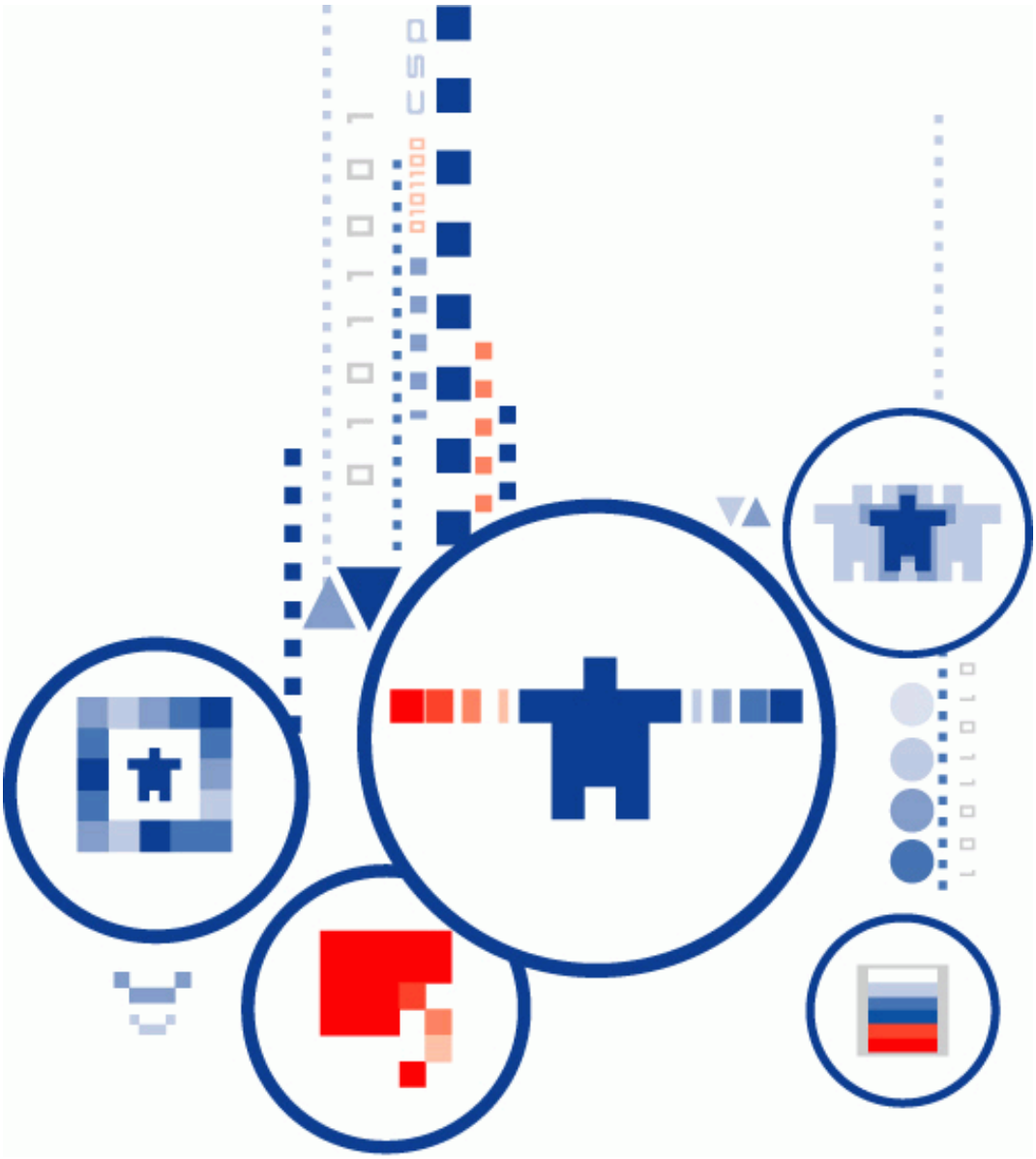

Сервер Электронной Подписи

«КриптоПро DSS»

Компонент ПАКМ «КриптоПро HSM»

DSS Client SDK. Руководство разработчика.

Android

СОДЕРЖАНИЕ

1. Общие сведения о DSS Client SDK

DSS Client SDK для встраивания в мобильное приложение представляет собой набор программных компонентов (framework) для использования в мобильных приложениях с функциями клиента КриптоПро DSS.

DSS Client 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 Client 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 TouchID/FaceID для защиты устанавливаемых векторов аутентификации.

  Ввод пароля и TouchID/FaceID. Окно отображается, если на сервере разрешена биометрия для защиты векторов аутентификации.

  В окне пользователю предлагается придумать ПИН-код для защиты вектора, после чего защитить его с помощью биометрии.
- Ввод пароля для доступа к векторам аутентификации.

  Окно отображается, если на сервере разрешен только пароль для защиты векторов.
- Ввод TouchID/FaceID для доступа к векторам аутентификации.

Ввод пароля или TouchID/FaceID: окно отображается, если на сервере разрешена биометрия для защиты векторов аутентификации.

- Подпись: сопровождающий текст и список документов.
- Просмотр документа.
- Просмотр «сырого» представления документа.

2. Интеграция DSS Client SDK в программный проект

Интеграция DSS Client SDK в программный проект состоит из следующих этапов.

1. DSSClient **.aar**-файл добавить в проект с помощью команды **file - new module - import .JAR/.AAR package**.
2. В gradle модуля в блок dependencies добавить строку **implementation project(':LIBNAME')**, где **LIBNAME** - имя библиотеки из файла **settings.gradle**.
3. В **gradle** модуля в блок **dependencies** добавить

```
implementation 'androidx.appcompat:appcompat:1.1.0'
implementation 'com.google.android.material:material:1.0.0'
implementation 'com.squareup.retrofit2:retrofit:2.5.0'
implementation 'com.squareup.okhttp3:logging-interceptor:3.12.0'
 implementation 'com.squareup.retrofit2:converter-gson:2.5.0'
implementation 'com.squareup.retrofit2:converter-scalars:2.5.0'
implementation 'com.fasterxml.jackson.core:jackson-databind:2.9.0'
implementation 'com.google.code.gson:gson:2.8.5'
implementation 'com.google.guava:guava:28.1-android'
implementation 'com.google.android.gms:play-services-vision:19.0.0'
implementation 'androidx.lifecycle:lifecycle-extensions:2.1.0'
```

При использовании DSSClientExternalCSP.aar требуется также добавить:

```
implementation 'org.ini4j:ini4j:0.5.1'
```

4. В **gradle** модуля в блок **android** добавить

```
dexOptions { preDexLibraries = false javaMaxHeapSize "4g" }
compileOptions { targetCompatibility 1.8 sourceCompatibility 1.8 }
packagingOptions { exclude 'META-INF/Digest.CP' exclude 'META-
INF/Sign.CP'
exclude 'META-INF/NOTICE.txt' exclude 'META-INF/LICENSE.txt' }
```

5. В **gradle** модуля добавить блок

```
configurations.all { resolutionStrategy.eachDependency {
DependencyResolveDetails details -> def requested = details.requested if
(requested.group == "com.android.support") { if
(!requested.name.startsWith("multidex")) { details.useVersion "26.+" } }
} }
```

6. Если используется **Jetifier** (опция **android.enableJetifier=true**) тогда необходимо **gradle.properties** проекта добавить **android.jetifier.blacklist = LIBNAME**, где **LIBNAME** - имя библиотеки из файла **settings.gradle**.
7. Для корректной работы DSS Client SDK необходим **json**-файл с сертификатами в папке **assets** модуля **certs.json** следующего формата: **{ "version":1, "root": [] "intermediate" : [] }**, где в элементах **root** и **intermediate** должен находиться массив сертификатов (корневые и промежуточные соответственно) в кодировке base64.
8. Для работы с DSS Client SDK необходимо вызвать **CryptoProDss().getInstance().init(context)**, где **context** — текущий контекст приложения.
9. Для корректной работы КриптоПро CSP необходимо в методах **onResume activity** вызвать **CryptoProDss().getInstance().registerActivityContext(context)**, где **context** - текущий контекст приложения.

3. Сценарии использования

3.1. Анонимная регистрация пользователя

Процедура анонимной регистрации пользователя на сервисе включает в себя следующие шаги:

1. Метод **init** (см. раздел 4.3.2). Добавление нового неподтвержденного устройства пользователя и установка вектора аутентификации на него.
2. Метод **confirm** (см. раздел 4.3.5). Подтверждение установки вектора аутентификации на устройстве пользователя.
3. Метод **scanQR** (см. раздел 4.3.1). Считывание **qr verification** и его загрузка в оперативную память
4. Метод **verify** (см. раздел 4.3.6). Отображение пользователю его профиля и используя загруженный qr, метод генерирует ответ с подтверждением привязки устройства к учетной записи пользователя на сервисе.

