level":0},
{"MethodUri":"http://dss.cryptopro.ru/identity/authenticationmethod/mobile","Level":1}]
```

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404	user_not_found	Пользователь не найден.

Получение списка операций, требующих подтверждения

Пример запроса

```
GET https://host/STS/ums/user/8386abb0-b3d0-44c0-96a7-90d635e45d21/operationpolicy HTTP/1.1
Accept: application/json
Host: 192.168.109.149
```

Пример ответа

Описание операций приведено в разделе [выше](#)

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Content-Length: 679

[{"Action":"Issue","ConfirmationRequired":true}, {"Action":"SignDocument","ConfirmationRequired":true},
{"Action":"SignDocuments","ConfirmationRequired":false},
{"Action":"DecryptDocument","ConfirmationRequired":false},
{"Action":"CreateRequest","ConfirmationRequired":false}, {"Action":"ChangePin","ConfirmationRequired":false},
{"Action":"RenewCertificate","ConfirmationRequired":true},
{"Action":"RevokeCertificate","ConfirmationRequired":false},
{"Action":"HoldCertificate","ConfirmationRequired":false},
{"Action":"UnholdCertificate","ConfirmationRequired":false},
{"Action":"DeleteCertificate","ConfirmationRequired":false},
{"Action":"PrivateKeyAccess","ConfirmationRequired":false}]
```

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404	user_not_found	Пользователь не найден.

## Отключение аутентификации через DSS SDK

Отключение аутентификации через myDSS состоит из последовательности шагов:

1. Отключить требования подтверждения операций
2. Отключить метода аутентификации DSS SDK
3. Удалить ключи аутентификации пользователя

## Отключение требований подтверждения операций

### Примечание

Отключение методов аутентификации требуется, если myDSS является единственным способом вторичной аутентификации. Если пользователю назначены другие способы аутентификации (например, одноразовые пароли по SMS), то отключение методов не требуется.

### Примечание

Если отключить метод аутентификации myDSS, не отключив требование подтверждения операций операций, то пользователь не сможет выполнить данные операции.

## Пример запроса

```
POST https://hostname/STS/ums/user/206adc4f-3262-469c-9871-b7a7cabaa979/operationpolicy HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: 192.168.109.149
Content-Length: 2
Expect: 100-continue

[]
```

## Пример ответа

Метод не имеет возвращаемого значения

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Content-Length: 0
```

## Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404	user_not_found	Пользователь не найден.

## Отключение метода аутентификации

## Пример запроса

```
DELETE https://hostname/STS/ums/user/e06a4bc1-4c31-4f8d-a7a7-920b66ca4ad6/authmethod/mydss HTTP/1.1
Accept: application/json
Host: host
Content-Length: 0
```

## Пример ответа

Метод не имеет возвращаемого значения

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Content-Length: 0
```

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404	user_not_found	Пользователь не найден.

Удаление ключа аутентификации

Пример запроса

```
DELETE https://hostname/STS/ums/user/a17efd43-181a-45b7-8d60-088b6889480c/mydss HTTP/1.1
Accept: application/json
Host: host
Content-Length: 0
```

Пример ответа

Метод не имеет возвращаемого значения.

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Content-Length: 0
```

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404	user_not_found	Пользователь не найден.
400	wrong_operation	Нельзя удалить ключ аутентификации пользователя не отключив метод аутентификации myDSS

Заполнение профиля пользователя

# Подтверждение операций с помощью DSS SDK

Раздел содержит руководство разработчика по подтверждению (отклонению) операций с помощью DSS SDK на примере подтверждения операции подписи. В разделе приведены основные сценарии использования, примеры HTTP-запросов и ответов REST-сервисов DSS.

Сценарии должны выполняться от имени Пользователя DSS.

## Прerequisites

- Зарегистрированный OAuth-клиент для интегрируемой системы

### Примечание

Рекомендуем использовать [сервисный OAuth-клиент](#)

- Пользователю создана учетная запись в Центре Идентификации
- Пользователю выпущен сертификат подписи

### Примечание

Последовательность действий по созданию УЗ пользователя приведена в разделе [Подключение метода аутентификации через DSS SDK](#)

- На Сервисе Обработки Документов (СОД) зарегистрированы плагины для отображения документов.

## Общие сведения

Последовательность шагов при подтверждении операции подписи на Сервисе Подписи:

1. [Аутентификация пользователя на Центре Идентификации](#)
2. [Загрузка документов для подписи](#)
3. [Создание операции подписи на Сервисе Подписи](#)
4. [Отправка запроса на подтверждение операции подписи](#)
5. Ожидание подтверждения операции в DSS SDK
6. [Получение подписанного документа на Сервисе Подписи](#)

### Примечание

Подтверждение других операций на Сервисе Подписи (создание запроса на сертификат, отзыв сертификата, подпись пакета документов и т.п.) состоит из аналогичной последовательности шагов - отличие в типе и параметрах операции, создаваемой на Сервисе Подписи.

Все запросы к Сервису Подписи и Сервису Обработки Документов должны быть аутентифицированы с помощью 'access\_token'. 'access\_token' передаётся в заголовке `Authorization` с типом Bearer. 'access\_token' выдаёт Центр Идентификации после успешной аутентификации пользователя. Сценарии аутентификации пользователя приведены в разделе [Аутентификация](#)

Подпись может быть выполнена в

- синхронном режиме
- [асинхронном режиме](#).

Режимы отличаются этапом, на котором интегрируемая система получит callback о подтверждении операции.

### Примечание

**Синхронный режим** сохранён для обратной совместимости с предыдущими версиями КриптоПро DSS (2.0.2882 и ранее).

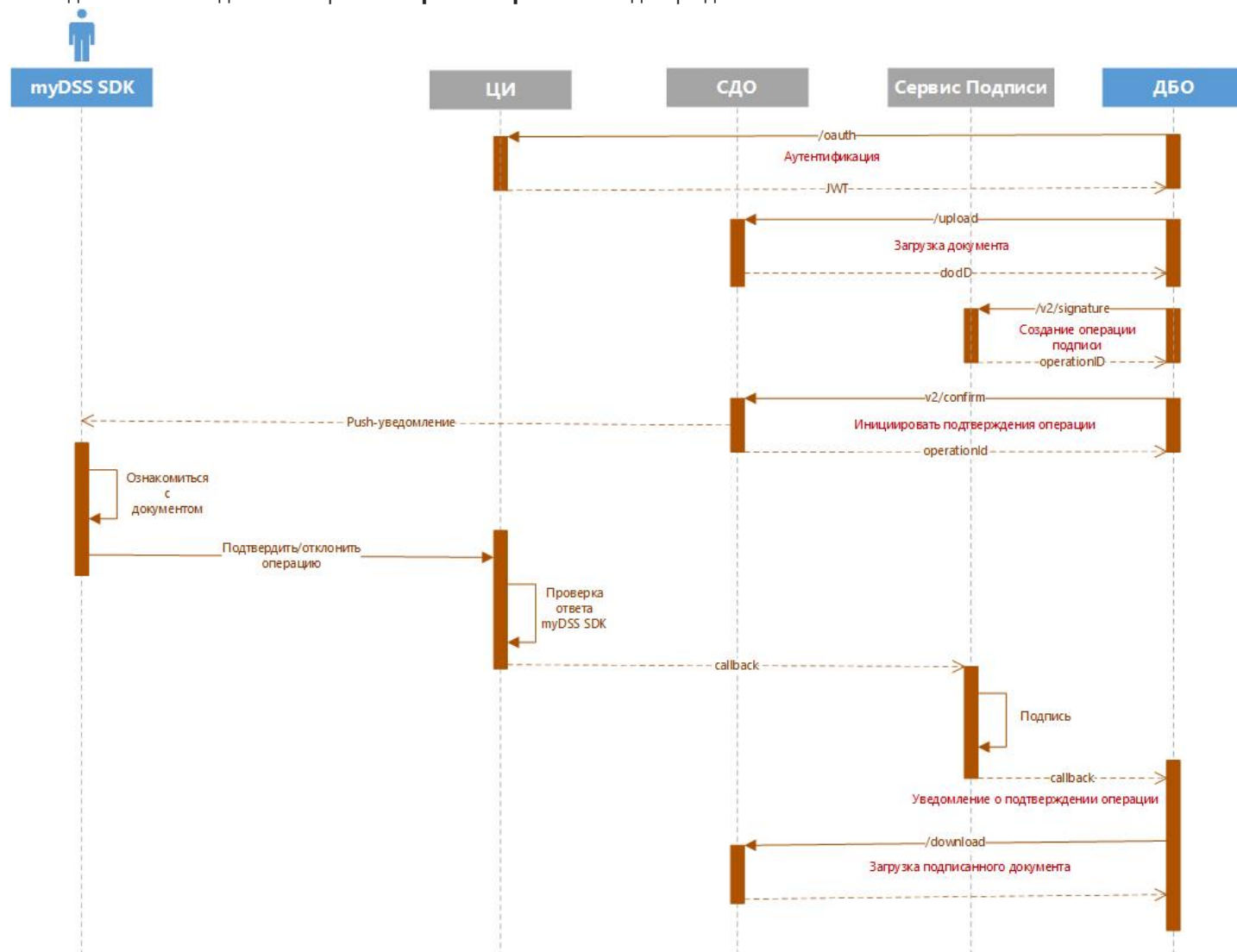
В **асинхронном режиме** callback будет отправлен в интегрируемую систему после того как пользователь подтвердит операцию в мобильном приложении и документ будет подписан. Получив callback интегрируемая система должна:

- выгрузить из Сервиса Обработки Документов подписанный документ

## Примечание

В случае ошибок подтверждения операции, отклонения операции вызывающая система получит callback от Центра Идентификации.

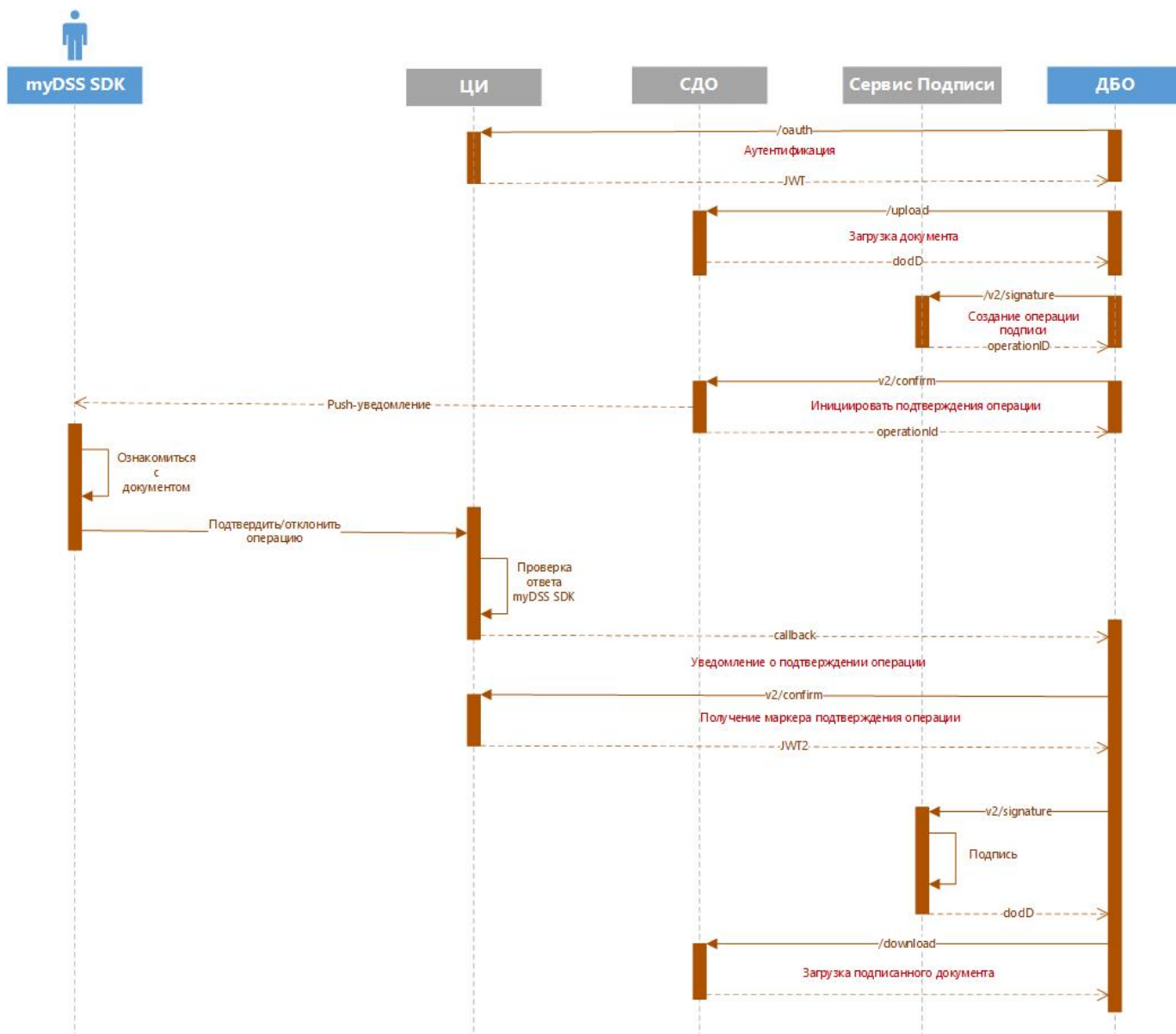
Последовательность действий при **асинхронном режиме** подтверждении



В **синхронном режиме** callback будет отправлен в интегрируемую систему после того как пользователь подтвердит операцию в мобильном приложении. Получив callback интегрируемая система должна:

- получить 'access\_token' в Центре Идентификации, содержащий сведения о подтверждённой операции
- с полученным 'access\_token' запросить подпись документа на Сервисе Подписи
- выгрузить из Сервиса Обработки Документов подписанный документ

Последовательность действий при **синхронном режиме** подтверждении



## Подтверждение операции на Сервисе Подписи

### Предварительные условия

- Пользователю создана учётная запись в DSS;
- Пользователю назначена аутентификация через DSS SDK;
- Пользователю выпущен сертификат электронной подписи;
- На сервере DSS зарегистрирован OAuth20 клиент.

В подтверждении операции задействованы следующие сервисы DSS:

КОНЕЧНАЯ ТОЧКА	СЕРВИС	ОПИСАНИЕ
https://<host>/<StsAppName>/oauth	Сервис Аутентификации	Аутентификация пользователей для возможности обращений к сервисам: * Сервису Подписи * Сервису Обработки Документов * Сервису Подтверждения Операций
https://<host>/<DocStoreAppName>/	Сервис Преобразования документов	Загрузка документов для подписания и получение подписанных документов



КОНЕЧНАЯ ТОЧКА	СЕРВИС	ОПИСАНИЕ
https://<host>/<SignServerAppName>/rest/api/v2/signature/	Сервис Подписи	Создание операции и получение результатов, подтвержденной операции
https://<host>/<StsAppName>/v2.0/confirmation	Сервис Подтверждения Операций	Подтверждение транзакций

## Примечание

У Администратора DSS необходимо получить значение параметров client\_id и resource. resource - идентификатор Сервиса Подписи, имеет вид: urn:cryptopro:dss:signserver:<SignServerAppName>

## Аутентификация пользователя на Центре Идентификации

В примере рассматривается авторизация с использованием учётных данных пользователя (логин/пароль). Подробная информация по протоколу аутентификации: [The OAuth 2.0 Authorization Framework](#)

Параметры запроса:

- grant\_type - тип разрешения, в данном сценарии равен **password**.
- password – пароль пользователя.
- resource – идентификатор Сервиса Подписи.

В заголовке Authorization HTTP-запроса клиент должен передать идентификатор OAuth-клиента и секрет (если используется): Authorization: Basic Base64(<client\_id>:<secret>)

## Примечание

В примере значение параметр **password** оставлено пустым, так как пользователю в качестве первичной аутентификации назначен метод "Только Идентификация"

## Пример запроса

```
POST https://host/STS/oauth/token HTTP/1.1
Authorization: Basic dGVzdENsawVudDo=
Content-Type: application/x-www-form-urlencoded
Host: host
Content-Length: 101
Expect: 100-continue
Connection: Keep-Alive

grant_type=password&username=mydss&password=&resource=urn%3Acryptopro%3Adss%3Asignserver%3Asignserver
```

В случае успешной аутентификации ответ будет содержать:

- access\_token - AccessToken, выпущенный Центром Идентификации DSS
- token\_type - Тип токена
- expires\_in - Время жизни токена в секундах

Значение параметра access\_token необходимо будет использовать при обращениях к Сервису Подписи и Сервису Подтверждения Операций.

## Пример ответа

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 2017
Content-Type: application/json; charset=utf-8
Expires: -1

{
 "access_token": "eyJ0eXAiOiJKV... 5Wti-H8CeXycwB6A",
 "expires_in": 300,
 "token_type": "Bearer"
}
```

## Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_client	OAuth-клиент не зарегистрирован или неверно указан clientID
400	unauthorized_client	OAuth-клиент использует незарегистрированный сценарий аутентификации (Flow)
400	invalid_request	Неверно сформирован параметр resource
500	An error has occurred	1. Проверяющая сторона с идентификатором resource не зарегистрирована.