Важно: до выполнения этого шага должен быть создан профиль пользователя.

Пример:

```
Context context = getContext();
DssUser dssUser = new DssUser();
dssUser.setServiceUrl("service Url");
dssUser.setAlias("alias");
RegisterInfo registerInfo = new RegisterInfo("push address", "app
version");
Auth auth = new Auth();
auth.init(context, dssUser, registerInfo,
Constants.KeyProtectionType.PASSWORD, null, new SdkDssUserCallback() {
   @Override
   public void onOperationSuccessful(@NonNull AuthVector authVector) {
       auth.confirm(context, authVector.getKid(), new SdkCallback() {
           @Override
           public void onOperationSuccessful() {
               auth.scanQr(context, null, new SdkQrCallback() {
                   @Override
                   public void onOperationSuccessful(@NonNull String s)
{
                           auth.verify(context, authVector.getKid(),
false, new SdkCallback() {
                               @Override
                               public void onOperationSuccessful() {
                                   //Успех
                               }

                               @Override
                               public void onOperationFailed(int i,
@Nullable String s, @Nullable Throwable throwable) {
                                   //Ошибка
                               }
                           });

                   }

                   @Override
                   public void onOperationFailed(int i, @Nullable String
s, @Nullable Throwable throwable) {
                       //Ошибка
```

```
                    }

                    @Override
                    public void onOperationCancelled() {
                    }
                });
            }

            @Override
            public void onOperationFailed(int i, @Nullable String s,
@Nullable Throwable throwable) {
                //Ошибка
            }
        });
    }

    @Override
    public void onOperationFailed(int i, @Nullable String s, @Nullable
Throwable throwable) {
        //Ошибка
    }
});
```

3.2. Регистрация устройства с использованием начального вектора аутентификации

Процедура регистрации устройства на сервисе с использованием начального вектора аутентификации включает в себя следующие шаги:

1. Метод **scanQR** (см. раздел 4.3.1). Считывание **qr kinit** и его загрузка в оперативную память
2. Метод **kinit** (см. раздел 4.3.3). Установка ключей на устройстве с использованием считанного qr для отправки аутентифицированного запроса
3. Метод **confirm** (см. раздел 4.3.5). Подтверждение установки ключей на устройстве
4. Метод **verify** (см. раздел 4.3.6). Отображает пользователю его профиль и отправляет ответ с подтверждением привязки устройства к учетной записи пользователя на сервисе

Пример:

```
Context context = getContext();
DssUser dssUser = new DssUser();
RegisterInfo registerInfo = new RegisterInfo("push address", "app
version");
Auth auth = new Auth();
auth.scanQr(context, null, new SdkQrCallback() {
    @Override
    public void onOperationSuccessful(@NonNull String s) {
        auth.kinit(context, dssUser, registerInfo,
Constants.KeyProtectionType.BIOMETRIC, null, null, new
SdkDssUserCallback() {
            @Override
            public void onOperationSuccessful(@NonNull AuthVector
authVector) {
                auth.confirm(context, authVector.getKid(), new
SdkCallback() {
                    @Override
                    public void onOperationSuccessful() {
```

```
                        auth.verify(context, authVector.getKid(), false,
new SdkCallback() {
                            @Override
                            public void onOperationSuccessful() {
                                //Успех
                            }

                            @Override
                            public void onOperationFailed(int i,
@Nullable String s, @Nullable Throwable throwable) {
                                //Обработка ошибок
                            }
                        });
                    }

                    @Override
                    public void onOperationFailed(int i, @Nullable String
s, @Nullable Throwable throwable) {
                        //Обработка ошибок
                    }
                });
            }

            @Override
            public void onOperationFailed(int i, @Nullable String s,
@Nullable Throwable throwable) {
                //Обработка ошибок
            }
        });
    }

    @Override
    public void onOperationFailed(int i, @Nullable String s, @Nullable
Throwable throwable) {
        //Обработка ошибок
    }

    @Override
    public void onOperationCancelled() {

    }
});
```

3.3. Добавление (привязка) еще одного устройства к учетной записи пользователя

Процедура добавления (привязки) другого (еще одного) устройства к учетной записи пользователя на сервисе включает в себя следующие шаги:

1. Метод **addNewDevice** (см. раздел 4.3.4). Установка ключей на другом устройстве пользователя. Выполняется с устройства 2.
2. Метод **confirmNewDevice** (см. раздел 4.3.12). Одобрение / отклонение привязки ключей к учетной записи пользователя. Выполняется с устройства 1.
3. Метод **checkStatus** (см. раздел 4.3.13). Проверка статуса привязки ключей к учетной записи пользователя. Выполняется с устройства 2.

4. Использование DSS Client SDK. Классы и функции

DSS Client SDK предоставляет следующие основные классы для работы:

- **Класс CSPProvider**: Содержит метод инициализации DSS Client SDK и определение активного контекста для интеграции графического интерфейса (см. раздел 4.2).
- **Класс Auth**: Управляет устройствами пользователей (см. раздел 4.3).
- **Класс Cert**: Управляет сертификатами пользователей (см. раздел 4.4).
- **Класс Policy**: Содержит методы получения настроек КриптоПро DSS (см. раздел 4.5).
- **Класс Docs**: Управляет передачей документов в КриптоПро DSS при подписи (см. раздел 4.6).
- **Класс Sign**: Получает информацию об операциях, подтверждает операции и подписывает документы (см. раздел 4.7).

4.1. Настройка TLS-соединения

Взаимодействие DSS Client SDK с сервером КриптоПро DSS осуществляется по протоколу TLS с использование только алгоритмов ГОСТ. Для обеспечения доверия клиента к серверу КриптоПро DSS в ресурсы приложения должен быть добавлен корневой сертификат Веб-сервера DSS. Корневой сертификат добавляется через файл certs.json.

Пример содержания файла certs.json:

```
{
    "version":1,
    "root": ["MIIFxzCCBXSgAwIBAgIRAdSExQ … 15ktc8p00v+A9Erolsd5Ig=="],
    "intermediate": []
}
```

В элементе root передаётся один или несколько корневых сертификатов Веб-сервера в кодировке Base64.

Элемент intermediate (опциональный) может содержать список сертификатов подчинённых УЦ.

4.2. Класс CSPProvider

4.2.1. Метод init (Context context)

Инициализирует DSS Client SDK.

```
boolean init(Context context)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.

Возвращаемые значения:

- **initOk** — инициализация прошла успешно.
- **initCertNotInstalled** — не удалось установить сертификаты в процессе инициализации.

- **initLockScreenNotInstalled** — экран блокировки не установлен.
- **initDeviceRootedOrHasSpyPrograms** — устройство работает с правами суперпользователя или обнаружены программы с правами отображения поверх других окон.
- **initCspNotInitialized** — КриптоПро CSP не был инициализирован.

4.2.2. Метод registerActivityContext

Регистрирует базовую **Activity** с интеграцией классов DSS Client SDK.

```
void registerActivityContext(Context context)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.

4.3. Класс Auth

4.3.1. Метод scanQR

Загружает данные, переданные в виде QR-кода.

```
void scanQr(@NonNull Context context, @Nullable String base64Qr,
@NonNull final SdkQrCallback callback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **base64QR** — QR-код, закодированный в base64.
- **callback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.3.2. Метод init

Создает неподтвержденное мобильное устройство (без привязки) в КриптоПро DSS с получением вектора аутентификации к нему.

```
void init(@NonNull Context context, @NonNull DssUser dssUser, @NonNull
RegisterInfo registerInfo, @NonNull KeyProtectionType keyProtectionType,
@Nullable String password, @NonNull final SdkDssUserCallback
sdkDssUserCallback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **dssUser** — сведения о пользователе (см. раздел 5.1).
- **registerInfo** — сведения о регистрируемом устройстве пользователя (см. раздел 5.2).
- **keyProtectionType** — способ защиты вектора аутентификации (см. раздел 5.3).

- **password** — ПИН-код для доступа к вектору аутентификации.
- **sdkDssUserCallback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.3.3. Метод kinit

Создает неподтвержденное мобильное устройство (без привязки) в КриптоПро DSS с получением начального вектора аутентификации к нему с использованием QR-кода.

```
void kinit(@NonNull Context context, @NonNull DssUser dssUser, @NonNull
RegisterInfo registerInfo, @NonNull KeyProtectionType keyProtectionType,
@Nullable String activationCode, @Nullable String password, @NonNull
final SdkDssUserCallback sdkDssUserCallback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **dssUser** — сведения о пользователе (см. раздел 5.1).
- **registerInfo** — сведения о регистрируемом устройстве пользователя (см. раздел 5.2).
- **keyProtectionType** — способ защиты вектора аутентификации (см. раздел 5.3).
- **activationCode** — код активации.
- **password** — ПИН-код для доступа к вектору аутентификации.
- **sdkDssUserCallback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.3.4. Метод addNewDevice

Инициализирует новое устройство пользователя.

```
void addNewDevice(@NonNull Context context, @NonNull DssUser dssUser,
@NonNull RegisterInfo registerInfo, @NonNull KeyProtectionType
keyProtectionType, @NonNull String uid, @Nullable String password,
@NonNull final SdkDssUserCallback authCallback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **dssUser** — сведения о пользователе (см. раздел 5.1).
- **registerInfo** — сведения о регистрируемом устройстве пользователя (см. раздел 5.2).
- **keyProtectionType** — способ защиты вектора аутентификации (см. раздел 5.3).
- **uid** — идентификатор пользователя КриптоПро DSS, к которому будет привязано мобильное устройство.
- **password** — ПИН-код для доступа к вектору аутентификации.
- **authCallback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.3.5. Метод confirm

Подтверждает установку векторов аутентификации. Данный метод всегда необходимо вызывать после выполнения регистрации нового неподтвержденного мобильного устройства в КриптоПро DSS. Данный метод может быть вызван только для векторов аутентификации, находящихся в состоянии **Created**.

```
void confirm(@NonNull Context context, @NonNull String kid, @NonNull
final SdkCallback callback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **callback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.3.6. Метод verify

Подтверждает привязку мобильного устройства к учетной записи пользователя. Данный метод может быть вызван только для векторов аутентификации, находящихся в состоянии **NotVerified**.

Если для данного мобильного устройства требуется осуществлять подтверждение присоединения с использованием **nonce**, то в этом случае перед отправкой запроса мобильное приложение должно отсканировать QR-код, содержащий значение **nonce**.

```
void verify(@NonNull Context context, @NonNull String kid, boolean
silent, @NonNull final SdkCallback callback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **silent** — флаг для скрытия/отображения диалоговых окон SDK. Используется только для создания усиленной неквалифицированной электронной подписи.
- **callback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.3.7. Метод setPassAuth

Ввод ПИН-кода на вектор аутентификации.

```
void setPassAuth(@NonNull Context context, @NonNull String kid,
@Nullable String password, @NonNull final SdkCallback sdkCallback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.

- **password** — ПИН-код для защиты вектора аутентификации. Параметр используется только для создания усиленной неквалифицированной электронной подписи.
- **sdkCallback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.3.8. Метод changePassAuth

Изменение ПИН-кода на вектор аутентификации.

```
void changePassAuth(@NonNull Context context, @NonNull String kid,
@NonNull KeyProtectionType keyProtectionType, @Nullable String password,
@Nullable String newPassword, @NonNull final SdkCallback sdkCallback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **keyProtectionType** — способ защиты вектора аутентификации (см. раздел 5.3).
- **password** — существующий ПИН-код (параметр используется только в **silent**-режиме и для создания усиленной неквалифицированной электронной подписи).
- **newPassword** — новый ПИН-код (параметр используется только в **silent**-режиме и для создания усиленной неквалифицированной электронной подписи).
- **sdkCallback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.3.9. Метод renameAuth

Переименование устройства пользователя.

```
void renameAuth(@NonNull Context context, @NonNull String kid, @NonNull
String newName)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **newName** — новое отображаемое имя вектора аутентификации.

4.3.10. Метод removeAuth

Удаление устройства пользователя и его вектора аутентификации.

```
void removeAuth(@NonNull Context context, @NonNull String kid, @NonNull
String deletedKid, boolean forceDelete, @NonNull final SdkCallback
sdkCallback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.

- **kid** — идентификатор устройства пользователя.
- **deletedKid** — идентификатор удаляемого устройства пользователя.
- **forceDelete** — удаление устройства вне зависимости ответа сервера.
- **sdkCallback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.3.11. Метод getAuthList

Получение сведений о зарегистрированных пользователях и их устройствах.

```
List<DssUser> getAuthList(@NonNull Context context)
```

Список полей типа **DSSUser** приведен в разделе 5.1.

4.3.12. Метод confirmNewDevice

Подтверждение запроса на добавление нового устройства.

```
void confirmNewDevice(@NonNull Context context, @NonNull String kid,
@NonNull String confirmedKid, boolean silent, @NonNull final SdkCallback
callback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **confirmedKid** — идентификатор устройства пользователя, добавление которого требуется подтвердить.
- **silent** — флаг для скрытия/отображения диалоговых окон SDK. Используется только для создания усиленной неквалифицированной электронной подписи.
- **callback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.3.13. Метод checkStatus

Проверка статуса запроса на добавление нового устройства.

```
void checkStatus(@NonNull Context context, @NonNull String kid, @NonNull
final SdkKeyStateCallback callback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **callback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.4. Класс Cert

4.4.1. Метод getCert

Создание запроса на сертификат ключа проверки электронной подписи.

```
void getCert(@NonNull Context context, @NonNull String kid, int caid,
@NonNull String tid, @NonNull Map<String, String> dn, @NonNull final
SdkCertificateCallback sdkCertificateCallback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **caId** — идентификатор обработчика УЦ.
- **tid** — идентификатор шаблона сертификата.
- **dn** — различительное имя субъекта в формате {"OID компонента имени", "Значение компонента имени"}.
- **cert** — сведения о созданном сертификате / запросе на сертификат при отсутствии ошибок (см. раздел 5.4).
- **sdkCertificateCallback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.4.2. Метод setCert

Установка сертификата ключа проверки электронной подписи.

```
void setCert(@NonNull Context context, @NonNull String kid, @NonNull
String crt, @NonNull final SdkCallback sdkCallback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **crt** —сертификат, закодированный в base64.
- **sdkCallback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.4.3. Метод getCertList

Получение списка запросов на сертификаты и списка сертификатов ключей проверки электронной подписи.

```
void getCertList(@NonNull Context context, @NonNull String kid, @NonNull
final SdkCertificateListCallback certCallback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.

- **certs** — список сертификатов и запросов на сертификат, возвращаемые при отсутствии ошибок (см. раздел 5.4).
- **certCallback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.4.4. Метод setNameCert

Установка отображаемого имени сертификата ключа проверки электронной подписи.

```
void setNameCert(@NonNull Context context, @NonNull String kid, @NonNull String cid, @NonNull String friendlyName, @NonNull final SdkCallback sdkCallback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **cid** — идентификатор сертификата.
- **frendlyName** — отображаемое имя сертификата.
- **sdkCallback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.4.5. Метод suspendCert

Приостановление действия сертификата ключа проверки электронной подписи.

```
void suspendCert(@NonNull Context context, @NonNull String kid, @NonNull String cid, long holdDate, long unholdDate, @NonNull final SdkCallback sdkCallback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **cid** — идентификатор сертификата.
- **holdDate** — дата приостановления сертификата. 0 — если требуется немедленное приостановление сертификата, конкретная дата — если требуется отложенное приостановление сертификата.
- **unholdDate** — дата возобновления действия сертификата.
- **sdkCallback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.4.6. Метод resumeCert

Возобновление действия сертификата ключа проверки электронной подписи.

```
void resumeCert(@NonNull Context context, @NonNull String kid, @NonNull String cid, @NonNull final SdkCallback sdkCallback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **cid** — идентификатор сертификата.
- **sdkCallback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.4.7. Метод revokeCert

Отзыв сертификата ключа проверки электронной подписи.

```
void revokeCert(@NonNull Context context, @NonNull String kid, @NonNull String cid, int reason, long date, @NonNull final SdkCallback sdkCallback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **cid** — идентификатор сертификата.
- **reason** — причина отзыва:
  0 (CRL_REASON_UNSPECIFIED) — Причина неизвестна.
  1 (CRL_REASON_KEY_COMPROMISE) — Компрометация ключей.
  2 (CRL_REASON_CA_COMPROMISE) — Компрометация Центра Сертификации.
  3 (CRL_REASON_AFFILIATION_CHANGED) — Имя пользователя или другая информация в сертификате изменена, но нет причины полагать, что секретный ключ скомпрометирован.
  4 (CRL_REASON_SUPERSEDED) — Сертификат заменен другим, но нет причины полагать, что секретный ключ скомпрометирован.
  5 (CRL_REASON_CESSATION_OF_OPERATION) — Сертификат более не нужен для целей, которых он выдавался, но нет причины полагать, что секретный ключ скомпрометирован.
- **date** — дата отзыва сертификата. 0 — если требуется немедленный отзыв сертификата, конкретная дата — если требуется отложенный отзыв сертификата.
- **sdkCallback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.4.8. Метод setDefaultCert

Метод установки сертификата ключа проверки электронной подписи сертификатом по умолчанию.