## Загрузка документа

Описание API Сервиса Преобразования документов - [Сервис Преобразования документов](#)

Подписываемый документ необходимо загрузить в Сервис Обработки Документов. Сервис подготовит необходимые данные для подтверждения операции подписи:

- хэш от документа (ГОСТ Р 34.11-2012)
- Выжимка из документа
- Печатное представление документа
- PDF-представление документа

Документ загружается на сервер в поточном режиме. После загрузки сервис вернёт ID загруженного документа. Идентификатор загруженного документа необходимо будет передать в Сервис Подписи при [создании операции подписи](#).

Выжимка из документа и PDF-представление документа являются обязательными атрибутами документа при подтверждении операции подписи через DSS SDK. PDF-представление документа всегда формируется сервисом. Выжимка из документа может быть сформирована как сервисом так и передана из вызывающей системы [PostDocumentInput](#) в параметре `AdditionalInfo` в ключе `SnippetTemplate`.

Печатное представление документа может быть сформировано как сервисом так и передано из вызывающей системы [PostDocumentInput](#) в параметре `AdditionalInfo` в ключе `DocumentTemplate`.

Содержимое документа передается в теле запроса с `Content-Type: application/octet-stream`, а дополнительные параметры [PostDocumentInput](#) – в отдельном HTTP заголовке `CPDSS-POSTDOC`, закодированном в формат Base64.

## Пример запроса

Загрузка документа на сервер

```
POST /documentstore/api/documents HTTP/1.1
Host: dss-sdk.cryptopro.ru
Content-Type: application/octet-stream
CPDSS-POSTDOC: eyJGawxlbmFtZSI6InRlc3QudHh0In0=
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cGE6YWV0IjoiZm9udC5kaW8iLCJpc29udHh0In0=
Content-Length: 1556
```

### Пример ответа

Сервис возвращает ID загруженного документа.

```
{
 "DocumentId": "8336ca2d-10b9-499a-8f85-6292003ffdf"
}
```

## Создание операции подписи на Сервисе Подписи

Описание API Сервиса Подписи - [Сервис Подписи](#)

На данном шаге создаётся операция подписи, которая будет подтверждена с помощью DSS SDK.

При создании операции подписи необходимо указать:

- список подписываемых документов
- идентификатор сертификата подписи
- параметры подписи

Результатом вызова является ID операции подписи. `Operation -> Id` будет использован на следующем шаге при запросе подтверждения операции.

### Примечание

Вызывающая система должна сохранить идентификатор созданной операции. Идентификатор может быть использован для проверки статуса операции, например, если вызывающая система не получила callback о завершении подтверждения операции в DSS SDK.

Если сертификат пользователя назначен 'Сертификатом по умолчанию', то в параметре `CertificateId` может быть передано значение `0`.

Параметры подписи могут быть переданы в двух вариантах:

- шаблон подписи (`ProcessingTemplateId`)
- явно задание параметров подписи (`Parameters`)

Шаблон подписи должен быть предварительно создан Администратором DSS. Идентификатор шаблона может быть либо явно задан в настройках вызывающей системы, либо может быть получен из [политики Сервиса Подписи](#)

### Примечание

На Сервисе Подписи может быть задано несколько шаблонов подписи. Выбор шаблона из политики сервиса подписи может быть использован если вызывающая система поддерживает интерактивное взаимодействие с пользователем.

Описание параметров подписи и примеры задания параметров подписи приведены - [Параметры подписи](#) и [Примеры запросов на подпись](#)

Если используется режим **асинхронной подписи**, то в запросе необходимо указать параметры `IsAsync`, `Callback`.

### Пример запроса

```
POST /SignServer/rest/api/v2/signature/ HTTP/1.1
Host: dss-sdk.cryptopro.ru
Content-Type: application/json
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbG... PsIMFZormrKd35nd1xRw8foIpww
Accept: */*
Content-Length: 154
Connection: keep-alive

{
 "BinaryData": [{
 "RefId" : "8336ca2d-10b9-499a-8f85-6292003ffddf"
 }],
 "Signature" :
 {
 "CertificateId": "0",
 "ProcessingTemplateId": "1"
 },
 "IsAsync": "true",
 "Callback": "https://hostname/callback"
}
```

### Пример ответа

```
{
 "Operation": {
 "Id": "5070c752-ae8a-46f4-aea3-c4d38930b915",
 "Result": null,
 "Status": "Created",
 "Error": null,
 "ErrorDescription": null,
 "ExpirationDate": "1588636611"
 }
}
```

### Отправка запроса на подтверждение операции подписи

После создания операции подписи на Сервисе Подписи необходимо отправить запрос на её подтверждение в DSS SDK. В запросе вызывающая система передаёт:

- Идентификатор операции сервиса подписи
- `callback` Адрес для оповещения о завершении операции

### Примечание

Если интегрируемая система не передала адрес `callback` при вызове `/v2/confirmation`, то статус подтверждения операции она сможет узнать периодически [опрашивая](#) `/v2.0/confirmation`.

### Пример запроса

```
POST /STS/v2.0/confirmation HTTP/1.1
Host: dss-sdk.cryptopro.ru
Content-Type: application/json
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cGE6YW90cm8iOnsiaWF0dXkiOiJkaWY6ZGVpdCI9fQ.
Content-Length: 118
```

```
{
 "Resource": "urn:cryptopro:dss:signserver:SignServer",
 "ClientId": "oauth-client-id",
 "ClientSecret": "oauth-client-secret",
 "OperationId": "5070c752-ae8a-46f4-aee3-c4d38930b915",
 "CallbackUri": "https://externalhost/system/callback"
}
```

### Пример ответа

```
{
 "Challenge": {
 "Title": {
 "Value": "Подтвердите операцию на устройстве с помощью приложения."
 },
 "TextChallenge": [
 {
 "Label": "Подпись документа. не задано. Тип подписи: CAdES. Сертификат: txusr300.",
 "ExpiresIn": 300,
 "CreatedAt": 1575834219,
 "ExpiresInSpecified": true,
 "IsHidden": false,
 "AuthnMethod": "http://dss.cryptopro.ru/identity/authenticationmethod/mydss",
 "RefID": "ac93dc5f-f374-4c0f-a37d-cb5a90e1af97",
 "Title": "Подтвердите операцию на устройстве с помощью приложения."
 }
]
 },
 "IsFinal": false,
 "IsError": false
}
```

Для подтверждения транзакции, созданной на Сервисе Подписи, пользователь отправляет запрос содержащий:

- `CallbackUri` - адрес для оповещения о завершении транзакции (опционально).
- `OperationId` – идентификатор транзакции, созданной на сервисе подписи.
- `Resource` – идентификатор Сервиса Подписи.
- `ClientId` - идентификатор OAuth клиента.
- `ClientSecret` - пароль OAuth клиента (для неконфиденциальных клиентов данный параметр не указывается).

В заголовке Authorization HTTP-запроса клиент должен передать токен, полученный на первом шаге: Authorization: Bearer <access\_token>.

При получении запроса Сервис Подтверждения Операций и сервис myDSS начнут процедуру подтверждения операции в мобильном приложении. В частности отправят Push-уведомление пользователю.

### Пример ответа

Ответ Сервиса Подтверждения Операций содержит:

ПОЛЕ	ОПИСАНИЕ
Challenge	Запрос на выполнение аутентификационного испытания

ПОЛЕ	ОПИСАНИЕ
AccessToken	Маркер доступа. Заполняется при <code>IsFinal</code> - true
ExpiresIn	Время жизни AccessToken в секундах. Заполняется при <code>IsFinal</code> - true
IsFinal	Является ли данный ответ последним в процессе подтверждения.
IsError	Содержит ли данный ответ ошибку обработки запроса. Заполняется при <code>IsFinal</code> - false
Error	Ошибка обработки запроса. Заполняется при <code>IsFinal</code> - false
ErrorDescription	Подробное описание ошибки обработки запроса

Поле `Challenge` содержит:

ПОЛЕ	ОПИСАНИЕ
Title	Текст, который вызывающая система может отобразить пользователю в своём интерфейсе
TextChallenge	Дополнительные данные для подтверждения операции

В поле `TextChallenge` содержится:

ПОЛЕ	ОПИСАНИЕ
Image	QR-код для Offline подтверждения операции
RefID	Идентификатор транзакции, созданной на Сервисе Подтверждения Операций
ExpiresIn	Срок действия транзакции, созданной на Сервисе Подтверждения Операций
AuthnMethod	Идентификатор метода используемый для подтверждения транзакции

**Примечание**

`RefId` - Идентификатор транзакции, созданной на Сервисе Подтверждения Операций. Идентификатор необходимо будет использовать при последующих обращениях на конечную точку `/confirmation`.

**Примечание**

При обработке ответа Сервиса Подтверждения Операций вызывающее приложение должно смотреть на значение двух флагов: `IsFinal` и `IsError`.

Если получен ответ с `IsError` - true, то дальнейшее подтверждение транзакции не возможно.

Если получен ответ с `IsFinal` - false, то подтверждение транзакции ещё не завершено.

Получение callback о подтверждении операции

Если в первом запросе к Сервису Подтверждения Операций пользователь указал `CallbackUri`, то после подтверждения операции на мобильном устройстве пользователя придёт оповещение о завершении транзакции.

Сообщение о завершении транзакции содержит:

- **Result** - результат подтверждения транзакции (success или failed)
- **TransactionId** - идентификатор транзакции на Сервисе Подтверждения операций (**RefId**)
- **Error** - код ошибки
- **ErrorDescription** - описание ошибки

## Примеры ответа на CallbackUri

Оповещение о подтверждении операции:

```
{
 "Result": "success",
 "TransactionId": "aa1a4a5d-bb4d-456b-87da-31818604fcd8",
 "Error": "",
 "ErrorDescription": null}
```

Оповещение об отказе (пользователь в мобильном приложении Отказался от подтверждения операции):

```
{
 "Result": "failed",
 "TransactionId": "e19de724-0a4f-4d29-903a-0d3c93735dad",
 "Error": "all_actions_declined",
 "ErrorDescription": "Все действия входящие в состав операции 'e19de724-0a4f-4d29-903a-0d3c93735dad' были отклонены."
}
```

Оповещение об истечении срока действия транзакции.

```
{
 "Result": "failed",
 "TransactionId": "bc0ffdee-7143-439f-bf6b-d1400725d8f1",
 "Error": "transaction_expired",
 "ErrorDescription": "Срок действия транзакции истёк"
}
```

Если пользователь подтвердил операцию на мобильном устройстве, необходимо обратиться на Сервис Подтверждения Операций для получения нового AccessToken. В запросе передаётся идентификатор RefId.

## Пример запроса

```
POST https://host/STS/v2.0/confirmation HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIJHMDf ... tz5Wti-H8CeYxcwB6A
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 212
Expect: 100-continue
```

```
{
 "Resource" : "urn:cryptopro:dss:signserver:SignServer",
 "ClientId": "oauth-client-id",
 "ClientSecret": "oauth-client-secret",
 "ChallengeResponse":
 {
 "TextChallengeResponse":
 [{ "RefId": "e7207ff7-5456-4943-bebf-a7cc624aadaa" }]
 }
}
```

### Пример ответа

Ответ Сервиса Подтверждения Операций будет содержать новый AccessToken, который необходимо использовать для получения подписанного документа.

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 2215
Content-Type: application/json; charset=utf-8
Expires: -1

{
 "AccessToken": "eyJ0eXAiOiJKV1QiLCB... YF3oFlBxXsK7iCkM81jQIwoldwtB5_Gw",
 "ExpiresIn": 600,
 "IsFinal": true,
 "IsError": false
}
```

## Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_transaction	1. Срок действия транзакции истёк 2. Передан неверный идентификатор транзакции (RefId)
400	transaction_pending	У пользователя есть неподтвержденная транзакция.

## Проверка статуса подтверждения операции

Если интегрируемая система не передала адрес `callback` в первом запросе к `/v2.0/confirmation`, то статус подтверждения операции она сможет узнать периодически опрашивая Сервис Подтверждения Операция, ожидая завершения подтверждения транзакции (флаг `IsFinal` = true).

## Пример запроса

```
POST https://host/STS/v2.0/confirmation HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1 .. mXqvC5_3W244A
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 212
Expect: 100-continue

{
 "Resource" : "urn:cryptopro:dss:signserver:SignServer",
 "ClientId": "oauth-client-id",
 "ClientSecret": "oauth-client-secret",
 "ChallengeResponse":
 {
 "TextChallengeResponse":
 [{"RefId": "de34f120-55d5-4f3e-8e7a-b15c1444d747"}]}
}
```

## Примеры ответов

Если подтверждение не завершено, то `IsFinal` - false



```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 352
Content-Type: application/json; charset=utf-8
Expires: -1

{
 "Challenge":
 {
 "Title":{"Value":""},
 "TextChallenge":
 [{
 "AuthnMethod":"http://dss.cryptopro.ru/identity/authenticationmethod/mobile",
 "RefID":"de34f120-55d5-4f3e-8e7a-b15c1444d747"
 }],
 "ContextData":{"RefID":"de34f120-55d5-4f3e-8e7a-b15c1444d747"}},
 "IsFinal":false,
 "IsError":false
 }
}
```

Если в ответе `IsFinal` - true, то Сервис вернул новый AccessToken.

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 2215
Content-Type: application/json; charset=utf-8
Expires: -1

{
 "AccessToken":"eyJ0eXAiOiJKV1QiLC... 5b1T6H1ytuWztMPGfz-0w",
 "ExpiresIn":600,
 "IsFinal":true,
 "IsError":false
}
```

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_transaction	1. Срок действия транзакции истёк 2. Передан неверный идентификатор транзакции ( <code>RefId</code> )

Получение подписанного документа на Сервисе Подписи

Примечание

Данный вызов используется в сценарии **синхронной подписи**.

Описание API Сервиса Подписи - [Конечная точка Signature](#)

После получения callback и нового access\_token вызывающая система обращается на Сервис Подписи непосредственно для подписания документов

Внимание!

В заголовке `Authorization` необходимо передать access\_token, полученный на предыдущем шаге.

Пример запроса

```
POST /SignServer/rest/api/v2/signature HTTP/1.1
Host: dss-sdk.cryptopro.ru
Content-Type: application/json
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbG... SaA_hBR82j4nZa0g
Content-Length: 2

{}
```

Сервис вернёт результат подписания [OperationInfo](#). Поле `Result` содержит результат подписи каждого из документов. Если статус `Status` документа `Completed`, то ID подписанного документа сохранено в поле `RefId`.