```
void setDefaultCert(@NonNull Context context, @NonNull String kid, @NonNull String cid, boolean def, @NonNull final SdkCallback sdkCallback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **cid** — идентификатор сертификата.
- **def** — флаг использования сертификата как сертификата по умолчанию.
- **sdkCallback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.4.9. Метод deleteCert

Метод удаления сертификата или запроса на сертификат ключа проверки электронной подписи.

```
void deleteCert(@NonNull Context context, @NonNull String kid, String
cid, String rid, @NonNull final SdkCallback sdkCallback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **cid** — идентификатор сертификата.
- **rid** — идентификатор запроса на сертификат.
- **sdkCallback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.5. Класс Policy

4.5.1. Метод getOperations

Получение списка операций пользователя данного устройства.

```
void getOperations(@NonNull Context context, @NonNull String kid,
@Nullable String type, @Nullable String opId, @NonNull final
SdkPolicyOperationsInfoCallback sdkPolicyOperationsInfoCallback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **type** — тип операции. Типы операций описаны в документе «ЖТЯИ.00096-02 91 02 КриптоПро DSS. Руководство Администратора»
- **opId** — идентификатор операции.
- **operationsInfo** — список операций пользователя данного устройства (см. раздел 5.7).
- **sdkPolicyOperationsInfoCallback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.5.2. Метод getHistoryOperations

Получение записей аудита для определенного пользователя и устройства

```
void getHistoryOperations(@NonNull Context context, @NonNull String kid,
@Nullable Integer count, @Nullable Integer bookmark, @Nullable
ArrayList<Integer> operationCodes, @NonNull final
SdkPolicyOperationHistoryCallback callback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **count** — выводимое количество записей.
- **bookmark** — идентификатор записи, относительно которой осуществляется поиск.
- **operationCodes** — разделенный запятой список кодов операций, которые должны быть включены в выборку (см. «ЖТЯИ.00096-02 91 02 КриптоПро DSS. Руководство Администратора»).
- **operationHistory** — список записей аудита для определенного пользователя и устройства (см. раздел 5.10).
- **callback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.5.3. Метод getParamDSS

Запрос параметров сервера КриптоПро DSS.

```
void getParamsDSS(@NonNull Context context, @NonNull String serviceUrl,
@NonNull final SdkPolicyParamsDssCallback sdkPolicyParamsDssCallback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **serviceURL** — адрес сервера КриптоПро DSS.
- **policy** — параметры сервера КриптоПро DSS (см. раздел 5.12).
- **sdkPolicyParamsDssCallback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.5.4. Метод setPersonalisation

Персонализация элементов интерфейса SDK.

```
void setPersonalisation(URI uri)
```

Параметры:

- **uri** — путь к файлу персонализации (см. Приложения 1–2).

4.5.5. Метод getUserDevices

Получение сведений об устройствах пользователя.

```
void getUserDevices(@NonNull Context context, @NonNull String kid,
@NonNull final SdkPolicyDevicesCallback callback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **devices** — сведения об устройствах пользователя (см. раздел 5.17).
- **callback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.5.6. Метод getCaParams

Запрос с сервера DSS параметров подписи: список профилей подписи, параметры Удостоверяющих Центров и т.п.

```
void getCaParams(@NonNull Context context, @NonNull String kid, @NonNull
final SdkPolicyCaParamsCallback sdkPolicyCaParamsCallback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **policy** — параметры подписи, полученные от КриптоПро DSS (см. раздел5.14).
- **sdkPolicyCaParamsCallback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.5.7. Метод initBioRng

Открывает окно ДСЧ.

```
void generateKeyPair(NonNull Context context)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.

4.6. Класс Docs

4.6.1. Метод uploadDocument

Загрузка документа в КриптоПро DSS.

```
void uploadDocument(@NonNull Context context, @NonNull String kid,
@NonNull UploadFile document, @NonNull final SdkUploadDocumentCallback
callback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **document** — информация о загружаемом документе (см. раздел 5.19).
- **uploadData** — информация о загруженном документе (см. раздел 5.20).
- **callback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.6.2. Метод downloadDocument

Выгрузка документа из КриптоПро DSS.

```
void downloadDocument(@NonNull Context context, @NonNull String kid,
@NonNull String docId, @NonNull final SdkGetDocumentCallback callback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **docId** — идентификатор документа.
- **callback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.7. Класс Sign

4.7.1. Метод signMT

Подтверждение операции, созданной на сервере КриптоПро DSS.

```
void signMT(@NonNull Context context, @NonNull String kid, @Nullable
Operation operation, boolean enableMultiSelection, boolean
immediateSendConfirm, boolean silent, final
SdkMtOperationWithSuspendCallback sdkMtOperationWithSuspendCallback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **operation** — сведения об операции (см. раздел 5.10).
- **enableMultiSelection** — режим выбора документов (по умолчанию «только все» — пользователь может подтвердить/отклонить операцию целиком или может выбрать отдельные документы из списка для подтверждения подписи) (см. раздел 5.22).
- **immediateSendConfirm** — режим отправки подтверждения (по умолчанию «немедленно» - сформированный запрос с подтверждением SDK сразу отправляет на сервер или приложение сохраняет данный запрос для возможности отправить его позднее) (см. раздел 5.23).
- **approveRequest** — запрос на подтверждение/отклонение операции (см. раздел 5.24).

- **silent** — флаг для скрытия/отображения диалоговых окон SDK. Используется только для создания усиленной неквалифицированной электронной подписи.
- **sdkMtOperationWithSuspendCallback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.7.2. Метод signMO

Подтверждение операции, созданной на клиенте (в мобильном приложении).

```
void signMO(@NonNull Context context, @NonNull String kid, @NonNull
Map<String, String> parameters, boolean enableMultiSelection, boolean
immediateSendConfirm, List<UploadDocInfo> uploadDocInfo, boolean silent,
final SdkMoOperationWithSuspendCallback callback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **parameters** — сведения об операции (см. раздел 5.21).
- **enableMultiSelection** — режим выбора документов (по умолчанию «только все» — пользователь может подтвердить/отклонить операцию целиком или может выбрать отдельные документы из списка для подтверждения подписи).
- **immediateSendConfirm** — режим отправки подтверждения (по умолчанию «немедленно» - сформированный запрос с подтверждением SDK сразу отправляет на сервер или приложение сохраняет данный запрос для возможности отправить его позднее).
- **uploadDocInfo** — список документов, переданных на подпись (см. раздел 5.20).
- **signatureResult** — Результат подписи пакета документов (см. раздел 5.25).
- **approveRequest** — запрос на подтверждение/отклонение операции (см. раздел 5.27).
- **silent** — флаг для скрытия/отображения диалоговых окон SDK. Используется только для создания усиленной неквалифицированной электронной подписи.
- **callback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.7.3. Метод deferredRequest (на сервере)

Отложенное подтверждение операции, созданной на сервере КриптоПро DSS.

```
void deferredRequest(@NonNull Context context, @NonNull String kid,
@NonNull ApproveRequestMT approveRequest, @NonNull final
SdkMtOperationCallback callback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **approveRequest** — запрос на подтверждение/отклонение операции (см. раздел 5.24).

- **callback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

4.7.4. Метод deferredRequest (на клиенте)

Отложенное подтверждение операции, созданной на клиенте (в мобильном приложении).

```
void deferredRequest(@NonNull Context context, @NonNull String kid,
@NonNull ApproveRequestMO approveRequest, @NonNull final
SdkMoOperationCallback callback)
```

Параметры:

- **context** — указатель на объект **Context** для получения доступа к базовым функциям приложения, в т.ч. для управления **Activity** при отображении окон графического интерфейса DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **approveRequest** — запрос на подтверждение/отклонение операции (см. раздел 5.27).
- **callback** — интерфейс обратного вызова для получения результата выполнения метода. Перечень ошибок приведен в Приложении 3.