Загрузить подписанный документ можно с помощью метода [/content](#)

### Пример ответа

```
{
 "Operation": {
 "Id": "6ec9d54a-7f4d-4579-abe9-660ce241e185",
 "Result": {
 "ProcessedDocuments": [
 {
 "RefId": "f7dbd740-9fad-4aea-88a7-39dfcc81c0ab",
 "OriginalRefId": "37bcc23f-73d8-4cb9-8b54-9b47ad03fd52",
 "Content": null,
 "Status": "Completed",
 "Error": null,
 "ErrorDescription": null
 }
]
 },
 "Status": "Completed",
 "Error": null,
 "ErrorDescription": null,
 "ExpirationDate": 1582121878
 }
}
```

### Пример ответа

В случае ошибки подписания документа соответствующий ему элемент списка `ProcessedDocuments` будет иметь Status `Error`

```

{
 "Operation": {
 "Id": "6ec9d54a-7f4d-4579-abe9-660ce241e185",
 "Result": {
 "ProcessedDocuments": [
 {
 "RefId": null,
 "OriginalRefId": "c27464c5-acd4-4c87-81af-06d7b72fe14a",
 "Content": null,
 "Status": "Error",
 "Error": "Ошибка при подписи документа:\r\nтип подписи CMS, параметры: [DocumentInfo, test.pdf], [CADESType, BES].\r\nВложенное сообщение:\r\nЗатребованный тип CAdES подписи не поддерживается: [Присоединённая подпись в потоковом режиме].",
 "ErrorDescription": null
 }
]
 },
 "Status": "Completed",
 "Error": null,
 "ErrorDescription": null,
 "ExpirationDate": 1582121878
 }
}

```

# Выпуск сертификата пользователя

Раздел содержит руководство разработчика по выпуску сертификата пользователя. В разделе приведены основные сценарии использования, примеры HTTP-запросов и ответов сервисов DSS.

В разделе приведены сценарии выпуска сертификата:

- [Оператором DSS](#)
- [Пользователем DSS](#)

Перед началом интеграции Администратору DSS необходимо:

- Выпустить и зарегистрировать на DSS сертификат аутентификации Оператора DSS
- Зарегистрировать OAuth-клиента
- Подключить к Сервису Подписи модуль взаимодействия с УЦ

Общий подход для Пользователей и Операторов при выпуске сертификата:

1. Аутентификация на Центре Идентификации
2. Получение параметров выпуска запроса на сертификат
3. Создание запроса на сертификат
4. Подтверждение создания запроса на сертификат (для пользователей)
5. Установка сертификата

## Примечание

В примере рассматривается выпуск сертификата пользователя через Сторонний УЦ. Созданный на шаге 3 запрос на сертификат (PKCS#10) Пользователь или Оператор самостоятельно передать в УЦ для выпуска Сертификата. Выпущенный сертификат должен быть установлен на Сервисе Подписи через API.

## Создание запроса на сертификат

Параметры выпуска запроса на сертификат можно получить из Политики Сервиса Подписи (метод [/policy](#)). Политика Сервиса Подписи содержит:

- Список параметров Удостоверяющих Центров, подключенных к DSS
- Список криптопровайдеров, подключенных к DSS

Каждый элемент списка [параметров УЦ](#) содержит:

- Идентификатор Удостоверяющего Центра
- Тип Удостоверяющего Центра
- Шаблон различительного имени (Distinguished Name)
- Список шаблонов сертификатов
- Отображаемое имя

В интерфейсе интегрируемой системы должна быть возможность выбора Удостоверяющего Центра, для которого будет создан запрос на сертификат. Для каждого Удостоверяющего Центра Сервис Подписи передаёт отображаемое имя (DSSCAPolicy -> `Name`), которое может быть показано пользователю.

Для выбранного пользователем Удостоверяющего Центра в интерфейсе интегрируемой системы должна отображаться форма для заполнения Идентифицирующих данных. Форма составляется в соответствии с шаблоном имени (DSSCAPolicy -> `NamePolicy`). У каждого компонента имени в шаблоне есть отображаемое имя (`Name`), строковый идентификатор (`StringIdentifier`) и требование к заполнению (`IsRequired`).

Так же на форме создания запроса должен быть отображен список шаблонов сертификатов (`EkuTemplates`). Каждый шаблон сертификата имеет отображаемое имя.

Если Политика Сервиса Подписи содержит более одного криптопровайдера, то необходимо предоставить пользователю возможность выбора.

Данные с формы передаются в метод [/requests](#) для создания запроса на сертификат:

- Идентификатор Удостоверяющего Центра
- Различительное имя
- Шаблон сертификата
- ПИН-код на закрытый ключ (опционально)
- Идентификатор криптопровайдера (опционально)

Данные передаются в структуре [CertificateRequest](#).

**Идентификатор Удостоверяющего Центра** (`AuthorityId`) является константой. Он может быть получен от Администратора DSS и зафиксирован в настройках интегрируемой системы.

### Примечание

Если Удостоверяющий Центр с заданным идентификатором отсутствует в Политике Сервиса Подписи, то либо он недоступен в данный момент, либо был отключен Администратором DSS. Для выяснения причин недоступности Удостоверяющего Центра следует обратиться к Администратору DSS.

**Различительное имя** может быть передано в двух форматах:

- Список пар oid:value (`DistinguishedName`)
- Строковое представление (`RawDistinguishedName`)

Объектные идентификаторы (**OID**) компонентов имени указаны в шаблоне имени.

### Примечание

Строковое представление различительного имени кодируется согласно [RFC 1779](#).

**Шаблон сертификата** представляет собой набор объектных идентификаторов, которые попадут в расширение Enhanced Key Usage (EKU) запроса на сертификат, или идентификатор шаблона сертификата КриптоПро УЦ 2.0, который попадёт в расширение Certificate Template (1.3.6.1.4.1.311.21.7).

Шаблон передаётся через разные поля запроса на сертификат в зависимости от типа:

- Enhanced key usage - передаётся в дополнительных параметрах запроса `Parameters` в ключе `EkuString` в формате oid1,oid2,...,oidN.

### Примечание

Данный шаблон используется при создании запроса на сертификат к Удостоверяющему Центру типа 0 (КриптоПро УЦ 1.5) и 2 (Сторонний УЦ).

- Certificate Template - передаётся в параметре `Template` запроса на сертификат.

### Примечание

Данный шаблон используется при создании запроса на сертификат к Удостоверяющему Центру типа 1 (КриптоПро УЦ 2.0) и 2 (Сторонний УЦ).

**Идентификатор криптопровайдера** должен быть задан, если в Политике Сервиса Подписи доступно более одного криптопровайдера. Идентификатор криптопровайдера (DSSCSPPolicy -> `GroupId`) передаётся в дополнительных параметрах запроса в ключе `GroupId`

Создание запроса на сертификат с подтверждением при помощи вторичной аутентификации

При создании запроса на сертификат с подтверждением при помощи вторичной аутентификации требуется выполнить следующую последовательность действий (шагов):

- [Создание транзакции на Сервисе Подписи](#)
- [Подтверждение транзакции на Сервисе Подтверждения Операций](#)
- [Получение результата операции на Сервисе Подписи](#)

При этом в массив параметров транзакции метода `/transactions` должны быть отображены следующие поля запроса на сертификат:

CERTIFICATE REQUEST	ПАРАМЕТРЫ ТРАНЗАКЦИИ
AuthorityId	CAId
PinCode	Не используется
Template	CertTemplateOid
DistinguishedName	Не используется
RawDistinguishedName	CertSubjectName
Parameters -> EkuString	EkuString
Parameters -> GroupId	GroupId

**Примечание**

При создании запроса на сертификат с подтверждением с подтверждением при помощи вторичной аутентификации различительное имя может быть передано только в строковом представлении.

Примеры запросов

Пример запроса с указанием различительного имени в строковом представлении:

```
POST https://host/SignServer/rest/api/requests HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cGE6YW50eXBvc2V0bzI6bnRniEg
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 153
Expect: 100-continue

{
 "AuthorityId":11,
 "PinCode": "",
 "RawDistinguishedName": "CN=dssUser,C=RU",
 "Parameters":
 {"EkuString": "1.2.643.2.2.34.2,1.2.643.2.2.34.4,1.3.6.1.5.5.7.3.2"}
}
```

Пример запроса с указанием различительного имени в виде набора компонентов:

```
POST https://host/SignServer/rest/api/requests HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJ... PhYmXscTmwGkD8b1SWy0nYQ
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 169
Expect: 100-continue

{
 "AuthorityId":11,
 "PinCode":"",
 "DistinguishedName":{"2.5.4.3":"dssUser","2.5.4.6":"RU"},
 "Parameters":
 {
 "EkuString":"1.2.643.2.2.34.2,1.2.643.2.2.34.4,1.3.6.1.5.5.7.3.2"
 }
}
```

Пример запроса с указанием шаблона сертификата:

```
POST https://host/SignServer/rest/api/requests HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV... Ysj1GpIVmR2hw
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 135
Expect: 100-continue

{
 "AuthorityId":11,
 "PinCode":"",
 "Template":"1.3.6.1.5.5.7.3.2",
 "DistinguishedName":{"2.5.4.3":"dssUser","2.5.4.6":"RU"},
 "Parameters":{}}
```

Пример ответа:

```
HTTP/1.1 200 OK
Content-Length: 723
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
Date: Tue, 04 Sep 2018 13:35:02 GMT

{
 "CertificateType":"ServerSide",
 "Base64Request":"MIIBQDCB8AIBADAfMQswCQYD... iOibLabDHZ2VY1G8CsaxjE",
 "CertificateAuthorityID":11,
 "CADisplayName":null,
 "DistName":"CN=dssUser, C=RU",
 "Subject":"dssUser",
 "Status":"PENDING",
 "ID":22,
 "CARquestID":null,
 "CertificateID":0,
 "RequestType":"Certificate",
 "GroupID":"e8e67f9e-7eed-4116-ad98-20582e4d766e"}
```

Запрос на сертификат **с подтверждением** с подтверждением при помощи вторичной аутентификации:

```
POST https://host/SignServer/rest/api/transactions HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cGE6YW50eXBkaWkiOiJkZWYyZjpoNl9RgQLA...
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 200
Expect: 100-continue
```

```
{
 "OperationCode":16,
 "Parameters":
 [
 { "Name":"CertSubjectName", "Value":"CN=dssUser,C=RU"},
 { "Name":"CAId", "Value":"11"},
 { "Name":"EkuString", "Value":"1.2.643.2.2.34.2,1.2.643.2.2.34.4,1.3.6.1.5.5.7.3.2"}
]
}
```

## Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	pending_requests_exist	У пользователя есть необработанный запрос на сертификат (статус PENDING).

## Обработка ответа Сервиса Подписи

При успешном создании запроса на сертификат Сервис Подписи в ответе вернёт структуру [DSSCertRequest](#).

Дальнейшее поведение пользователя зависит от значения поля `Status` в структуре `DSSCertRequest` и типа УЦ, на котором создавался запрос на сертификат.

**ACCEPTED** - запрос на сертификат принят и обработан УЦ. В данном случае в поле CertificateID будет записан идентификатор выпущенного сертификата.

**REGISTRATION** - запрос на сертификат принят в КриптоПро УЦ 2.0 и находится на этапе регистрации пользователя УЦ. В зависимости от настроек подключения DSS к КриптоПро УЦ 2.0, необходимо:

- ожидать одобрения запроса на сертификат Администратором УЦ;
- одобрить запрос Оператором DSS.

**PENDING** - запрос на сертификат находится в обработке. Если запрос отправлен на КриптоПро УЦ 2.0, то в зависимости от настроек подключения DSS к КриптоПро УЦ 2.0 необходимо:

- ожидать одобрения запроса на сертификат Администратором УЦ;
- одобрить запрос Оператором DSS.

Если запрос создавался через "Сторонний Удостоверяющий Центр", необходимо:

- скачать запрос на сертификат по идентификатору `/requests;`
- передать запроса на сертификат в УЦ;
- выпущенный сертификат установить в DSS.

Запрос на сертификат (PKCS#10) в формате Base64 содержится в поле `Base64Request` структуры `DSSCertRequest`.

**REJECTED** - запрос отклонён. Дальнейшая обработка запроса невозможна. Для выяснения причин отклонения запроса необходимо обратиться к Администратору УЦ.

## Выпуск сертификата Оператором DSS

## Аутентификация Оператора DSS на Центре Идентификации

Аутентификация Оператора DSS производится по сертификату (HTTPS с аутентификацией клиента).



Для получения AccessToken используется OAuth сценарий с использованием кода авторизации. Подробная информация по протоколу аутентификации: [The OAuth 2.0 Authorization Framework](#)

Администратор DSS должен предварительно настроить OAuth клиента на сервере DSS:

- создав нового клиента:

```
Add-DssClient -Identifier testClient -Name testClient -Description "Test Client Description" -RedirectUri urn:ietf:wg:oauth:2.0:oob:auto -AllowedFlow ResourceOwner,AuthorizationCode
```

- изменив настройки существующего клиента:

```
Set-DssClient -ClientId testClient -RedirectUri urn:ietf:wg:oauth:2.0:oob:auto -AllowedFlow ResourceOwner,AuthorizationCode
```

## Примечание

Значение `RedirectUri` **urn:ietf:wg:oauth:2.0:oob:auto** говорит серверу DSS о том, что AccessToken необходимо вернуть непосредственно в ответе на запрос клиента. Данное значение используется в тех случаях, когда для клиента трудозатратно открыть слушателя на другом URL.

Последовательность шагов:

1. [Инициация аутентификации](#), путём отправки запроса на конечную точку /authorize/certificate по HTTPS с аутентификацией по сертификату.
2. [Получение кода авторизации](#).
3. [Получение AccessToken по коду авторизации](#).
4. [Получение делегирующего AccessToken](#).

AccessToken, полученный на шаге 3, позволит Оператору DSS получить [Политику Сервиса Подписи](#).

Для выполнения действий от имени пользователя на Сервисе Подписи необходимо получить [делегирющий AccessToken](#). Делегирующий AccessToken позволит Оператору DSS выпустить сертификат пользователя, просмотреть список сертификатов и запросов пользователя и т.п.

## Инициация аутентификации

Конечная точка для аутентификации Оператора DSS: `/authorize/certificate`

Параметры запроса:

- `redirect_uri` – зарегистрированный на Центре Идентификации адрес возврата (по этому адресу будет возвращён запрошенный код авторизации). Допустимые значения данного параметра сохраняются в ЦИ на этапе регистрации клиента.
- `response_type` – в данном сценарии всегда должен иметь значение **code**.
- `scope` – области использования маркера. Должен содержать значение **dss**.
- `resource` – идентификатор Сервиса Подписи.

## Примеры запросов

```
GET https://host/STS/oauth/authorize/certificate?
client_id=testClient&response_type=code&scope=dss&redirect_uri=urn:ietf:wg:oauth:2.0:oob:auto&resource=urn:cr
uptopro:dss:signserver:signserver
Host: host
Connection: Keep-Alive
```

## Получение кода авторизации

В случае успешной аутентификации

- ответ сервера будет иметь статус HTTP 302
- В заголовке `Location` будет содержаться адрес получения AccessToken.

Т.е. в примере используется специальное значение `redirect_uri`, то клиенту необходимо из заголовка `Location` извлечь значение параметра **code**. Значение параметра **code** будет использовано для получения AccessToken на следующем шаге.

### Пример ответа

```
HTTP/1.1 302 Found
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Location: urn:ietf:wg:oauth:2.0:oob:auto?code=65e4322a9751cf9ba43012692ce02ec1
Date: Fri, 07 Sep 2018 10:30:24 GMT
Content-Length: 0
```

### Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_client	OAuth-клиент не зарегистрирован или неверно указан clientID
400	unauthorized_client	OAuth-клиент использует незарегистрированный сценарий аутентификации (Flow)
400	invalid_request	Неверно сформирован параметр resource
500	An error has occurred	1. Проверяющая сторона с идентификатором resource не зарегистрирована.

### Получение AccessToken

Для получения маркера доступа используется конечная точка `oauth/token`.

Параметры запроса:

- `grant_type` - тип разрешения, в данном сценарии равен **authorization\_code**.
- `code` – код авторизации, полученный на предыдущем шаге.
- `resource` – идентификатор Сервиса Подписи.
- `redirect_uri` – зарегистрированный на Центре Идентификации адрес возврата.

В заголовке Authorization HTTP-запроса клиент должен передать идентификатор OAuth-клиента и секрет (если используется): Authorization: Basic Base64(<client\_id>:<secret>)

### Пример запроса

```
POST https://host/STS/oauth/token HTTP/1.1
Authorization: Basic dGVzdENSaWVudDo=
Content-Type: application/x-www-form-urlencoded
Host: host
Content-Length: 180
Expect: 100-continue

grant_type=authorization_code&code=65e4322a9751cf9ba43012692ce02ec1&redirect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aoob%3Aauto&resource=urn%3Acryptopro%3Adss%3Asignserver%3Asignserver
```

В случае успешной аутентификации ответ будет содержать:

- `access_token` - AccessToken, выпущенный Центром Идентификации DSS
- `token_type` - Тип токена

- `expires_in` - Время жизни токена в секундах

Значение параметра `access_token` необходимо будет использовать при обращениях к Сервису Подписи.

## Примечание

Данный `access_token` не даёт право Оператору DSS выполнять операции на Сервисе Подписи от имени пользователей.  
`access_token` может быть использован для получения [Политики Сервиса Подписи](#).

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 2268
Content-Type: application/json; charset=utf-8
Expires: -1
```

```
{"access_token":"eyJ0eXAiOiJKV1Q...LnS1sAundSE1hh3A5n8W7lhPSM4z_VA","expires_in":300,"token_type":"Bearer"}
```

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_client	OAuth-клиент не зарегистрирован или неверно указан clientID
400	unauthorized_client	OAuth-клиент использует незарегистрированный сценарий аутентификации (Flow)
400	invalid_request	Неверно сформирован параметр resource
400	invalid_grant	Невалидный код авторизации.
500	An error has occurred	1. Проверяющая сторона с идентификатором resource не зарегистрирована.

## Получение делегирующего AccessToken

Для получения AccessToken для делегирования используется конечная точка `oauth/token`. Подробная информация по протоколу получения AccessToken для делегирования: [OAuth 2.0 Token Exchange](#).

Параметры запроса:

- `grant_type` - тип разрешения, в данном сценарии равен **urn:ietf:params:oauth:grant-type:token-exchange**.
- `resource` – идентификатор Сервиса Подписи.
- `actor_token` - AccessToken, полученный на предыдущем шаге
- `actor_token_type` – тип маркера доступа, должен иметь значение **urn:ietf:params:oauth:token-type:jwt**.
- `subject_token_type` – тип маркера доступа, должен иметь значение **urn:ietf:params:oauth:token-type:jwt**.
- `subject_token` – неподписанный JWT-токен, содержащий логин управляемого пользователя.

В декодированном виде `subject_token` имеет вид:

```
{
 "alg": "none",
 "typ": "JWT"
}.
{
 "unique_name": "mydss",
 "nbf": 1488312889,
 "exp": 1488316489,
 "iat": 1488312889
}
.
```

Пример кодирования JWT-токена можно посмотреть по [ссылке](#).

Первая часть (до точки) называется header, вторая – payload. Для получения закодированного значения необходимо выполнить следующее преобразование:

```
Base64UrlEncode(UTF8GetBytes(header)) + "." + Base64UrlEncode(UTF8GetBytes(payload)) + "."
```

### Внимание!

Символ "." в конце получившегося значения является обязательным.

В заголовке Authorization HTTP-запроса клиент должен передать идентификатор OAuth-клиента и секрет (если используется): Authorization: Basic Base64(<client\_id>:<secret>)

### Пример запроса

```
POST https://host/STS/oauth/token HTTP/1.1
Authorization: Basic dGVzdENSaWVudDo=
Content-Type: application/x-www-form-urlencoded
Host: host
Content-Length: 2529
Expect: 100-continue

grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Atoken-exchange&actor_token=eyJ0eXAiOiJKV1QiLCJhbGc ...
E1hh3A5n8W7lhPSM4z_VA&actor_token_type=urn%3Aietf%3Aparams%3Aoauth%3Atoken-
type%3Ajwt&subject_token=e30.eyJ1bm1xdWVfbmFtZSI6Im15ZHNzIn0.&subject_token_type=urn%3Aietf%3Aparams%3Aoauth%
3Atoken-type%3Ajwt&resource=urn%3Acryptopro%3Adss%3Asignserver%3Asignserver
```

В случае успешной аутентификации ответ будет содержать:

- `access_token` - делегирующий AccessToken, выпущенный Центром Идентификации DSS
- `token_type` - Тип токена
- `expires_in` - Время жизни токена в секундах

Значение параметра `access_token` необходимо будет использовать при обращениях к Сервису Подписи.

### Пример ответа

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 2475
Content-Type: application/json; charset=utf-8
Expires: -1

{"access_token":"eyJ0eXAiOiJKV1QiLCJhbGc ... h0X-
7aUneD_po8p5uD3nJGQ5VlHJHw4vA", "expires_in":300, "token_type":"Bearer"}
```

### Получение Политики Сервиса Подписи

#### Пример запроса

```
GET https://host/SignServer/rest/api/policy HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJK ... K0ePGIpg
Host: host
```

#### Пример ответа

```
HTTP/1.1 200 OK
Content-Length: 4821
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
```

Date: Mon, 03 Sep 2018 09:21:40 GMT

```
{
 "CAPolicy":
 [{
 "ID":11,
 "Name":"Out of Band",
 "Active":true,
 "AllowUserMode":false,
 "SNChangesEnable":true,
 "NamePolicy":
 [
 {"IsRequired":false,"Order":2,"OID":"1.2.840.113549.1.9.1","Name":"E-Mail","Value":null,"StringIdentifier":"E"},
 {"IsRequired":false,"Order":8,"OID":"1.2.643.3.131.1.1","Name":"ИНН","Value":null,"StringIdentifier":"INN"},
 {"IsRequired":false,"Order":4,"OID":"2.5.4.7","Name":"Населенный пункт","Value":null,"StringIdentifier":"L"},
 {"IsRequired":false,"Order":7,"OID":"1.2.643.100.1","Name":"ОГРН","Value":null,"StringIdentifier":"OGRN"},
 {"IsRequired":false,"Order":5,"OID":"2.5.4.10","Name":"Организация","Value":null,"StringIdentifier":"O"},
 {"IsRequired":false,"Order":6,"OID":"2.5.4.11","Name":"Подразделение","Value":null,"StringIdentifier":"OU"},
 {"IsRequired":false,"Order":3,"OID":"2.5.4.8","Name":"Регион","Value":null,"StringIdentifier":"S"},
 {"IsRequired":false,"Order":9,"OID":"2.5.4.6","Name":"Страна","Value":null,"StringIdentifier":"C"},
 {"IsRequired":true,"Order":1,"OID":"2.5.4.3","Name":"Общее имя","Value":null,"StringIdentifier":"CN"}
],
 "EKUTemplates":
 {
 "Временный сертификат администратора УЦ":["1.2.643.2.2.34.2","1.2.643.2.2.34.4","1.3.6.1.5.5.7.3.2"],
 "Временный сертификат оператора УЦ":["1.2.643.2.2.34.2","1.2.643.2.2.34.5","1.3.6.1.5.5.7.3.2"],
 "Временный сертификат пользователя УЦ":["1.2.643.2.2.34.2","1.3.6.1.5.5.7.3.2"],
 "Временный сертификат пользователя УЦ1":["1.2.643.2.2.34.6","1.3.6.1.5.5.7.3.2"],
 "Сертификат пользователя УЦ":["1.2.643.2.2.34.6","1.3.6.1.5.5.7.3.2"]
 },
 "CAType":"DSSOutOfBandEnroll",
 "ValidationMode":"CertificateAuthority"
 }],
 "CSPsPolicy":
 [
 {
 "ID":"67e6f39d-c6c5-4ce6-9535-4e22bae84786",
 "GroupID":"e8e67f9e-7eed-4116-ad98-20582e4d766e",
 "TypeID":2,
 "ProviderName":"Crypto-Pro GOST R 34.10-2001 Cryptographic Service Provider",
 "ProviderType":75,"KeyLength":512,"HashAlgorithms":["GOST R 34.11-94"],
 "Description":"GOST 2001"
 },
 {
 "ID":"60e9912c-68a8-4608-b1c2-0c6a074456e8",
 "GroupID":"648092d3-46a9-422a-9240-d32c58cc498b",
 "TypeID":2,
 "ProviderName":"Crypto-Pro GOST R 34.10-2012 Cryptographic Service Provider",
 "ProviderType":80,
 "KeyLength":512,
 "HashAlgorithms":["GR 34.11-2012 256"],
 "Description":"GOST 2012"
 }
],
 "ActionPolicy":[{"DisplayName":"Выпуск маркера (вход в ЦИ)","Uri":"http://dss.cryptopro.ru/identity/claims/action/Issue","Action":"Issue","MfaRequired":false},
```

```
{
 "DisplayName": "Подпись документа",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/SignDocument",
 "Action": "SignDocument",
 "MfaRequired": true
}, {
 "DisplayName": "Подпись пакета документов",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/SignDocuments",
 "Action": "SignDocuments",
 "MfaRequired": false
}, {
 "DisplayName": "Расшифрование документа",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/DecryptDocument",
 "Action": "DecryptDocument",
 "MfaRequired": false
}, {
 "DisplayName": "Создание запроса на сертификат",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/CreateRequest",
 "Action": "CreateRequest",
 "MfaRequired": false
}, {
 "DisplayName": "Смена пин-кода закрытого ключа",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/ChangePin",
 "Action": "ChangePin",
 "MfaRequired": false
}, {
 "DisplayName": "Обновление сертификата",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/RenewCertificate",
 "Action": "RenewCertificate",
 "MfaRequired": false
}, {
 "DisplayName": "Отзыв сертификата",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/RevokeCertificate",
 "Action": "RevokeCertificate",
 "MfaRequired": false
}, {
 "DisplayName": "Приостановление действия сертификата",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/HoldCertificate",
 "Action": "HoldCertificate",
 "MfaRequired": false
}, {
 "DisplayName": "Возобновление действия сертификата",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/UnholdCertificate",
 "Action": "UnholdCertificate",
 "MfaRequired": false
}, {
 "DisplayName": "Удаление сертификата",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/DeleteCertificate",
 "Action": "DeleteCertificate",
 "MfaRequired": false
}, {
 "DisplayName": "Доступ к закрытому ключу",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/PrivateKeyAccess",
 "Action": "PrivateKeyAccess",
 "MfaRequired": false
}],
"PinCodeMode": "Allow",
"TspServices": [
 {
 "Name": "TestTSP",
 "Title": "TestTSP",
 "Url": "http://TEST-DSS-W8R2/TSP/tsp.srf"
 }
],
"TransactionConfirmation": "NotSet",
"AllowedSignatureTypes": ["GOST3410", "CMS", "CAES", "XMLDSig", "MSOffice", "PDF"]
}
```

## Создание запроса на сертификат

### Пример запроса

Согласно параметрам УЦ из Политики Сервиса Подписи Оператор формирует запроса на сертификат.

```
POST https://host/SignServer/rest/api/requests HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1 .. so0AiQOhtllgg
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 167
Expect: 100-continue

{"AuthorityId":11,"PinCode":"","DistinguishedName":{"2.5.4.3":"mydss","2.5.4.6":"RU"},"Parameters":{"EkuString":"1.2.643.2.2.34.2,1.2.643.2.2.34.4,1.3.6.1.5.5.7.3.2"}}
```

### Пример ответа

```
HTTP/1.1 200 OK
Content-Length: 719
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
```

```
{
 "CertificateType": "ServerSide",
 "Base64Request": "MIIBPjCB7gIBADAdMQswCQYD ... 9MmKj3pHKCuhwuZCfzU+gKLuzWrQ==",
 "CertificateAuthorityID": 11,
 "CADisplayName": null,
 "DistName": "CN=mydss, C=RU",
 "Subject": "mydss",
 "Status": "PENDING",
 "ID": 23,
 "CARequestID": null,
 "CertificateID": 0,
 "RequestType": "Certificate",
 "GroupID": "e8e67f9e-7eed-4116-ad98-20582e4d766e"
}
```

## Установка сертификата

Сервер вернул запрос на сертификат в поле `Base64Request`. Запрос на сертификат необходимо передать в Удостоверяющий Центр для выпуска сертификата.

## Пример запроса

```
POST https://host/SignServer/rest/api/certificates HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIU2 ... PYErWVsOP9IM7oahdJ3iRg
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 1154
Expect: 100-continue

{"Certificate": "MIIDHTCCAasyAwIBAgITEgAs1guTuGxFdbofF ... qJNHeSr2QvryU0IPn1jRE/VZnuEMf80PZ1\r\n"}
```

## Пример ответа

HTTP/1.1 200 OK  
Content-Length: 1491  
Content-Type: application/json; charset=utf-8  
Server: Microsoft-IIS/7.5

```
{
 "CertificateType": "ServerSide",
 "ID": 14,
 "DName": "CN=mydss, C=RU",
 "CertificateBase64": "MIIDHTCCAsygAwIBAgITEgAs ... NHeSr2QvryU0IPn1jRE/VZnuEMf80PZ1",
 "Status": {
 "Value": "ACTIVE",
 "RevocationInfo": null,
 "PinCode": null,
 "ActiveCertId": 0
 },
 "IsDefault": false,
 "CertificateAuthorityID": 11,
 "CspID": "e8e67f9e-7eed-4116-ad98-20582e4d766e",
 "HashAlgorithms": ["GOST R 34.11-94"],
 "ProviderName": null,
 "ProviderType": 0,
 "PrivateKeyNotBefore": null,
 "PrivateKeyNotAfter": null,
 "HasPin": false,
 "FriendlyName": ""
}
```

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_certificate_format	Сертификат имеет неверный формат. Например, сертификат передан с заголовками.

Выпуск сертификата Пользователем DSS

Аутентификация пользователя на Центре Идентификации

В примере рассматривается авторизация с использованием учётных данных пользователя (логин/пароль). Подробная информация по протоколу аутентификации: [The OAuth 2.0 Authorization Framework](#)

Параметры запроса:

- grant\_type - тип разрешения, в данном сценарии равен **password**.
- password – пароль пользователя.
- resource – идентификатор Сервиса Подписи.

В заголовке Authorization HTTP-запроса клиент должен передать идентификатор OAuth-клиента и секрет (если используется): Authorization: Basic Base64(<client\_id>:<secret>)

Примечание

В примере значение параметр **password** оставлено пустым, так как пользователю в качестве первичной аутентификации назначен метод "Только Идентификация"

Примеры запросов



```
POST https://host/STS/oauth/token HTTP/1.1
Authorization: Basic dGVzdENSawVudDo=
Content-Type: application/x-www-form-urlencoded
Host: host
Content-Length: 101
Expect: 100-continue
Connection: Keep-Alive

grant_type=password&username=mydss&password=&resource=urn%3Acryptopro%3Adss%3Asignserver%3Asignserver
```

В случае успешной аутентификации ответ будет содержать:

- `access_token` - AccessToken, выпущенный Центром Идентификации DSS
- `token_type` - Тип токена
- `expires_in` - Время жизни токена в секундах

Значение параметра `access_token` необходимо будет использовать при обращениях к Сервису Подписи и Сервису Подтверждения Операций.