5. Типы данных

5.1. Тип DSSUser

Поле	Тип	Описание
kid	String	Идентификатор устройства пользователя.
uid	String	Идентификатор пользователя DSS.
alias	String	Человекочитаемый идентификатор мобильного устройства пользователя.
state	String	Состояние начального вектора аутентификации (см. раздел 5.5).
profile	String	Профиль пользователя в DSS.
notBefore	Int64	Дата начала действия вектора аутентификации. Время в формате Unix Time.
notAfter	Int64	Дата окончания действия вектора аутентификации. Время в формате Unix Time.
serviceUrl	String	URL для взаимодействия с сервером DSS.
name	String	Дружественное устройства пользователя.
keyType	Bool	Тип вектора аутентификации. Для создания усиленной квалифицированной электронной подписи используется значение **`True`**.

5.2. Тип RegisterInfo

Поле	Тип	Описание
pushAddress	String	PUSH-адрес устройства пользователя.
appVersion	String	Версия приложения, в которое встроен DSS Client SDK (опционально).
userName	String	Логин пользователя (опционально)
phone	String	Номер телефона пользователя (опционально)
email	String	Адрес электронной почты пользователя (опционально)
token	String	Токен аутентификации интегрированной

Поле	Тип	Описание
		информационной системы (опционально)

5.3. Тип ProtectionType

Тип **`ProtectionType`** содержит способы защиты вектора аутентификации и может принимать следующие значения:

- PASSWORD — ПИН-код;
- NO_PROTECTION — без защиты;
- BIOMETRIC — биометрические данные (отпечаток пальца, лицо и т.д.).

5.4. Тип Certificate

Поле	Тип	Описание
type	String	Тип объекта: "crt" - сертификат, "req" - запрос на сертификат.
cid	String	Идентификатор сертификата.
rid	String	Идентификатор запроса на сертификат.
content	String	Содержимое сертификата.
caId	Int	Идентификатор обработчика УЦ.
dn	[String: String]	Различительное имя субъекта в формате {"OID компонента имени", "Значение компонента имени"}.
notBefore	Int32	Дата начала действия сертификата.
notAfter	Int32	Дата окончания действия сертификата.
state	String	Статус сертификата. Допустимые значения: **`Active`**, **`Not_valid`**, **`Revoked`**, **`Out_of_order`**.
friendlyName	String	Отображаемое имя сертификата.
isDefault	Bool	Флаг, определяющий, является ли данный сертификат сертификатом по умолчанию

5.5. Тип state

Поле	Описание
Created	Вектора аутентификации пользователя созданы.
Installed	Вектора аутентификации пользователя установлены на устройство пользователя.
NotVerified	Требуется подтверждение учетной записи пользователя.
Active	Устройство пользователя привязано и готово к использованию.
ApproveRequired	Требуется подтверждение привязки нового устройства пользователя.

5.6. Тип type (тип загруженного объекта)

Тип **`type`** указывает на информацию внутри QR-кода.

Поле	Описание
kinit	Начальный вектор аутентификации
verification	Сведения для подтверждения привязки устройства пользователя к учетной записи.
transaction	Сведения для подтверждения операции.
newdev	Сведения о новом добавляемом устройстве пользователя.

5.7. Тип OperationsInfo

Тип представляет собой список типа **`OperationInfo`** (см. раздел 5.10).

5.8. Тип OperationDescription

Поле	Тип	Описание
type	String	Тип операции (см. документ «ЖТЯИ.00096-02 91 02 КриптоПро DSS. Руководство Администратора»).
caption	String	Краткое описание операции.
description	String	Описание операции.

5.9. Тип Document

Поле	Тип	Описание
id	String	Идентификатор документа.
documentInfo	String	Сопровождающий текст о документе.
documentHash	String	Хэш-значение от документа.
snippet	String	Краткая информация о документе в формате html.
snippetHash	String	Хэш-значение от краткой информации о документе.
fileSize	Int64	Размер документа в мегабайтах.
pageCount	Int	Количество страниц документа.
isPrintableViewAvailable	Bool	Флаг, определяющий, доступность в мобильном приложении печатной формы документа.
isSnippetViewAvailable	Bool	Флаг, определяющий доступность в мобильном приложении краткой информации о документе.
isRawViewAvailable	Bool	Флаг, определяющий доступность в мобильном приложении исходного документа.

5.10. Тип Operation

Поле	Тип	Описание
description	OperationDescription	Описание операции (см. раздел 5.8).
createdAt	Int64	Дата и время создания операции Unix Time.
expiresAt	Int64	Дата и время истечения операции Unix Time.
documentCount	Int	Количество документов в операции.
transactionId	String	Идентификатор транзакции.
parameters	[String: String]	Параметры операции (см. раздел 5.21).
documents	[Document]	Один или несколько документов, с которыми совершается операция (см. раздел 5.9).

5.11. Тип OperationHistory

Поле	Тип	Описание
records	[AuditRecord]	Список записей аудита. Записи аудита приводятся в документе «ЖТЯИ.00096-02 91 02 КриптоПро DSS. Руководство Администратора».
totalCount	Int	Количество выведенных записей аудита.
bookmark	Int	Идентификатор записи, относительно которой осуществляется поиск.

5.12. Тип PolicyPayload

Поле	Тип	Описание
selfRegistrationEnabled	Bool	Флаг, позволяющий/запрещающий самостоятельную регистрацию пользователя.
externalLoginRequired	Bool	Флаг, позволяющий/запрещающий «прозрачную» регистрацию пользователей.
keyActivationRequired	Bool	Флаг, определяющий, требуется ли ввод кода активации.
keyProtectionFlags	KeyProtectionFlags	Требования к защите вектора аутентификации (см. раздел 5.13).
keyActivationTypes	[String]	Список способов доставки кода активации. Допустимые значения: `SMS`, `Email`.

5.13. Тип keyProtectionFlags

Поле	Тип	Описание
fingerprintRequired	Bool	Требуется привязка векторов аутентификации к устройству.
collectEvents	Bool	Зарезервирован для использования в будущем.
collectDeviceInfo	Bool	Зарезервирован для использования в будущем.
collectSimInfo	Bool	Зарезервирован для использования в будущем.

Поле	Тип	Описание
collectLocation	Bool	Зарезервирован для использования в будущем.
passwordPolicy	Int	Требования к паролю. Допустимые значения: 0 – нет требований, 1 – только цифры, 2 – цифры и буквы, 3 – цифры, буквы и специальные символы.
denyOSProtection	Bool	Разрешена / запрещена биометрия для защиты векторов аутентификации.
scoringEnabled	Bool	Зарезервирован для использования в будущем.
strongKeyProtectionType	Bool	Тип векторов аутентификации. Для создания усиленной квалифицированной электронной подписи используется значение **True**.

5.14. Тип SignServerPolicy

Поле	Тип	Описание
CAPolicy	[CAPolicy]	Параметры обработчика УЦ (см. раздел 5.15).
ProcessingTemplateInfo	[ProcessingTemplateInfo]	Список шаблонов подписи (см. раздел 5.16)

5.15. Тип CAPolicy

Поле	Тип	Описание
cryptoProviderInfos	[String: [CryptoProviderInfo]]	Идентификаторы криптопровайдера. Допустимые значения: 80, 81.
showInUi	Bool	Зарезервирован для использования в будущем.
extensionsPolicy	String	Расширения сертификата.
id	Int	Идентификатор обработчика УЦ.
name	String	Отображаемое имя обработчика УЦ.
active	Bool	Статус обработчика УЦ.

Поле	Тип	Описание
allowUserMode	Bool	Зарезервирован для использования в будущем.
snChangesEnable	Bool	Разрешить изменять имя субъекта в сертификате.
namePolicy	[NamePolicy]	Конфигурация компонентов имени пользователя.
ekuTemplates	[String: [String]]	Конфигурация шаблонов сертификатов пользователя.
caType	String	Тип обработчика УЦ.
validationMode	String	Режим проверки сертификата ЭП перед использованием. Возможные значения: **`ChainOffline`** - для локально установленного CRL, **`ChainOnline`** - для локально установленного или загруженного по сети CRL ИЛИ при помощи OCSP-службы, **`NoCheck`** - не проверять.

5.16. Тип ProcessingTemplateInfo

Поле	Тип	Описание
id	Int	Идентификатор шаблона подписи.
description	String	Отображаемое имя шаблона подписи.

5.17. Тип Devices

Тип представляет собой список типа **`deviceinfo`** (см. раздел 5.18).

5.18. Тип Deviceinfo

Поле	Тип	Описание
uid	String	Идентификатор пользователя DSS.
kid	String	Идентификатор устройства пользователя.
userName	String	Логин пользователя.
profile	String	Профиль пользователя в DSS.