**Пример ответа**

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 2017
Content-Type: application/json; charset=utf-8
Expires: -1

{
 "access_token": "eyJ0eXAiOiJKV...5Wti-H8CeXycwB6A",
 "expires_in": 300,
 "token_type": "Bearer"
}
```

**Типовые ошибки**

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_client	OAuth-клиент не зарегистрирован или неверно указан clientID
400	unauthorized_client	OAuth-клиент использует незарегистрированный сценарий аутентификации (Flow)
400	invalid_request	Неверно сформирован параметр resource
500	An error has occurred	1. Проверяющая сторона с идентификатором resource не зарегистрирована.

**Получение Политики Сервиса Подписи**

**Пример запроса**

```
GET https://host/SignServer/rest/api/policy HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJK...K0ePGIpg
Host: host
```

**Пример ответа**

```
HTTP/1.1 200 OK
Content-Length: 4821
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5
```

Date: Mon, 03 Sep 2018 09:21:40 GMT

```
{
 "CAPolicy":
 [{
 "ID":11,
 "Name":"Out of Band",
 "Active":true,
 "AllowUserMode":false,
 "SNChangesEnable":true,
 "NamePolicy":
 [
 {"IsRequired":false,"Order":2,"OID":"1.2.840.113549.1.9.1","Name":"E-Mail","Value":null,"StringIdentifier":"E"},
 {"IsRequired":false,"Order":8,"OID":"1.2.643.3.131.1.1","Name":"ИНН","Value":null,"StringIdentifier":"INN"},
 {"IsRequired":false,"Order":4,"OID":"2.5.4.7","Name":"Населенный пункт","Value":null,"StringIdentifier":"L"},
 {"IsRequired":false,"Order":7,"OID":"1.2.643.100.1","Name":"ОГРН","Value":null,"StringIdentifier":"OGRN"},
 {"IsRequired":false,"Order":5,"OID":"2.5.4.10","Name":"Организация","Value":null,"StringIdentifier":"O"},
 {"IsRequired":false,"Order":6,"OID":"2.5.4.11","Name":"Подразделение","Value":null,"StringIdentifier":"OU"},
 {"IsRequired":false,"Order":3,"OID":"2.5.4.8","Name":"Регион","Value":null,"StringIdentifier":"S"},
 {"IsRequired":false,"Order":9,"OID":"2.5.4.6","Name":"Страна","Value":null,"StringIdentifier":"C"},
 {"IsRequired":true,"Order":1,"OID":"2.5.4.3","Name":"Общее имя","Value":null,"StringIdentifier":"CN"}
],
 "EKUTemplates":
 {
 "Временный сертификат администратора УЦ":["1.2.643.2.2.34.2","1.2.643.2.2.34.4","1.3.6.1.5.5.7.3.2"],
 "Временный сертификат оператора УЦ":["1.2.643.2.2.34.2","1.2.643.2.2.34.5","1.3.6.1.5.5.7.3.2"],
 "Временный сертификат пользователя УЦ":["1.2.643.2.2.34.2","1.3.6.1.5.5.7.3.2"],
 "Временный сертификат пользователя УЦ1":["1.2.643.2.2.34.6","1.3.6.1.5.5.7.3.2"],
 "Сертификат пользователя УЦ":["1.2.643.2.2.34.6","1.3.6.1.5.5.7.3.2"]
 },
 "CAType":"DSSOutOfBandEnroll",
 "ValidationMode":"CertificateAuthority"
 }],
 "CSPsPolicy":
 [
 {
 "ID":"67e6f39d-c6c5-4ce6-9535-4e22bae84786",
 "GroupID":"e8e67f9e-7eed-4116-ad98-20582e4d766e",
 "TypeID":2,
 "ProviderName":"Crypto-Pro GOST R 34.10-2001 Cryptographic Service Provider",
 "ProviderType":75,"KeyLength":512,"HashAlgorithms":["GOST R 34.11-94"],
 "Description":"GOST 2001"
 },
 {
 "ID":"60e9912c-68a8-4608-b1c2-0c6a074456e8",
 "GroupID":"648092d3-46a9-422a-9240-d32c58cc498b",
 "TypeID":2,
 "ProviderName":"Crypto-Pro GOST R 34.10-2012 Cryptographic Service Provider",
 "ProviderType":80,
 "KeyLength":512,
 "HashAlgorithms":["GR 34.11-2012 256"],
 "Description":"GOST 2012"
 }
],
 "ActionPolicy":[{"DisplayName":"Выпуск маркера (вход в ЦИ)","Uri":"http://dss.cryptopro.ru/identity/claims/action/Issue","Action":"Issue","MfaRequired":false},
```

```
{
 "DisplayName": "Подпись документа",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/SignDocument",
 "Action": "SignDocument",
 "MfaRequired": true
}, {
 "DisplayName": "Подпись пакета документов",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/SignDocuments",
 "Action": "SignDocuments",
 "MfaRequired": false
}, {
 "DisplayName": "Расшифрование документа",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/DecryptDocument",
 "Action": "DecryptDocument",
 "MfaRequired": false
}, {
 "DisplayName": "Создание запроса на сертификат",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/CreateRequest",
 "Action": "CreateRequest",
 "MfaRequired": false
}, {
 "DisplayName": "Смена пин-кода закрытого ключа",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/ChangePin",
 "Action": "ChangePin",
 "MfaRequired": false
}, {
 "DisplayName": "Обновление сертификата",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/RenewCertificate",
 "Action": "RenewCertificate",
 "MfaRequired": false
}, {
 "DisplayName": "Отзыв сертификата",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/RevokeCertificate",
 "Action": "RevokeCertificate",
 "MfaRequired": false
}, {
 "DisplayName": "Приостановление действия сертификата",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/HoldCertificate",
 "Action": "HoldCertificate",
 "MfaRequired": false
}, {
 "DisplayName": "Возобновление действия сертификата",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/UnholdCertificate",
 "Action": "UnholdCertificate",
 "MfaRequired": false
}, {
 "DisplayName": "Удаление сертификата",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/DeleteCertificate",
 "Action": "DeleteCertificate",
 "MfaRequired": false
}, {
 "DisplayName": "Доступ к закрытому ключу",
 "Uri": "http://dss.cryptopro.ru/identity/claims/action/PrivateKeyAccess",
 "Action": "PrivateKeyAccess",
 "MfaRequired": false
}],
"PinCodeMode": "Allow",
"TspServices": [
 {
 "Name": "TestTSP",
 "Title": "TestTSP",
 "Url": "http://TEST-DSS-W8R2/TSP/tsp.srf"
 }
],
"TransactionConfirmation": "NotSet",
"AllowedSignatureTypes": [
 "GOST3410",
 "CMS",
 "CADES",
 "XMLDSig",
 "MSOffice",
 "PDF"
]
}
```

# Аутентификация и подтверждение операций по одноразовым паролям, отправляемых по СМС

При использовании данного метода для подтверждения входа и операций у Пользователя дополнительно будет запрашиваться ввод одноразового пароля, доставляемого в СМС, на телефон Пользователя.

Раздел содержит руководство программиста по интеграции подтверждения операций по СМС:

- [Предварительная настройка Администратором DSS](#) - руководство Администратора по предварительной настройке DSS
- [Назначение и управление аутентификацией](#) - руководство программиста по работе через Оператора DSS
- [Подтверждение операций](#) - руководство программиста по работе через Пользователя DSS

# Предварительная настройка DSS для работы с аутентификацией по одноразовым паролям, отправляемым в СМС

Для использования данного метода вторичной аутентификации и подтверждения операций необходимо:

- 1. Включить метод аутентификации по одноразовым паролям, отправляемым в СМС.

Идентификатор метода аутентификации по одноразовым паролям, отправляемым в СМС:

<http://dss.cryptopro.ru/identity/authenticationmethod/otpviasms>

Для включения данного метода необходимо использовать следующий командлет:

```
Enable-DssAuthenticationMethod -Uri http://dss.cryptopro.ru/identity/authenticationmethod/otpviasms
```

- 2. Зарегистрировать компонент для рассылки сообщений через СМС.

## Вариант 1 Регистрация тестового СМС-плагина.

Сообщения, отправляемые через тестовый плагин, будут сохраняться в файлы в указанной при настройке директории. Данный пример демонстрирует настройку компонента оповещения через тестовый СМС-плагин.

```
Директория для сохранения файлов с сообщениями
$SMSBaseDir = "C:\tempsms\"

Write-Host "Добавление плагина для отправки СМС-сообщений"
Add-DSSSignServerPlugin -PluginTypeName
"CryptoPro.DSS.SmsService.StubPlugin.SmsStub,DSS.SmsService.StubPlugin" -PluginType SMS -Settings
@{"WorkingDirectory" = $SMSBaseDir}

Write-Host "Добавление плагина для форматирования СМС-сообщений"
Add-DSSSignServerPlugin -PluginTypeName "CryptoPro.DSS.MessageFormatter.SMSFormatter,DSS.MessageFormatter" -
PluginType Formatter -Settings @{Header="криптопро DSS."}

Write-Host "Добавление модуля оповещения для отправки СМС-сообщений"
Add-DSSSignServerNotifier -TransportPluginID 1 -FormatterPluginID 2 -NotifierType SMS -Settings
@{"MinQueueSize"="0";"MaxQueueSize"="10000";"TimerInterval"="500";"TTL"="1";"MessageWindow"="50";"ThreadCount"
"="1";"Enabled"="true"}
```

## Вариант 2 Регистрация стандартного плагина DSS.SmsService.SmppPlugin.

Данный плагин предназначен для работы со службой рассылки SMS по протоколу SMPP. Для его регистрации необходимо использовать командлет *Add-DSSSignServerPlugin*.

Тип плагина, указываемый при регистрации в параметре *PluginTypeName*:

*CryptoPro.DSS.SmsService.SmppPlugin.SmppPlugin,DSS.SmsService.SmppPlugin*

Параметры плагина SMPP:

ПАРАМЕТР	ОПИСАНИЕ	ЗНАЧЕНИЕ ПО УМОЛЧАНИЮ	ОБЯЗАТЕЛЬНЫЙ
ServiceAddress	Адрес SMPP сервера	Нет	Да
ServicePort	Порт доступа к серверу SMPP	Нет	Да
SystemId	Логин для доступа к сервису	Нет	Да
SystemPassword	Пароль для доступа к сервису	Нет	Да
Source	Адрес отправителя	null	Нет

### Вариант 3 Создание собственного СМС-плагина и его регистрация.

Модуль отправки коротких текстовых сообщений (СМС) должен представлять собой сборку .NET. Сборка должна содержать единственный публичный класс, реализующий интерфейс *ISmsPlugin*. Интерфейс *ISmsPlugin* описан в сборке *CryptoPro.DSS.Common.dll*. Интерфейс предоставляет метод *SendSms*, предназначенный для отправки SMS, определён следующим образом:

```
public interface ISmsPlugin : IDSSPlugin
{
 void SendSms(string message, string phone);
}
```

Метод `SendSms` принимает следующие параметры:

- `message` – текстовое сообщение;
- `phone` – номер телефона; Список дополнительных параметров задаётся с помощью командлета *Set DSSStsProperties*.

При возникновении ошибки в процессе выполнения метода `SendSms` необходимо сгенерировать исключение `SmsException`.

# Назначение и управление аутентификацией по СМС

В данном разделе описывается последовательность действий для назначения метода аутентификации по одноразовым паролям, отправляемых по СМС. Все запросы выполняются от Оператора DSS.

## Внимание!

Перед началом работы с Центром Идентификации, должны быть выполнены [предварительные настройки сервиса](#) Администратором DSS.

### 1. Назначение номера мобильного телефона пользователю

В первую очередь требуется задать номер мобильного телефона пользователя, на который будут приходить одноразовые пароли.

#### 1.1. Получение информации о добавленных номерах телефона

Один или несколько номеров мобильного телефона пользователя могли быть уже заданы, например, при регистрации, поэтому первым шагом должно быть получение информации об уже добавленных номерах телефона пользователя.

#### Запрос на получение информации о номерах телефона пользователя

```
GET https://host/STS/ums/user/8f4406bf-bd90-4baa-9787-866656907ada/phones HTTP/1.1
```

#### Пример ответа

Метод возвращает список добавленных для этого пользователя номеров мобильных телефонов. Для каждого номера телефона указано подтвержден ли он, используется ли для входа (параметр Primary) и прочая информация об использовании данного телефона.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 187
```

```
[{
 "Type": "PhoneNumber",
 "Contact": "71238889900",
 "Confirmed": true,
 "Primary": true,
 "Notification": true,
 "Usages": []
}]
```

#### 1.2. Добавление нового номера мобильного телефона

Если список номеров мобильного телефона пользователя пуст или среди заданных номеров нет желаемого для вторичной аутентификации номера телефона, то необходимо добавить новый номер мобильного телефона.

## Примечание

Номер мобильного телефона можно задавать в любом формате, с пробелами и без.

#### Запрос на добавление номера телефона

```
POST https://host/STS/ums/user/8f4406bf-bd90-4baa-9787-866656907ada/phones HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 14

"+71238889900"
```

#### Пример ответа

Метод возвращает информацию о только что добавленном номере мобильного телефона.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 110

{
 "Type": "PhoneNumber",
 "Contact": "71238889900",
 "Confirmed": true,
 "Primary": true,
 "Notification": true,
 "Usages": []
}
```

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404	user_not_found	Пользователь не найден.
400	invalid_phone	Переданное значение не является номером телефона.

1.3. Подтверждение существующего номера телефона

Если пользователь уже имеет требуемый номер мобильного телефона, но этот номер не подтвержден (в полученной информации о номере телефона параметр Confirmed имеет значение false), то этот номер мобильного телефона необходимо подтвердить. В зависимости от настройки Центра Идентификации для подтверждения номера телефона может понадобиться одноразовый пароль, отправленный на подтверждаемый номер телефона.

Запрос на подтверждение без одноразового пароля

```
POST https://host/STS/ums/user/8f4406bf-bd90-4baa-9787-866656907ada/phones/71238889900/confirm HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 2

{}
```

Пример ответа

Метод возвращает информацию о только что подтвержденном номере мобильного телефона.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 110

{
 "Type": "PhoneNumber",
 "Contact": "71238889900",
 "Confirmed": true,
 "Primary": true,
 "Notification": true,
 "Usages": []
}
```

Типовые ошибки



HTTP-код	ОШИБКА	ОПИСАНИЕ
404	user_not_found	Пользователь не найден.
400	invalid_phone	Переданное значение не является номером телефона.
400	wrong_operation	Неправильный вызов метода, например, номер телефона не добавлен пользователю или добавлен, но уже подтвержден.
400	contact_confirmation_required	Требуется подтверждение с использованием одноразового пароля из SMS.

Запрос отправки SMS с одноразовым паролем

Примечание

Если требованиями настроек ЦИ установлено подтверждение номера телефона с использованием одноразового пароля, то сначала необходимо отправить запрос на отпратку SMS с одноразовым паролем для подтверждения, а затем передать этот одноразовый пароль в запросе на подтверждение телефона.

После вызова данного метода на номер телефона пользователя будет отправлено SMS-сообщение с одноразовым паролем.

```
POST https://host/STS/ums/user/8f4406bf-bd90-4baa-9787-866656907ada/phones/71238889900/requireconfirm
HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 2

{}
```

Пример ответа

Метод не имеет возвращаемого значения.

```
HTTP/1.1 200 OK
```

Типовые ошибки

HTTP-код	ОШИБКА	ОПИСАНИЕ
404	user_not_found	Пользователь не найден.
400	invalid_phone	Переданное значение не является номером телефона.
400	wrong_operation	Неправильный вызов метода, например, номер телефона не добавлен пользователю или добавлен, но уже подтвержден.

Далее следует передать этот одноразовый пароль сервису, для завершения операции подтверждения номера телефона.

Подтверждение номера телефона пользователя с помощью одноразового пароля

```
POST https://host/STS/ums/user/8f4406bf-bd90-4baa-9787-866656907ada/phones/71238889900/submitconfirm
HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 7

"05643"
```

Пример ответа

Метод возвращает информацию о только что подтвержденном номере мобильного телефона.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 110

{
 "Type": "PhoneNumber",
 "Contact": "71238889900",
 "Confirmed": true,
 "Primary": true,
 "Notification": true,
 "Usages": []
}
```

Типовые ошибки

HTTP-код	ошибка	описание
404	user_not_found	Пользователь не найден.
400	invalid_phone	Переданное значение не является номером телефона.
400	wrong_operation	Неправильный вызов метода, например, когда номер телефона не добавлен пользователю или добавлен, но уже подтвержден.

1.4. Назначение номера телефона в качестве адресата получения одноразовых паролей

Сервису необходимо явно указать, какой именно номер телефона будет использоваться для вторичной аутентификации с помощью одноразовых паролей.

Запрос на назначение номера телефона в качестве адресата получения одноразовых паролей

```
POST https://host/STS/ums/user/8f4406bf-bd90-4baa-9787-866656907ada/phones/71238889900/secondaryauth HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 2

{}
```

Пример ответа

Метод не имеет возвращаемого значения.

```
HTTP/1.1 200 OK
```

Типовые ошибки

HTTP-код	ошибка	описание
404	user_not_found	Пользователь не найден.
400	invalid_phone	Переданное значение не является номером телефона.
400	wrong_operation	Неправильный вызов метода, например, когда номер телефона не добавлен пользователю.

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	contact_confirmation_required	Номер телефона пользователя не подтвержден.

2. Назначение метода аутентификации

POST Назначение метода аутентификации

```
POST https://host/STS/ums/user/0a0eaf08-665e-452c-a5b0-e342b3c43a3c/authmethod/otpviasms?level=1 HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Content-Length: 2

{}
```

**Внимание!**

Значение параметра `level` должно быть равно 1

Пример ответа

Метод не имеет возвращаемого значения.