Поле	Тип	Описание
nonceRequired		Для подтверждения учетной записи требуется сканирование QR-кода типа, указанного в типе **type** (см. раздел 5.6).
deviceName	String	Отображаемое имя устройства пользователя.
notBefore	Int64	Дата начала действия вектора аутентификации. Время в формате Unix Time.
notAfter	Int64	Дата окончания действия вектора аутентификации. Время в формате Unix Time.
state	String	Состояние начального вектора аутентификации (см. раздел 5.5).

5.19. Тип UploadFile

Поле	Тип	Описание
documentInfo	String	Сопровождающий текст о документе.
snippetTemplate	String	Html-шаблон для формирования краткой информации о документе.
previewTemplate	String	Html-шаблон для формирования печатной формы документа
url	URL	Идентификатор документа.

5.20. Тип UploadDocInfo

Поле	Тип	Описание
docId	String	Идентификатор документа.

5.21. Тип parameters

Поле	Тип	Описание
TemplateId	String	Идентификатор шаблона подписи.

5.22. Тип DocumentSelectedMode

Поле	Описание
allDocuments	Режим выбора документов «только все».
withSelected	Режим выбора отдельных документов для подписи.

5.23. Тип ConfirmationSendingMode

Поле	Описание
online	Режим онлайн-отправки кода подтверждения.
offline	Режим оффлайн-отправки кода подтверждения.

5.24. Тип ApproveRequestMT

Поле	Тип	Описание
approvedOperation	String	Данные операции, для которых вычисляется код подтверждения.
hmac	String	Код подтверждения.

5.25. Тип SignatureResult

Список **`OperationResultInfo (см. раздел 5.26).`**

5.26. Тип OperationResultInfo

Поле	Тип	Описание
refId	String	Идентификатор подписанного документа.
originalRefId	String	Идентификатор исходного документа.
status	String	Результат операции.
error	String	Код ошибки взаимодействия с DSS.
errorDescription	String	Описание ошибки взаимодействия с DSS.

5.27. Тип approveRequestMO

Поле	Тип	Описание
approvedOperation	ApprovedOperation	Структурированные данные операции, для которых вычисляется код подтверждения (см. раздел 5.28).

5.28. Тип ApprovedOperation

Поле	Тип	Описание
id	String	Идентификатор транзакции.
type	String	Тип операции (возвращается только в методе **`SignMT`**).
caption	String	Краткое описание операции (возвращается только в методе **`SignMT`**).
parameters	[String: String]	Параметры операции (см. раздел 5.21).
confirmedDocuments	[ConfirmedDocument]	Список подтверждаемых документов.
declinedDocuments	[DeclinedDocument]	Список отклоняемых документов.
timeStamp	Int64	Метка времени, полученная на момент вычисления кода подтверждения.

5.29. Тип ConfirmedDocument

Поле	Тип	Описание
id	String	Идентификатор документа.
documentHash	String	Хэш-значение от документа.
snippetHash	String	Хэш-значение от краткой информации о документе.

5.30. Тип DeclinedDocument

Поле	Тип	Описание
id	String	Идентификатор документа.
documentHash	String	Хэш-значение от документа.

Поле	Тип	Описание
snippetHash	String	Хэш-значение от краткой информации о документе.
reason	String	Причина отклонения операции с документом.

5.31. Тип Error

Содержит информацию об ошибках. Полный перечень ошибок приведен в Приложении 3.

Приложение 1. Файл стилизации графического интерфейса DSS Client SDK

Задание параметров стилей графического интерфейса DSS Client SDK производится при помощи файла стилизации **SDKStyles.json**. При необходимости разработчик может переопределять стили, тем самым персонализируя фрагменты графического интерфейса под дизайн своего приложения.

Значения по умолчанию файла стилизации **SDKStyles.json** представлены в Приложении 2. Ниже приведены пояснения по применению этих параметров.

При задании значений цветов используется цветовая модель **#RGBA**, где параметр A является альфа-каналом.

1. **header.** Стилизация заголовка окна.
   - 1.1. **backgroundColor.** Цвет фона хэдера. Применяется для всех окон sdk.
   - 1.2. **rightIcon.** Картинка (png) в формате base64. Данная картинка используется во view отображения списка документов в операции, ожидающий подтверждения пользователя (при signMT).
   - 1.3. **iconsTintColor**. Цвет вышеуказанной картинки.
   - 1.4. **title**. Стилизация заголовка окна (цвет, размер текста, шрифт).
2. **body.** Стилизация основной части окна.
   - 2.1. **backgroundColor.** Цвет основной части окна. Применяется для всех окон sdk.
   - 2.2. **signCell.** Стилизация ячейки, отображающая информацию об подписываемом документе. Используется при signMT и signMO.
     - 2.2.1. **cell.** Стилизация контура ячейки.
       - 2.2.1.1. **backgroundColor**. Цвет фона ячейки.
       - 2.2.1.2. **shadowStyle.** Наложение тени на ячейку.
       - 2.2.1.3. **borderStyle**. Стилизация границ ячейки.
     - 2.2.2. **textColor.** Цвет основного текста ячейки.
     - 2.2.3. **moreImage.** Картинка (png) в формате base64.
     - 2.2.4. **imageTintColor.** Цвет вышеуказанной картинки.
     - 2.2.5. **subMenu.** Стилизация всплывающего подменю.
       - 2.2.5.1. **backgroundColor**. Цвет фона подменю.
       - 2.2.5.2. **shadowStyle.** Наложение тени на подменю.
       - 2.2.5.3. **borderStyle**. Стилизация границ подменю.
     - 2.2.6. **subMenuText.** Стилизация текста подменю (цвет, размер текста, шрифт).
   - 2.3. **modalView.** Стилизация окна - предупреждения об отклонении операции.
     - 2.3.1. **backgroundColor**. Цвет фона окна.
     - 2.3.2. **text.** Стилизация текста - предупреждения (цвет, размер текста, шрифт).
     - 2.3.3. **saveChoiceText.** Стилизация текста - предупреждения о небезопасном использовании sdk (цвет, размер текста, шрифт).
     - 2.3.4. **switchOnTintColor**. Цвет тумблера.
     - 2.3.5. **switchThumbTintColor**. Цвет активного состояния тумблера.
     - 2.3.6. **shadowStyle.** Наложение тени на окно – предупреждение.
     - 2.3.7. **borderStyle**. Стилизация границ окна – предупреждения.
     - 2.3.8. **confirm.** Стилизация кнопки – подтверждения.
       - 2.3.8.1. **backgroundColor**. Цвет фона кнопки.
       - 2.3.8.2. **shadowStyle.** Наложение тени на кнопку.
       - 2.3.8.3. **borderStyle**. Стилизация границ кнопки.
       - 2.3.8.4. **title.** Стилизация текста кнопки (цвет, размер текста, шрифт).
     - 2.3.9. **reject.** Стилизация кнопки – отклонения.
       - 2.3.9.1. **backgroundColor**. Цвет фона кнопки.
       - 2.3.9.2. **shadowStyle.** Наложение тени на кнопку.
       - 2.3.9.3. **borderStyle**. Стилизация границ кнопки.

2.3.9.4. title. Стилизация текста кнопки (цвет, размер текста, шрифт).

2.4. profileCell. Стилизация ячейки, отображающая профиль пользователя. Используется в функции verify.

2.4.1. cell. Стилизация контура ячейки.

2.4.1.1. backgroundColor. Цвет фона ячейки.

2.4.1.2. shadowStyle. Наложение тени на ячейку.

2.4.1.3. borderStyle. Стилизация границ ячейки.

2.4.2. title. Стилизация заголовка ячейки (цвет, размер текста, шрифт).

2.4.3. subTitle. Стилизация основного текста ячейки (цвет, размер текста, шрифт).

2.5. textField. Стилизация textField. Используется в окнах ввода пароля.

2.5.1. backgroundColor. Цвет фона textField.

2.5.2. lineColor. Цвет линии.

2.5.3. shadowStyle. Наложение тени на textField.

2.5.4. borderStyle. Стилизация границ textField.

2.5.5. title. Стилизация заголовка textField (цвет, размер текста, шрифт).

2.5.6. text. Стилизация основного текста textField (цвет, размер текста, шрифт).

2.6. errorLabel. Стилизация текста - описания ошибки (цвет, размер текста, шрифт).

2.7. attemptsDescriptionLabel. Стилизация текста - описания количества попыток для ввода пароля (цвет, размер текста, шрифт).

2.8. hintLabel. Стилизация текста - подсказки об используемой сложности пароля (цвет, размер текста, шрифт).

2.9. operationInfoCell. Стилизация окна, отображающую подробную информацию об операции. Используется в signMT.