```
HTTP/1.1 200 OK
```

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404	user_not_found	Пользователь не найден.
400	invalid_authentication_scheme	Указан неверный уровень метода аутентификации.
400	wrong_operation	Данный метод аутентификации уже назначен пользователю.
400	authn_method_not_confirmed	Попытка назначить метод аутентификации без указания подтвержденного номера мобильного телефона.

3. Назначение операций, требующих подтверждения по СМС

После сохранения номера мобильного телефона пользователя и подключения метода вторичной аутентификации по СМС необходимо задать список операций, требующих подтверждения.

Про типы операций, для которых можно настроить подтверждение по СМС, и их коды можно прочитать [на странице Типы Операций](#).

В запросе необходимо перечислить [коды операций](#), которые будет подтверждать пользователь.

POST назначение политик подтверждения операций пользователю

POST https://host/STS/ums/user/0a0eaf08-665e-452c-a5b0-e342b3c43a3c/operationpolicy HTTP/1.1  
Accept: application/json  
Content-Type: application/json; charset=utf-8  
Content-Length: 5

[2, 16, 1024]

#### Пример ответа

Метод не имеет возвращаемого значения.

HTTP/1.1 200 OK

#### Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	wrong_operation	Оператору запрещено изменять список операций, требующих подтверждения.
404	user_not_found	Пользователь не найден.

### 4. Отключение метода аутентификации и удаление номера мобильного телефона пользователя

Если у пользователя заданы политики подтверждения операций с помощью данного метода, то сначала нужно назначить другой метод для подтверждения операций или отключить подтверждение операций совсем. После этого можно отключить метод аутентификации, а затем удалить номер мобильного телефона пользователя.

#### DELETE Удаление метода аутентификации по одноразовым паролям, отправленным в СМС

DELETE https://host/STS/ums/user/0a0eaf08-665e-452c-a5b0-e342b3c43a3c/authmethod/otpviasms HTTP/1.1

#### Пример ответа

Метод не имеет возвращаемого значения.

HTTP/1.1 200 OK

Если номер мобильного телефона не задан или не подтвержден, нужно добавить его. Поскольку запрос выполняется от Оператора DSS, указанный в запросе номер считается подтвержденным.

#### DELETE Удаление номера мобильного телефона пользователя

DELETE https://host/STS/ums/user/0a0eaf08-665e-452c-a5b0-e342b3c43a3c/phones/71238889900 HTTP/1.1

#### Пример ответа

Метод не имеет возвращаемого значения.

HTTP/1.1 200 OK

#### Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_authentication_scheme	Указан неверный уровень метода аутентификации.

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404	user_not_found	Пользователь не найден.
400	authn_method_not_confirmed	Попытка назначить метод аутентификации без указания номера мобильного телефона.

## 5. Обратная совместимость с версией 2.0.3

В целях обратной совместимости с версией DSS 2.0.3 поддерживаются методы комплексного задания номера телефона и получение информации о заданном номере телефона пользователя.

Данный метод позволяет получить информацию о заданном номере мобильного телефона пользователя.

### Получение информации о номере телефона пользователя

```
GET https://host/STS/ums/user/0a0eaf08-665e-452c-a5b0-e342b3c43a3c/phonenumbers HTTP/1.1
```

#### Пример ответа

Метод возвращает номер мобильного телефона пользователя и значение, показывающее является ли номер телефона подтвержденным.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 47

{"PhoneNumber": "79150000001", "Confirmed": true}
```

### Изменение номера телефона пользователю

Данный метод объединяет функции добавления номера телефона (если такой номер не был добавлен), назначения его в качестве логина и в качестве адресата для отправки оповещений и одноразовых паролей для вторичной аутентификации.

#### Примечание

Номер мобильного телефона можно задавать в любом формате без пробелов.

```
POST https://host/STS/ums/user/0a0eaf08-665e-452c-a5b0-e342b3c43a3c/phonenumbers HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 14

"+79150455647"
```

#### Пример ответа

Метод не имеет возвращаемого значения.

```
HTTP/1.1 200 OK
```

#### Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_authentication_scheme	Указан неверный уровень метода аутентификации.

<b>HTTP-КОД</b>	<b>ОШИБКА</b>	<b>ОПИСАНИЕ</b>
404	user_not_found	Пользователь не найден.
400	authn_method_not_confirmed	Попытка назначить метод аутентификации без указания номера мобильного телефона.

## Поиск пользователя

Сервис Управления пользователями предоставляет несколько возможностей поиска пользователя:

- По логину, номеру телефона или email
- По идентификатору DssUserId
- Расширенный поиск

По логину, номеру телефона или email

## Пример запроса

Тип ключа поиска может принимать значения (значение параметра `type`):

- Login
- PhoneNumber
- Email

```
GET https://host/STS/ums/user?type=Login&value=DssTest-dc3bf3f5 HTTP/1.1
Accept: application/json
Host: host
```

## Пример ответа

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Content-Length: 334

{"UserId":"d1831dea-985f-4df1-a54b-2497eeace2f2","Login":"DssTest-dc3bf3f5","PhoneNumber":null,"Email":null,"PhoneConfirmed":false,"EmailConfirmed":false,"DisplayName":null,"DistinguishName":"","AccountLocked":false,"Group":"Default","CreationDate":"2018-08-24T14:36:33.02","LockoutDate":null,"LastLoginDate":"2018-08-24T14:36:33.02"}
```

По идентификатору DssUserId

## Пример запроса

```
GET https://host/STS/ums/user/d1831dea-985f-4df1-a54b-2497eeace2f2 HTTP/1.1
Accept: application/json
Host: host
```

## Пример ответа

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Content-Length: 334

{"UserId": "d1831dea-985f-4df1-a54b-2497eeace2f2", "Login": "DssTest-
dc3bf3f5", "PhoneNumber": null, "Email": null, "PhoneConfirmed": false, "EmailConfirmed": false, "DisplayName": null, "D
istinguishName": "", "AccountLocked": false, "Group": "Default", "CreationDate": "2018-08-
24T14:36:33.02", "LockoutDate": null, "LastLoginDate": "2018-08-24T14:36:33.02"}
```

Расширенный поиск

Расширенный поиск позволяет применять различные фильтры для поиска пользователей. Результатом выполнения метода может быть группа пользователей, отвечающая параметрам фильтра.

Поиск пользователей можно выполнить по одному или нескольким параметрам:

ПАРАМЕТР	КОД	ОПИСАНИЕ
Login	0	Логин пользователя
PhoneNumber	1	Номер телефона
Email	2	Адрес электронной почты
CreateDate	3	Дата создания учётной записи
GroupId	4	Идентификаторы группы пользователя

Код параметра указывается в поле

Операции сравнения могут быть следующих типов:

ТИП	КОД	ОПИСАНИЕ
Equal	0	Строгое равенство
NotEqual	1	Не равно
Like	2	Содержит
Greater	3	Больше
Less	4	Меньше

Код операции указывается в поле

Тип сравнения Like определяет, совпадает ли указанная символьная строка с заданным шаблоном. Шаблон может включать обычные символы и символы-шаблоны. Во время сравнения с шаблоном необходимо, чтобы его обычные символы в точности совпадали с символами, указанными в строке. Символы-шаблоны могут совпадать с произвольными элементами символьной строки.

Поддерживаются следующие символы шаблоны:

СИМВОЛ-ШАБЛОН	ОПИСАНИЕ	ПРИМЕР
%	Любая строка, содержащая ноль или более символов.	%вано%
(подчеркивание)	Любой одиночный символ.	_етров
[ ]	Любой одиночный символ, содержащийся в диапазоне ([a-f]) или наборе ([abcdef]).	[Л-С]омов
[^]	Любой одиночный символ, не содержащийся в диапазоне ([^a-f]) или наборе ([^abcdef]).	'ив[^a]%

Параметры `StartPosition` и `EndPosition` определяют начальную и конечную позицию из итоговой выборки. Данные параметры могут быть использованы для страничной выборки пользователей

При поиске пользователей по времени создания значение фильтра должно иметь следующий формат: **yyyy-MM-ddThh:mm:ss**

Общее количество элементов подпадающих под критерии фильтра возвращается в параметре `TotalCount`. Количество элементов отданных методом возвращается в параметре `AffectedCount`: `AffectedCount <= EndPosition - StartPosition`

Примеры запросов

Получить пользователя с заданным логином:

```
POST https://host/STS/ums/users HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 101
Expect: 100-continue

{"StartPosition":1,"EndPosition":1,"Filters":[{"Column":0,"Operation":0,"Value":"DssTest-dc3bf3f5"}]}
```

Проверка были ли создан пользователь с заданным логином в указанном промежуток времени:

```
POST https://host/STS/ums/users HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 172
Expect: 100-continue

{"StartPosition":1,"EndPosition":1,"Filters":[{"Column":0,"Operation":0,"Value":"DssTest-2fa204c5"}, {"Column":3,"Operation":3,"Value":"2018-08-24T15:12:12.4683672+03:00"}]}
```

Получить пользователей созданных в указанный промежуток времени:

```
POST https://host/STS/ums/users HTTP/1.1
Accept: application/json
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 161
Expect: 100-continue

{"StartPosition":1,"EndPosition":10,"Filters":[{"Column":3,"Operation":4,"Value":"2018-08-25T04:24:50"}, {"Column":3,"Operation":3,"Value":"2018-08-23T04:24:50"}]}
```

Пример ответа



```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Content-Length: 386

{"UserInfos":[{"UserId":"d1831dea-985f-4df1-a54b-2497eeace2f2","Login":"DssTest-
dc3bf3f5","PhoneNumber":null,"Email":null,"PhoneConfirmed":false,"EmailConfirmed":false,"DisplayName":null,"D
istinguishName":"","AccountLocked":false,"Group":"Default","CreationDate":"2018-08-
24T14:36:33.02","LockoutDate":null,"LastLoginDate":"2018-08-
24T14:36:33.02"}],"TotalCount":2047,"AffectedCount":1}
```

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
404	user_not_found	Пользователь не найден.
500	An error has occurred	Неверно указано значение или тип фильтра.

# Подтверждение операций

В данном разделе описывается последовательность действий для подтверждения операции по одноразовым паролям, отправляемых по СМС. Все запросы выполняются от Пользователя DSS.

## Внимание!

Перед началом работы с Центром Идентификации, должны быть выполнены [предварительные настройки сервиса Администратором DSS](#).

Предварительно должно быть:

1. Пользователю создана учетная запись в DSS.
2. Пользователю назначена [вторичная аутентификация по одноразовым паролям](#), отправляемых по СМС, и заданы операции, требующие подтверждения.
3. Пользователю выпущен сертификат подписи.
4. Зарегистрированный на сервере OAuth 2.0 клиент.

## Аутентификация пользователя на Центре Идентификации

В примере рассматривается авторизация с использованием учётных данных пользователя (логин/пароль). Подробная информация по протоколу аутентификации: [The OAuth 2.0 Authorization Framework](#)

Параметры запроса:

- `grant_type` - тип разрешения, в данном сценарии равен **password**.
- `password` – пароль пользователя.
- `resource` – идентификатор Сервиса Подписи.

В заголовке Authorization HTTP-запроса клиент должен передать идентификатор OAuth-клиента и секрет (если используется): Authorization: Basic Base64(<client\_id>:<secret>)

## Примечание

В примере значение параметр **password** оставлено пустым, так как пользователю в качестве первичной аутентификации назначен метод "Только Идентификация"

## Пример запроса

```
POST https://host/STS/oauth/token HTTP/1.1
Authorization: Basic dGVzdENSaWVudDo=
Content-Type: application/x-www-form-urlencoded
Host: host
Content-Length: 101
Expect: 100-continue
Connection: Keep-Alive

grant_type=password&username=mydss&password=&resource=urn%3Acryptopro%3Adss%3Asignserver%3Asignserver
```

В случае успешной аутентификации ответ будет содержать:

- `access_token` - AccessToken, выпущенный Центром Идентификации DSS
- `token_type` - Тип токена
- `expires_in` - Время жизни токена в секундах

Значение параметра `access_token` необходимо будет использовать при обращениях к Сервису Подписи и Сервису Подтверждения Операций.

## Пример ответа

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 2017
Content-Type: application/json; charset=utf-8
Expires: -1

{
 "access_token": "eyJ0eXAiOiJKV... 5Wti-H8CeXycwB6A",
 "expires_in": 300,
 "token_type": "Bearer"
}
```

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_client	OAuth-клиент не зарегистрирован или неверно указан clientID
400	unauthorized_client	OAuth-клиент использует незарегистрированный сценарий аутентификации (Flow)
400	invalid_request	Неверно сформирован параметр resource
500	An error has occurred	1. Проверяющая сторона с идентификатором resource не зарегистрирована.

1. Создание транзакции подписи на Сервисе Подписи

Для подтверждения любых операций на Сервисе Подписи используется метод [/transactions](#).

В заголовке Authorization HTTP-запроса клиент должен указать AccessToken, полученный при аутентификации:  
*Authorization: Bearer <access\_token>*

В запросе необходимо указать:

- `OperationCode` – Тип создаваемой транзакции.
- `Parameters` – Параметры транзакции.
- `Document` – Подписываемый документ.

Идентификатор сертификата подписи `CertificateID` можно получить, запросив список сертификатов пользователя у конечной точки `\certificates`.

В данном примере показано, как пользователь инициирует подписание документа, после того как аутентифицировался в Сервисе Подписи. Параметры создания транзакций для других операций приведены [здесь](#)

1.1 POST Запрос на создание транзакции подписи на Сервисе Подписи

В примере создаётся прикреплённая CAdES-BES подпись.

```
POST https://host/SignServer/rest/api/transactions HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh... 8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 355049
Expect: 100-continue
```

```
{
 "OperationCode":2,
 "Parameters":
 [
 {"Name":"SignatureType","Value":"CMS"},
 {"Name":"CertificateID","Value":"13"},
 {"Name":"DocumentInfo","Value":"testPdf.pdf"},
 {"Name":"DocumentType","Value":"pdf"},
 {"Name":"IsDetached","Value":"false"},
 {"Name":"CADESType","Value":"BES"}
],
 "Document":"JVBERi0xLjUNCiW1tbW14Kfu"
}
```

### Пример ответа

Сервис Подписи вернёт идентификатор созданной транзакции.

```
HTTP/1.1 200 OK
Content-Length: 38
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5

"d5ebd393-e093-4aa8-bdcf-f5e497dc6b4d"
```

Далее пользователю требуется, используя полученный идентификатор, подтвердить транзакцию на Сервисе Подтверждения Операций.

### Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_certificate	Неверный идентификатор сертификата
400	invalid_request	Неверно указаны параметры подписи

### 2. Подтверждение транзакции подписи на Сервисе Подтверждения Операций

Для подтверждения транзакции, созданной на Сервисе Подписи, пользователь должен отправить запрос на Сервис Подтверждения Операций.

В заголовке Authorization HTTP-запроса клиент должен указать AccessToken, полученный при аутентификации:  
*Authorization: Bearer <access\_token>*

В запросе необходимо указать:

- TransactionTokenId – Идентификатор транзакции, созданной на Сервисе Подписи (пункт 1.1).
- Resource – Идентификатор Сервиса Подписи в формате urn:cryptopro:dss:signserver:<signserver app name> (по умолчанию имеет значение **urn:cryptopro:dss:signserver:signserver**).
- ClientId - идентификатор OAuth клиента.
- ClientSecret - пароль OAuth клиента (для неконфиденциальных клиентов данный параметр не указывается).

#### 2.1 POST-запрос на подтверждение созданной транзакции

```

POST https://host/STS/confirmation HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIHM ... 5aPB98A3NAVduJbtz5Wti-H8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 246
Expect: 100-continue

{
 "Resource": "urn:cryptopro:dss:signserver:signserver",
 "ClientId": "oauth-client-id",
 "ClientSecret": "oauth-client-secret",
 "TransactionTokenId": "d5ebd393-e093-4aa8-bdcf-f5e497dc6b4d",
}

```

В зависимости от количества подключенных способов подтверждения операций возможно два варианта ответа на запрос:

- Если у Пользователя подключен один метод подтверждения операций, то при успешном выполнении запроса он получит одноразовый пароль на указанный номер мобильного телефона (*Пример ответа 1*).
- Если у Пользователя подключено несколько методов подтверждения операций, то при успешном выполнении запроса он получит доступные методы подтверждения операции (*Пример ответа 2*). В этом случае необходимо выбрать метод подтверждения операции (пункт 2.1.1).

#### *Пример ответа 1*

При успешном выполнении запроса Пользователь должен получить на свой номер мобильного телефона одноразовый пароль. Необходимо запомнить идентификатор транзакции *RefID* для ее подтверждения в пункте 2.2.