2.9.1. cell. Стилизация контура ячейки.

2.9.1.1. backgroundColor. Цвет фона ячейки.

2.9.1.2. shadowStyle. Наложение тени на ячейку.

2.9.1.3. borderStyle. Стилизация границ ячейки.

2.9.2. header. Стилизация заголовка окна (цвет, размер текста, шрифт).

2.9.3. title. Стилизация заголовка ячейки (цвет, размер текста, шрифт).

2.9.4. subTitle. Стилизация основного текста ячейки (цвет, размер текста, шрифт).

2.10. biometric. Стилизация окна биометрики (только Android).

2.10.1. shadowStyle. Наложение тени на окно биометрики (только Android)

2.10.2. borderStyle. Стилизация границ окна биометрики (только Android)

2.10.3. title. Стилизация заголовка окна биометрики (только Android)

2.10.4. errorTitle. Стилизация заголовка окна с ошибкой биометрики (только Android)

2.10.5. description. Стилизация текста подсказки окна биометрики (только Android)

2.10.6. errorDescription. Стилизация текста подсказки окна с ошибкой биометрики (только Android)

2.10.7. cancelButtonTitle. Стилизация текста кнопки отмены биометрики (только Android)

2.10.8. fingerprintIcon. Картинка (png) в формате base64 в окне с биометрикой (только Android)

2.10.9. fingerprintIconColor. Цвет вышеуказанной картинки (только Android)

2.10.10. errorIcon. Картинка (png) в формате base64 в окне с ошибкой биометрики (только Android)

2.10.11. errorIconColor. Цвет вышеуказанной картинки (только Android)

2.11. qrReader. Стилизация окна, сканирования qr кода.

2.11.1. description. Стилизация текста - подсказки (цвет, размер текста, шрифт).

2.11.2. error. Стилизация текста - описания ошибки (цвет, размер текста, шрифт).

2.11.3. galleryIcon. Картинка (png) в формате base64.

2.11.4. imageTintColor. Цвет вышеуказанной картинки.

3. footer. Стилизация футера окна.

3.1. backgroundColor. Цвет фона футера. Применяется для всех окон sdk.

3.2. shadowStyle. Наложение тени на футер.

3.3. borderStyle. Стилизация границ футера.

3.4. confirm. Стилизация кнопки – подтверждения.

3.4.1. backgroundColor. Цвет фона кнопки.

3.4.2. shadowStyle. Наложение тени на кнопку.

3.4.3. borderStyle. Стилизация границ кнопки.

3.4.4. title. Стилизация текста кнопки (цвет, размер текста, шрифт).

3.5. reject. Стилизация кнопки – отклонения.

3.5.1. backgroundColor. Цвет фона кнопки.

3.5.2. shadowStyle. Наложение тени на кнопку.

3.5.3. borderStyle. Стилизация границ кнопки.

3.5.4. title. Стилизация текста кнопки (цвет, размер текста, шрифт).

3.6. slider.

3.6.1. minimumTrackTintColor. Цвет минимального изображения дорожки.

3.6.2. maximumTrackTintColor. Цвет максимального изображения дорожки.

3.6.3. thumbTintColor. Цвет ползунка

3.7. pageCounter. Стилизация текста - количества страниц в pdf (цвет, размер текста, шрифт).

4. dialogString. Текстовка окон.