```

HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 6736
Content-Type: application/json; charset=utf-8
Expires: -1

{
 "Challenge": {
 "Title": {
 "Value": "Код подтверждения отправлен на ваш номер мобильного телефона"
 },
 "TextChallenge": [
 {
 "AuthnMethod": "http://dss.cryptopro.ru/identity/authenticationmethod/otpviasms",
 "RefID": "5460867e-be6a-4940-bebc-9aeb516fa448",
 "Label": "02.11.2018 9:47:05. Подпись документа. testPdf.pdf. Идентификатор операции gcoaryoy. Пользователь Test2. Сертификат: test2.",
 "MaxLenSpecified": false,
 "HideTextSpecified": false,
 "ExpiresIn": 86400,
 "ExpiresInSpecified": true
 }
],
 "ContextData": {
 "RefID": "5460867e-be6a-4940-bebc-9aeb516fa448"
 }
 },
 "IsFinal": false,
 "IsError": false
}

```

#### *Пример ответа 2*

Необходимо запомнить идентификатор транзакции *RefID* для выбора метода ее подтверждения в пункте 2.1.1.

```

HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 6736
Content-Type: application/json; charset=utf-8
Expires: -1

{
 "Challenge": {
 "Title": {
 "Value": "Для подтверждения операции необходимо выбрать способ аутентификации"
 },
 "ChoiceChallenge": [
 {
 "Choice": [
 {
 "RefID": "http://dss.cryptopro.ru/identity/authenticationmethod/otpviasms",
 "Label": "Аутентификация с помощью СМС"
 },
 {
 "RefID": "http://dss.cryptopro.ru/identity/authenticationmethod/mobile",
 "Label": "Аутентификация с помощью мобильного приложения"
 }
],
 "RefID": "7406c54e-0ba5-4340-bbaa-3c2feebdc852",
 "Label": "Для подтверждения операции необходимо выбрать способ аутентификации",
 "ExactlyOne": true,
 "ExactlyOneSpecified": true,
 "ExpiresIn": 86400,
 "ExpiresInSpecified": true
 }
],
 "ContextData": {
 "RefID": "7406c54e-0ba5-4340-bbaa-3c2feebdc852"
 }
 },
 "IsFinal": false,
 "IsError": false
}

```

Ответ Сервиса Подтверждения Операций содержит:

ПОЛЕ	ОПИСАНИЕ
Challenge	Запрос на выполнение аутентификационного испытания
AccessToken	Маркер доступа. Заполняется при <code>IsFinal</code> - true
ExpiresIn	Время жизни AccessToken в секундах. Заполняется при <code>IsFinal</code> - true
IsFinal	Является ли данный ответ последним в процессе подтверждения.
IsError	Содержит ли данный ответ ошибку обработки запроса. Заполняется при <code>IsFinal</code> - false
Error	Ошибка обработки запроса. Заполняется при <code>IsFinal</code> - false

ПОЛЕ	ОПИСАНИЕ
ErrorDescription	Подробное описание ошибки обработки запроса

Поле `Challenge` содержит:

ПОЛЕ	ОПИСАНИЕ
Title	Текст, который вызывающая система может отобразить пользователю в своём интерфейсе
TextChallenge	Дополнительные данные для подтверждения операции
ChoiceChallenge	Возвращается только в случае, если у Пользователя включено несколько способов подтверждения транзакций. Содержит информацию об этих способах подтверждения транзакций.

В поле `TextChallenge` содержится:

ПОЛЕ	ОПИСАНИЕ
RefID	Идентификатор транзакции, созданной на Сервисе Подтверждения Операций
ExpiresIn	Срок действия транзакции, созданной на Сервисе Подтверждения Операций
AuthnMethod	Идентификатор метода, используемый для подтверждения транзакции

**Примечание**

`RefId` - Идентификатор транзакции, созданной на Сервисе Подтверждения Операций. Его необходимо будет использовать при последующих обращениях на конечную точку `/confirmation` (пункты 2.1.1 и 2.2).

В поле `ChoiceChallenge` содержится:

ПОЛЕ	ОПИСАНИЕ
RefID	Идентификатор транзакции, созданной на Сервисе Подтверждения Операций
Choice	Информация о доступных способах подтверждения транзакций

В поле `Choice` содержится:

ПОЛЕ	ОПИСАНИЕ
RefID	Идентификатор метода подтверждения транзакции
Label	Название метода подтверждения транзакции

**Примечание**

При обработке ответа Сервиса Подтверждения Операций вызывающее приложение должно смотреть на значение двух флагов: `IsFinal` и `IsError`.

Если получен ответ с `IsError` - true, то дальнейшее подтверждение транзакции невозможно.

Если получен ответ с `IsFinal` - false, то подтверждение транзакции ещё не завершено.

### 2.1.1 POST Запрос на выбор метода подтверждения транзакции (только если подключено несколько методов подтверждения)

В заголовке Authorization HTTP-запроса клиент должен указать AccessToken, полученный при аутентификации:

*Authorization: Bearer <access\_token>*

В запросе необходимо указать:

- `ChoiceSelected.RefID` – Идентификатор выбранного метода подтверждения транзакции
- `RefId` – Идентификатор транзакции, созданной на Сервисе Подтверждения Операций (пункт 2.1).
- `Resource` – Идентификатор Сервиса Подписи в формате `urn:cryptopro:dss:signserver:<signserver app name>` (по умолчанию имеет значение **`urn:cryptopro:dss:signserver:signserver`**).

```
POST https://host/STS/confirmation HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIJHM ... 5aPB98A3NAVduJbtz5Wti-H8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 246
Expect: 100-continue
{
 "Resource" : "urn:cryptopro:dss:signserver:signserver",
 "ClientId": "oauth-client-id",
 "ClientSecret": "oauth-client-secret",
 "ChallengeResponse" : {
 "ChoiceChallengeResponse" : [{
 "RefId" : "49744464-5cd7-419d-891e-2495b8f49539",
 "ChoiceSelected": [{
 "RefID": "http://dss.cryptopro.ru/identity/authenticationmethod/otpviasms",
 }]
 }]
 }
}
```

#### Пример ответа

При успешном выполнении запроса Пользователь должен получить на свой номер мобильного телефона одноразовый пароль. Необходимо запомнить *новый* идентификатор транзакции `RefID` для ее подтверждения в пункте 2.2.



```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 6736
Content-Type: application/json; charset=utf-8
Expires: -1

{
 "Challenge": {
 "Title": {
 "Value": "Код подтверждения отправлен на ваш номер мобильного телефона"
 },
 "TextChallenge": [
 {
 "AuthnMethod": "http://dss.cryptopro.ru/identity/authenticationmethod/otpviasms",
 "RefID": "5460867e-be6a-4940-bebc-9aeb516fa448",
 "Label": "02.11.2018 9:47:05. Подпись документа. testPdf.pdf. Идентификатор операции gsoaryoy. Пользователь Test2. Сертификат: test2.",
 "MaxLenSpecified": false,
 "HideTextSpecified": false,
 "ExpiresIn": 86400,
 "ExpiresInSpecified": true
 }
],
 "ContextData": {
 "RefID": "5460867e-be6a-4940-bebc-9aeb516fa448"
 }
 },
 "IsFinal": false,
 "IsError": false
}
```

## Примечание

**RefId** - Идентификатор транзакции, созданной на Сервисе Подтверждения Операций. Его необходимо будет использовать при следующем обращении на конечную точку /confirmation (пункт 2.2).

## 2.2 POST Запрос на получение результата подтверждения транзакции

Для отправки кода подтверждения операции, созданной на Сервисе Подписи, пользователь должен отправить запрос на Сервис Подтверждения Операций.

В заголовке Authorization HTTP-запроса клиент должен указать AccessToken, полученный при аутентификации:

*Authorization: Bearer <access\_token>*

В запросе необходимо указать:

- **RefId** – Идентификатор транзакции, созданной на Сервисе Подтверждения Операций (пункт 2.1 или 2.1.1).
- **Value** – Одноразовый пароль, который Пользователь получил на свой номер мобильного телефона.

```
POST https://host/STS/confirmation HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIHM ... 5aPB98A3NAVduJbtz5Wti-H8CeXycwB6A
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 229
Expect: 100-continue
```

```
{
 "Resource" : "urn:cryptopro:dss:signserver:signserver",
 "ClientId": "oauth-client-id",
 "ClientSecret": "oauth-client-secret",
 "ChallengeResponse":
 {
 "TextChallengeResponse":
 [
 {
 "RefId": "5460867e-be6a-4940-bebc-9aeb516fa448",
 "Value": "12..56"
 }
]
 }
}
```

#### Пример ответа

Если в ответе `IsFinal` - `true`, то Сервис вернул новый `AccessToken` для завершения операции на Сервисе Подписи.

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 2215
Content-Type: application/json; charset=utf-8
Expires: -1

{
 "AccessToken": "eyJ0eXAiOiJKV1QiLC ... 5b1T6H1ytuWztMPGFz-0w",
 "ExpiresIn": 600,
 "IsFinal": true,
 "IsError": false
}
```

#### Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_transaction	1. Срок действия транзакции истёк 2. Передан неверный идентификатор транзакции ( <code>RefId</code> )
400	authentication_failed	Передан неверный код подтверждения

При успешном выполнении запросов транзакция подписи является подтвержденной на Сервисе Подтверждения Операций. Пользователю возвращается новый `AccessToken`, который понадобится при обращении к Сервису Подписи за результатом операции.

#### 3. Получение подписанного документа на Сервисе Подписи

Для получения подписанного документа необходимо отправить запрос Сервису Подписи на конечную точку `/documents`.

#### Примечание

В заголовке `Authorization` HTTP-запроса клиент должен указать `AccessToken` полученный от **Сервиса Подтверждения Операций** в пункте 2.2: `Authorization: Bearer <access_token>`

3.1 POST Запрос на получение подписанного документа

```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLC... 5b1T6H1ytuWztMPGfz-Ow
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 2
Expect: 100-continue

{
}
```

**Примечание**

Если закрытый ключ сертификата защищён на ПИН-коде, то ПИН-код должен быть указан при обращении на конечную точку /documents

Пример запроса с указанием ПИН-кода:

```
POST https://host/SignServer/rest/api/documents HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLC... 5b1T6H1ytuWztMPGfz-Ow
Content-Type: application/json; charset=utf-8
Host: host
Content-Length: 97
Expect: 100-continue

{
 "Signature":{"PinCode":"1234"}
}
```

Пример ответа

```
HTTP/1.1 200 OK
Content-Length: 356734
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/7.5

"MIMEFRcG ... gkRSA"
```

Типовые ошибки

HTTP-КОД	ОШИБКА	ОПИСАНИЕ
400	invalid_pin	Неверно указан ПИН-код на закрытый ключ

# Разное

В данном разделе собраны статьи на различные темы.

- [Сервис обнаружения;](#)
- [Подпись HTTP-сообщений.](#)

# Сервис обнаружения

В одном домене (под доменом понимается некоторая физическая среда, стоящая за одним доменным именем) может быть установлено несколько экземпляров DSS. В такой конфигурации пользователю может быть предоставлен выбор, к какому конкретному экземпляру подключиться для выполнения операции. Для принятия решения пользователь должен ознакомиться с описанием всех доступных экземпляров. Клиентская система может получить данное описание с помощью **Сервиса обнаружения (Discovery Service)**.

## Протокол WebFinger

Сервиса обнаружения использует протокол **WebFinger (RFC7033)**.

Для получения информации о всех зарегистрированных экземплярах служб DSS клиент отправляет следующий HTTP запрос:

```
GET https://dss.cryptopro.ru/.well-known/webfinger
?resource=https%3A%2F%2Fdss.cryptopro.ru
```

В ответ сервер возвращает так называемый JSON описатель ресурса (Json Resource Descriptor, JRD). В данном случае ресурсом становится сам сервер (именно поэтому в качестве параметра **resource** передавался URL адрес самого сервера):

```
[
{
 "subject" : "https://dss.cryptopro.ru",
 "links" : [
 {
 "title" :
 {
 "ru-RU" : "Тестовый Сервер Электронной Подписи КристоПро",
 "en-US" : "The Crypto-Pro Test Digital Signature Server"
 },
 "rel" : "http://dss.cryptopro.ru/signserver",
 "href" : "https://dss.cryptopro.ru/SignServer/",
 "properties" :
 {
 "http://dss.cryptopro.ru/sts" : "https://dss.cryptopro.ru/STS/",
 "http://dss.cryptopro.ru/frontend" : "https://dss.cryptopro.ru/Frontend/"
 }
 },
 {
 "title" :
 {
 "ru-RU" : "Тестовый Сервер Электронной Подписи Lite КристоПро",
 "en-US" : "The Crypto-Pro Test Digital Signature Server Lite"
 },
 "rel" : "http://dss.cryptopro.ru/signserver",
 "href" : "https://dss.cryptopro.ru/SignServerLite/",
 "properties" :
 {
 "http://dss.cryptopro.ru/sts" : "https://dss.cryptopro.ru/STS/",
 "http://dss.cryptopro.ru/frontend" : "https://dss.cryptopro.ru/Lite/"
 "http://dss.cryptopro.ru/signserver-type" : "client"
 }
 },
 {
 "title" :
 {
 "ru-RU" : "Тестовый Центр Идентификации КристоПро",
 "en-US" : "The Crypto-Pro TestIdentity Provider"
 },
 "rel" : "http://dss.cryptopro.ru/sts",
 "href" : "https://dss.cryptopro.ru/STS/"
 }
]
}
```

```

 "href" : "https://dss.cryptopro.ru/STS/"
 },
 {
 "title" :
 {
 "ru-RU" : "Веб-интерфейс Сервиса Подписи КриптоПро",
 "en-US" : "The Crypto-Pro Sign Service Web Interface"
 },
 "rel" : "http://dss.cryptopro.ru/frontend",
 "href" : "https://dss.cryptopro.ru/Frontend/",
 "properties" :
 {
 "http://dss.cryptopro.ru/sts" : "https://dss.cryptopro.ru/STS/",
 "http://dss.cryptopro.ru/signserver" : "https://dss.cryptopro.ru/SignServer/"
 }
 },
 {
 "title" :
 {
 "ru-RU" : "Веб-интерфейс Сервиса Подписи КриптоПро Lite",
 "en-US" : "The Crypto-Pro Sign Service Lite Web Interface"
 },
 "rel" : "http://dss.cryptopro.ru/frontend",
 "href" : "https://dss.cryptopro.ru/Frontend/",
 "properties" :
 {
 "http://dss.cryptopro.ru/sts" : "https://dss.cryptopro.ru/STS/",
 "http://dss.cryptopro.ru/signserver" : "https://dss.cryptopro.ru/SignServerLite/"
 }
 }
}
]
}
]

```

## JRD документ

JRD описывает некоторый ресурс (в данном случае сам сервер). Его основными элементами является свойство **subject** (являющееся идентификатором описываемой сущности; его значение может и не совпадать со значением параметра resource, переданного в запросе) и набора **связей (link)**. В данном случае под связью понимаются экземпляры служб DSS, установленные на данном сервере.

Каждая связь характеризуется следующими параметрами:

- **rel** - тип отношения. В случае описания сервера DSS под типом понимаются различные типы сервисов DSS (Центр Идентификации, Сервис Подписи, Веб-интерфейс и т.д.)
- **href** - ссылка на экземпляр службы.
- **title** - отображаемое название экземпляра службы, может отличаться для разных языков.
- **properties** - словарь (ключ/значение) дополнительных свойств связи.

В JRD примере, представленном выше, содержится пять связей, соответствующих пяти службам DSS, развёрнутым на этом сервере (естественно, что служб может быть развёрнуто больше, но администратор сервере решил описать только указанные в JRD).

### Типы связей

Сервис обнаружения КриптоПро DSS оперирует следующими типами связей:

ИДЕНТИФИКАТОР ТИПА	ОПИСАНИЕ	ПРИМЕЧАНИЕ
http://dss.cryptopro.ru/signserver	Сервис подписи	<b>href</b> задаёт базовый адрес веб-приложения

ИДЕНТИФИКАТОР ТИПА	ОПИСАНИЕ	ПРИМЕЧАНИЕ
<code>http://dss.cryptopro.ru/sts</code>	Центр Идентификации	<b>href</b> задаёт базовый адрес веб-приложения
<code>http://dss.cryptopro.ru/frontend</code>	Веб-интерфейс пользователя	<b>href</b> задаёт адрес веб-приложения

## Свойства связей

Свойства позволяют описать дополнительную информацию о связанной службе. Например, свойство `http://dss.cryptopro.ru/frontend` описывает адрес веб-интерфейса пользователя, соответствующего данной службе Сервиса Подписи.

Сервис обнаружения КриптоПро DSS оперирует следующими типами свойств:

ИДЕНТИФИКАТОР ТИПА	ОПИСАНИЕ
<code>http://dss.cryptopro.ru/signserver</code>	URL-адрес сервиса подписи
<code>http://dss.cryptopro.ru/sts</code>	URL адрес Центра Идентификации
<code>http://dss.cryptopro.ru/frontend</code>	URL адрес веб-интерфейса
<code>http://dss.cryptopro.ru/signserver-type</code>	Режим работы сервиса подписи

## Свойство `signserver-type`

Данное свойство определяет режим работы экземпляра службы Сервиса Подписи КриптоПро DSS. Возможные значения

- `client` - Сервис Подписи работает в режиме КриптоПро DSS Lite, то есть с ключами, хранящимися на устройстве пользователя.
- `server` - Сервис Подписи работает с "облачными" ключами. Если свойство `signserver-type` не указано, используется это значение по умолчанию.
- `server-client` - Сервис Подписи работает в смешанном режиме.

## Обработка JRD документа

Рассмотрим один из вариантов чтения JRD документа.

1. Клиентское приложение (далее просто **клиент**) запрашивает у пользователя адрес сервера, к которому он хочет подключиться.
2. Клиент отправляет запрос по протоколу WebFinger к указанному серверу, указывая в качестве параметра resource адрес сервера.
3. Получив в ответ JRD описание, клиент находит в нём связи с типом `rel=http://dss.cryptopro.ru/signserver`, тем самым перечисляя все установленные на этом сервере экземпляры службы Сервиса Подписи.
4. Клиент отображает пользователю найденные службы, показывая в списке их понятное описание (`title`).
5. После того, как пользователь выбрал экземпляр, клиент может получить URL-адрес Центра Идентификации из свойства `http://dss.cryptopro.ru/sts` и сформировать запрос на авторизацию.

## Получение конкретной связи

В запрос по протоколу WebFinger можно добавить параметр `rel` с указанием конкретного типа связи, который хочется получить. В этом случае сервер вернёт только запрошенные связи:

## Запрос

```
GET https://dss.cryptopro.ru/.well-known/webfinger
?resource=https%3A%2F%2Fdss.cryptopro.ru
&rel=http%3A%2F%2Fdss.cryptopro.ru%2Fsignserver
```

## Ответ

```
[
{
 "subject" : "https://dss.cryptopro.ru",
 "links" : [
 {
 "title" :
 {
 "ru-RU" : "Тестовый Сервер Электронной Подписи КристоПро",
 "en-US" : "The Crypto-Pro Test Digital Signature Server"
 },
 "rel" : "http://dss.cryptopro.ru/signserver",
 "href" : "https://dss.cryptopro.ru/SignServer/",
 "properties" :
 {
 "http://dss.cryptopro.ru/sts" : "https://dss.cryptopro.ru/STS/",
 "http://dss.cryptopro.ru/frontend" : "https://dss.cryptopro.ru/Frontend/"
 }
 },
 {
 "title" :
 {
 "ru-RU" : "Тестовый Сервер Электронной Подписи Lite КристоПро",
 "en-US" : "The Crypto-Pro Test Digital Signature Server Lite"
 },
 "rel" : "http://dss.cryptopro.ru/signserver",
 "href" : "https://dss.cryptopro.ru/SignServerLite/",
 "properties" :
 {
 "http://dss.cryptopro.ru/sts" : "https://dss.cryptopro.ru/STS/",
 "http://dss.cryptopro.ru/frontend" : "https://dss.cryptopro.ru/Lite/"
 "http://dss.cryptopro.ru/signserver-type" : "client"
 }
 }
]
}
]
```

## Установка и удаление сервиса обнаружения

Сервис обнаружения входит в состав дистрибутива DSS в виде самораспаковывающегося архива

`dss-webfinger-{version}.exe`.

После распаковки архива необходимо установить HTTP модуль для IIS. Это делается с помощью следующих PowerShell командлетов:

```
Импортируем PowerShell модуль.
Import-Module "<Путь установки>\PowerShell\CryptoPro.DSS.WebFinger\CryptoPro.DSS.WebFinger.psm1"
Add-WebFingerService -SiteName "Default Web Site"
```

Для удаление сервиса с сервера IIS необходимо выполнить следующие команды

```
Импортируем PowerShell модуль.
Import-Module "<Путь установки>\PowerShell\CryptoPro.DSS.WebFinger\CryptoPro.DSS.WebFinger.psm1"
Remove-WebFingerService -SiteName "Default Web Site"
```



Если в каком-либо из командлетов не указать значение для параметра `SiteName`, то будет использовано значение по умолчанию `Default Web Site`.

## Заполнение JRD документа

После разворачивания сервиса в папке, где расположен веб-сайт (для Default Web Site это `C:\inetpub\wwwroot\.well-known\webfinger`) появится файл `registry.json`. Содержимое файла заполнено демонстрационными данными (как в примере выше). Данный файл можно редактировать в любом текстовом редакторе. Сразу после редактирования новые данные будут возвращены сервисом.

# Подпись HTTP сообщений

В качестве дополнительной защиты от подмены сообщения в TLS-канале отправляемое сообщение может быть подписано.

## Формирование подписи

Клиент формирует подпись отправляемого сообщения следующим образом.

1. Создаётся случайный ключ `K`.
2. На основе отправляемого HTTP сообщения формируется строка `M`.
3. Вычисляется подпись `S` на ключе `K` от строки `M`: `BASE64(HMAC(K, UTF8.GetBytes(M)))`.
4. Формируется заголовок `CP-Signature`, содержащий `S` и `K`, и добавляется к HTTP запросу.

## Проверка подписи

Сервер проверяет подпись следующим образом.

1. Из заголовка `CP-Signature` извлекается ключ `K` и подпись `S`.
2. На основе полученного HTTP сообщения формируется строка `M`.
3. Вычисляется подпись `S'` на ключе `K` от строки `M`: `BASE64(HMAC(K, UTF8.GetBytes(M)))`.
4. Подпись `S'` сравнивается с `S`. Если значения совпадают, то подпись верна, иначе в сообщении что-то было изменено.

## Формирование строки `M`

Строка `M` формируется на основе заголовков HTTP сообщения и его содержимого. Целевой URL считается псевдозаголовком `(request-target)`.

Строка `M` представляет собой конкатенацию двух строк: `headers` и `body`.

### Формирование строки `headers`

Для каждого *подписываемого* заголовка (какие заголовки будут подписаны определяется клиентом) вычисляются строки `header-name: header-value`, где `header-name` - это название заголовка в нижнем регистре, а `header-value` значение заголовка без лидирующих и конечных пробелов.

Если заголовком является `(request-target)`, то его значение вычисляется как `method URL`, где `method` - название HTTP метода в нижнем регистре, а URL - целевой URL запроса (без изменений).

Все сформированные строки конкатенируются через символ новой строки `\n`. После последней строки символа `\n` нет.

### Пример

Для HTTP запроса

[illegible][illegible]

Строка body представляет собой содержимое HTTP запроса (без каких-либо изменений).

## Формирование заголовка CP-Signature

Заголовок CP-Signature имеет вид (для предыдущего примера): Здесь переносы строк добавлены только для наглядности.

```
CP-Signature: headers="(request-target) content-length content-type authorization",
key="QH5xtLRVDncY50KC82RprzX9AIhaDNioX4wzb9P1Gn8=",
signature="8z8qEydpfZdnzOwRn8eVvxSmIUwx3akScU6azm3SNUc="
```

В параметре key передаётся значение ключа K в BASE64, в параметре headers передаётся перечень всех подписываемых заголовков, в параметре signature - значение подписи S в BASE64.

# Преобразование утверждений

При аутентификации через сторонний ЦИ в КриптоПро DSS передается маркер доступа, содержащий некоторый набор утверждений о пользователе. Некоторые ЦИ позволяют настраивать содержимое маркера, но часто это бывает недоступно.

Для успешной аутентификации в КриптоПро DSS пользователь должен предоставить маркер как минимум с двумя утверждениями:

1. Name (uri).
2. Role

В первом утверждении содержится уникальный идентификатор пользователя в пределах стороннего ЦИ, во втором роль пользователя в DSS. Если во входящем маркере нет указанных утверждений, аутентификация завершится с ошибкой.

DSS позволяет для каждого стороннего ЦИ задавать правила преобразования входящих утверждений.

## Правила преобразования утверждений

Преобразования входящих утверждений описывается правилами. Каждое правило состоит из двух частей:

1. Селектора, определяющего какие входящие утверждения должны обрабатываться правилом.
2. Модификатора, определяющего какие действия нужно сделать с выбранными селектором утверждениями.

Общий вид правила:

```
id:[condition]=>action(body);
```

Процесс преобразования состоит из следующих шагов:

1. Копирование всех входящих утверждений в рабочий набор.
2. Применение селектора для выбора утверждений из рабочего набора.
3. Применение модификатора для всех утверждений, выбранных селектором.
4. Сохранение результата в рабочий набор и (опционально) в выходной набор.
5. Шаги 2-4 повторяются для каждого правила.
6. Результатом работы является выходной набор утверждений.

Рассмотрим более подробно каждый компонент правила.

### Селекторы

Селекторы выбирают утверждения из рабочего набора и передают их модификатору. Селектор имеет следующий вид:

```
id:[condition]
```

, где

1. `id` - идентификатор выбранного утверждения, с помощью этого идентификатора на выбранное утверждение можно ссылаться в модификаторе.
2. `condition` - условие выбора. Если в рабочем наборе есть утверждение, удовлетворяющее условию, оно будет передано модификатору и ему будет присвоен идентификатор `id`.

Условия бывают двух видов:

1. Пустое условие: `id:[ ]` - под это условие подходит любое утверждение.
2. Выборка по свойству утверждения: `id:[prop1 = value1, prop2 = value2]` - под это условие подходит утверждение, свойство `prop1` которого равно `value1`, свойство `prop2` равно `value2`.

Утверждение обладает следующими свойствами:

1. `type` - тип утверждения.
2. `value` - значение утверждения.

В условии могут быть указаны все свойства, либо только часть из них.

Существует особый вид селектора - пустой селектор. В результате его работы ничего не передаётся модификатору, который в этом случае вызывается безусловно и один раз. Данный селектор подходит для добавления в выходные утверждения с константным значением и типом. Правило с пустым селектором имеет следующий вид:

```
=>action(body)
```

Если селектор не нашёл подходящих под условие утверждений, модификатор не вызывается.

## Модификаторы

Модификаторы работают со списком утверждений, возвращённых селектором. Общий вид модификатора:

```
=>action(body)
```

, где

1. `action` - действие. Можете принимать значение `issue` или `add`. Действие `issue` добавляет модифицированное утверждение в рабочий и выходной набор утверждений, действие `add` - только в рабочий.
2. `body` - тело модификатора, определяющее, какую операцию нужно сделать с выбранным утверждением.

Модификатор может выполнять две операции:

1. Копирование выбранного утверждения. Выбранное утверждение копируется в соответствующий набор(ы) без изменений.
2. Создание нового утверждения на основе свойств выбранного утверждения.

# Подсистема оповещения

В данном разделе собраны статьи о подсистеме оповещения. Раздел содержит описание интерфейсов для разработки плагинов отправки SMS, Email, Push уведомлений.

- [Push-плагин](#).

# Push-плагин

КриптоПро DSS может оповещать пользователей, использующих мобильные приложения на основе DSS SDK, о различных событиях DSS:

- выпуске сертификата
- запросе на подпись документов
- изменении учётной записи
- и т.п.

Список событий, для которых доступно оповещение пользователей, приведён в разделе [Оповещение](#)

Отправка Push-уведомлений осуществляется через подключаемый к Центру Идентификации DSS модуль оповещения типа `Push`. Модуль оповещения состоит из двух плагинов:

- Плагин форматирования - плагин для формирования тела Push-уведомления.
- Транспортный плагин - плагин для отправки Push-уведомлений.

## Примечание

В состав КриптоПро DSS входит модуль для отправки Push-уведомлений через сервисы FCM (Google) и APN (Apple).

Алгоритм оповещения пользователей

- При вызове методов API DSS создаёт событие

## Примечание

Событие представляет собой объект типа `DSSTargetedMessage`. Объект содержит:

- идентификатор (код) события
  - Идентификатор пользователя (ID-пользователя, логин)
  - Параметры события (например, сертификат подписи, данные о документе и т.п.)
- Событие передаётся в сервис оповещения пользователей
  - Сервис оповещения передаёт событие в модули оповещения доступные для пользователя

## Примечание

Сервис оповещения собирает доступные для пользователя каналы оповещения (SMS, Email, Push). Доступные каналы оповещения настраиваются в профиле пользователя:

- Номер телефона для SMS-оповещения
  - Адрес электронной почты для Email-оповещения
  - Push-адрес приложения на мобильном устройстве пользователя для Push-уведомлений Push-адрес задаётся в профиле пользователя при инициализации мобильного приложения на основе DSS SDK.
- Модуль оповещения передаёт событие в *плагин форматирования*
  - *Плагин форматирования* на основе данных события формирует сообщение для пользователя

## Примечание

*Сообщение* может быть любым объектом (строкой, .NET-объектом), с которым умеет работать транспортный плагин.

- Сформированное *Плагином форматирования* сообщение и адрес получателя модуль оповещения передаёт в *транспортный плагин*
- *Транспортный плагин* осуществляет отправку сообщения пользователю

Реализация плагинов для Push модуля оповещения



Плагин представляет собой динамическую библиотеку для .NET Framework 4.8 и старше. Библиотека должна содержать два класса, реализующие интерфейсы:

- [IPushPlugin](#) - транспортный плагин
- [IMessageFormatter](#) - плагин форматирования

Интерфейсы [IPushPlugin](#), [IMessageFormatter](#) описаны в библиотеке CryptoPro.DSS.Common.dll.

### Инициализация плагинов

Плагины инициализируются один раз при старте сервиса Центр Идентификации КриптоПро DSS. Для инициализации плагина вызывается метод [IDssPlugin.Initialize](#). Метод принимает словарь параметров, задаваемый при регистрации плагина.

### Оповещение пользователей

Модуль оповещения вызывает метод `Format` у [плагина форматирования](#) и передаёт описание события [DSSTargetedMessage](#). Плагин форматирования должен сформировать сообщение для отправки через транспортный плагин. Сообщение должно быть сериализовано в строку.

Полученное сериализованное сообщение, [событие](#) и [адрес получателя](#) модуль оповещения передаёт в [транспортный плагин](#).

### Регистрация плагинов

- Плагин должен быть скопирован на сервер DSS в папку C:\Program Files\Crypto Pro\DSS\Plugins\Custom
- Регистрация и настройка плагина выполняется в консоли Powershell с помощью команд `Add-DssStsPlugin`, `Set-DssStsPlugin`.