Приложение 2. Значения по умолчанию файла стилизации графического интерфейса DSS Client SDK

```
{
 "header": {
   "backgroundColor": "#311b92ff",
   "rightIcon": "",
   "iconsTintColor": "#ffffffff",
   "title": {
     "color": "#ffffffff",
     "size": 17,
     "font": ""
   },
   "biometricTitle": {
     "color": "#bdbdbdff",
     "size": 17,
     "font": ""
   },
   "biometricErrorTitle": {
     "color": "#512da8ff",
     "size": 17,
     "font": ""
   }
 },
 "body": {
   "backgroundColor": "#ffffffff",
   "signCell": {
     "cell": {
       "backgroundColor": "#ffffffff",
       "shadowStyle": {
         "shadowColor": "#000000ff",
         "shadowOffsetWidth": 0,
         "shadowOffsetHeight": 0,
         "shadowRadius": 0,
         "shadowOpacity": 0
       },
       "borderStyle": {
         "borderColor": "#191919FF",
         "borderWidth": 1.5
       }
     },
     "textColor": "#000000ff",
     "moreImage": "",
     "imageTintColor": "#000000ff",
     "subMenu": {
       "backgroundColor": "#ffffffff",
       "shadowStyle": {
         "shadowColor": "#434343ff",
         "shadowOffsetWidth": 0,
         "shadowOffsetHeight": 0,
         "shadowRadius": 3,
         "shadowOpacity": 1
       },
       "borderStyle": {
         "borderColor": "#464646ff",
         "borderWidth": 0
       }
     },
     "subMenuText": {
       "color": "#000000ff",
       "size": 15,
       "font": ""
     }
   },
   "modalView": {
     "backgroundColor": "#ffffffff",
     "text": {
       "color": "#000000ff",
       "size": 20,
       "font": ""
     },
     "shadowStyle": {
       "shadowColor": "#000000ff",
       "shadowOffsetWidth": 0,
       "shadowOffsetHeight": 0,
       "shadowRadius": 0,
       "shadowOpacity": 0
     },
     "borderStyle": {
       "borderColor": "#000000ff",
       "borderWidth": 0
     },
     "confirm": {
       "backgroundColor": "#4527a0ff",
       "shadowStyle": {
         "shadowColor": "#4527a000",
         "shadowOffsetWidth": 0,
         "shadowOffsetHeight": 0,
         "shadowRadius": 0,
         "shadowOpacity": 0
       },
       "borderStyle": {
         "borderColor": "#4527a0ff",
         "borderWidth": 0.3
       },
       "title": {
         "color": "#ffffffff",
         "size": 17,
         "font": ""
       }
     },
     "reject": {
       "backgroundColor": "#ffffffff",
       "shadowStyle": {
         "shadowColor": "#bdbdbd00",
         "shadowOffsetWidth": 0,
         "shadowOffsetHeight": 0,
         "shadowRadius": 0,
         "shadowOpacity": 0
       },
       "borderStyle": {
         "borderColor": "#dcdcdcFF",
         "borderWidth": 0.3
       },
       "title": {
         "color": "#512da8ff",
         "size": 17,
         "font": ""
       }
     }
   },
   "profileCell": {
     "cell": {
       "backgroundColor": "#00000000",
       "shadowStyle": {
         "shadowColor": "#000000ff",
         "shadowOffsetWidth": 0,
         "shadowOffsetHeight": 0,
         "shadowRadius": 0,
         "shadowOpacity": 0
       },
       "borderStyle": {
         "borderColor": "#000000ff",
         "borderWidth": 0
       }
     },
     "title": {
       "color": "#000000ff",
       "size": 15,
       "font": ""
     },
     "subTitle": {
       "color": "#000000ff",
       "size": 15,
       "font": ""
     }
   },
   "textField": {
     "backgroundColor": "#eeeeeeff",
```

```
    "lineColor": "#512da8ff",
    "shadowStyle": {
      "shadowColor": "#000000ff",
      "shadowOffsetWidth": 0,
      "shadowOffsetHeight": 0,
      "shadowRadius": 0,
      "shadowOpacity": 0
    },
    "borderStyle": {
      "borderColor": "#000000ff",
      "borderWidth": 0
    },
    "title": {
      "color": "#512da8ff",
      "size": 15,
      "font": ""
    },
    "text": {
      "color": "#000000ff",
      "size": 17,
      "font": ""
    }
  },
  "errorLabel": {
    "color": "#c62828ff",
    "size": 15,
    "font": ""
  },
  "attemptsDescriptionLabel": {
    "color": "#512da8ff",
    "size": 15,
    "font": ""
  },
  "hintLabel": {
    "color": "#000000ff",
    "size": 11,
    "font": ""
  },
  "operationInfoCell": {
    "cell": {
      "backgroundColor": "#ffffffff",
      "shadowStyle": {
        "shadowColor": "#000000ff",
        "shadowOffsetWidth": 0,
        "shadowOffsetHeight": 0,
        "shadowRadius": 0,
        "shadowOpacity": 0
      },
      "borderStyle": {
        "borderColor": "#000000ff",
        "borderWidth": 0
      }
    },
    "header": {
      "color": "#000000ff",
      "size": 20,
      "font": ""
    },
    "title": {
      "color": "#000000ff",
      "size": 17,
      "font": ""
    },
    "subTitle": {
      "color": "#000000ff",
      "size": 15,
      "font": ""
    }
  },
  "biometric": {
    "shadowStyle": {
      "shadowColor": "#000000ff",
      "shadowOffsetWidth": 0,
      "shadowOffsetHeight": 0,
      "shadowRadius": 0,
      "shadowOpacity": 0
    },
    "borderStyle": {
      "borderColor": "#000000ff",
      "borderWidth": 0
    },
    "description": {
      "color": "#000000FF",
      "size": 15,
      "font": ""
    },
    "errorDescription": {
      "color": "#bdbdbdFF",
      "size": 15,
      "font": ""
    },
    "cancelButtonTitle": {
      "color": "#000000ff",
      "size": 15,
      "font": ""
    },
    "fingerprintIcon": "",
    "fingerprintIconColor": "",
    "errorIcon": "",
    "errorIconColor": ""
  },
  "qrReader": {
    "description": {
      "color": "#000000FF",
      "size": 15,
      "font": ""
    },
    "error": {
      "color": "#000000FF",
      "size": 13,
      "font": ""
    },
    "galleryIcon": "",
    "imageTintColor": "#000000FF"
  }
},
"footer": {
  "backgroundColor": "#ffffffff",
  "shadowStyle": {
    "shadowColor": "#000000ff",
    "shadowOffsetWidth": 0,
    "shadowOffsetHeight": 0,
    "shadowRadius": 0,
    "shadowOpacity": 0
  },
  "borderStyle": {
    "borderColor": "#000000ff",
    "borderWidth": 0
  },
  "confirm": {
    "backgroundColor": "#4527a0ff",
    "shadowStyle": {
      "shadowColor": "#4527a000",
      "shadowOffsetWidth": 0,
      "shadowOffsetHeight": 0,
      "shadowRadius": 0,
      "shadowOpacity": 0
    },
    "borderStyle": {
      "borderColor": "#4527a0ff",
      "borderWidth": 0.3
    },
    "title": {
      "color": "#ffffffff",
      "size": 17,
      "font": ""
    }
  },
  "reject": {
    "backgroundColor": "#ffffffff",
    "shadowStyle": {
      "shadowColor": "#bdbdbd00",
      "shadowOffsetWidth": 0,
      "shadowOffsetHeight": 0,
      "shadowRadius": 0,
      "shadowOpacity": 0
    },
    "borderStyle": {
      "borderColor": "#dcdcdcff",
      "borderWidth": 0.3
```

```
    },
    "title": {
      "color": "#512da8ff",
      "size": 17,
      "font": ""
    }
  },
  "slider": {
    "minimumTrackTintColor": "#512da8ff",
    "maximumTrackTintColor": "#bdbdbdff",
    "thumbTintColor": "#512da8ff"
  },
  "pageCounter": {
    "color": "#000000ff",
    "size": 17,
    "font": ""
  }
 },
 "dialogString": {
   "passwordHeader": "Ввод пароля",
   "passwordHeaderWithConfirm": "Введите новый
пароль",
   "passwordTitle": "Пароль",
   "passwordPlaceholder": "Введите пароль",
   "passwordConfirmTitle": "Подтверждение
пароля",
   "passwordConfirmPlaceholder": "Подтвердите
пароль",
   "passwordContinue": "ПОДТВЕРДИТЬ",
   "passwordEasy": "Пароль должен быть не менее
6 символов",
   "passwordMedium": "Пароль должен быть не
менее 8 символов и содержать строчные и
прописные буквы",
   "passwordStrong": "Пароль должен быть не
менее 8 символов и содержать строчные и
прописные буквы, цифры",
   "passwordEmpty": "Поле не может быть
пустым",
   "passwordsNotMatch": "Пароли не совпадают",
   "incorrectPassword": "Неверный пароль",
   "passwordEasyHint": "Буквы a-z, A-Z, не
менее 6 символов",
   "passwordMediumyHint": "Буквы a-z, A-Z, не
менее 8 символов",
   "passwordStrongHint": "Цифры 0-9, буквы a-z,
A-Z, не менее 8 символов",
   "passwordTouchIDHint": "Введите пароль для
использования в случае ошибки TouchID",
   "numberOfAttempts": "Количество попыток: ",
   "numberOfAttemptsExceeded": "Вы превысили
число попыток. Попробуйте снова через минуту",
   "biometricAuthReason": "Аутентификация для
доступа к персональным данным",
   "biometricAuthReasonSave": "Сохранение
ключа",
   "biometricErrorTitle": "Ошибка",
   "biometricDescription": "Прикоснитесь к
сенсору",
   "biometricErrorDescription": "Не удалось
распознать отпечаток. Прикоснитесь еще раз",
   "incorrectQRCode": "Некорректные данные в QR
коде",
   "qrDescription": "Наведите камеру на QR-
код",
   "settingsTitle": "Доступ к камере отключен",
   "settingsSubtitle": "Перейти в Настройки для
разрешения доступа к камере?",
   "settingsCancel": "Отмена",
   "settingsSettings": "Настройки",
   "operationHeader": "Содержимое операции",
   "operationCancel": "ОТМЕНА",
   "operationConfirm": "ПОДТВЕРДИТЬ",
   "operationRefuse": "ОТКАЗАТЬСЯ",
   "printPdfHeader": "Версия для печати (pdf)",
   "rawPdfHeader": "Оригинал документа (pdf)",
   "profileHeader": "Профиль пользователя",
   "profileConfirm": "ПОДТВЕРДИТЬ",
   "confirmedDeviceInfo": "Добавляемое
устройство",
   "confirmedDeviceName": "Имя устройства",
   "confirmedDeviceConfirm": "ПОДТВЕРДИТЬ",
   "confirmedDeviceRefuse" : "ОТКАЗАТЬСЯ",
   "moreAboutOfOperation": "Подробнее об
операции",
   "modalViewDescription": "Вы уверены, что
хотите отправить запрос 'Отмена'?",
   "modalViewConfirm": "ПОДТВЕРДИТЬ",
   "modalViewRefuse": "ОТКАЗАТЬСЯ",
   "pdfPageNotLoaded": "Не удалось загрузить
страницу"
 }
}
```

Приложение 3. Сообщения об ошибках

Приложение содержит идентификаторы и тексты сообщений об ошибках, возвращаемых в методах DSS Client SDK.

Идентификатор	**Сообщение об ошибке**
invalid_input	Отсутствует тело запроса
invalid_identifier	Не передана информация о регистрируемом устройстве.
wrong_operation	Операция саморегистрации пользователей myDSS запрещена.
invalid_login	Логин не задан или имеет невалидный формат.
not_unique_login	Пользователь с данным логином уже существует.
invalid_phone	Полученное значение не является номером телефона.
not_unique_phone	Пользователь с таким номером телефона уже существует.
invalid_email	Полученное значение не является адресом электронной почты.
not_unique_email	Пользователь с таким адресом электронной почты уже существует.
invalid_device_fingerprint	Параметр deviceFingerprint не задан.
not_unique_device_fingerprint	Пользователь с таким отпечатком мобильного устройства уже существует.
invalid_device_params	Значение параметра osType не поддерживается или deviceName / pushAddress не задан.
key_already_confirmed	Информация об устройстве уже заполнена для ключа аутентификации.
invalid_key_id	Параметр [kid] не задан. или запрос на добавление устройства с идентификатором ключа [kid] не найден.
key_not_found	Пользователь не имеет назначенный токен аутентификации.
requested_user_not_found	Пользователь не найден.
invalid_certificate	Сертификат не найден или недействителен.
invalid_certificate_status	Статус сертификата не корректный.
certificate_not_found	Сертификат не найден.
existing_device_fingerprint	Такой отпечаток устройства уже зарегистрирован.
content_required	Не передано содержимое документа.

user_not_found	Не удалось найти пользователя по идентификатору набора ключей kid.
user_blocked	Учётная запись пользователя заблокирована.
invalid_authentication_scheme	Пользователю не назначен способ аутентификации через мобильное приложение.
key_expired_or_not_yet_valid	Срок действия ключей аутентификации истек или еще не наступил.
invalid_hmac	Неправильное значение HMAC.
invalid_license	Отсутствует действительная лицензия на способ аутентификации через мобильное приложение.
invalid_grant	Не удалось проверить заголовок аутентификации (подробности в журнале сервера).
internal_error	Внутренняя ошибка сервера.
BadRequest	Запрос некорректен.
invalid_kinit	Установка одноразового вектора аутентификации (kinit) уже подтверждена.
invalid_key_state	Данный ключ не активирован (invalid_key_state).