



PayControl

Описание интеграции в приложения Android

PayControl SDK. Версия 2.3

Лист истории изменений

Версия	Описание изменений
2.0	Первоначальная версия
2.1	<ul style="list-style-type: none">– Заменены библиотеки сканера штрих-кодов, уменьшен общий размер библиотек– Добавлена поддержка Android 6 Marshmallow– Добавлен метод <code>getFullTransactionInfo</code> для получения подробной информации из содержания QR-кода
2.2	<ul style="list-style-type: none">– Добавлена локализация сообщений на экране QR-сканера
2.3	<ul style="list-style-type: none">– Добавлена автоматическая синхронизация времени с сервером времени для работы TOTP

Оглавление

Интеграция SDK в проект	4
Добавление библиотек.....	4
Настройка зависимостей	4
Manifest.xml	5
Инициализация библиотек.....	5
PayControlApplication.....	5
Activity.....	6
Описание функций и методов.....	6
Локализация сообщений	6

Интеграция SDK в проект

Добавление библиотек

Для работы с SDK требуются следующие библиотеки:

- `ps_lib.jar` - работа с SDK
 - библиотеку необходимо поместить в каталог «`app/libs`»
- `libkey_gen.so` – нативные библиотеки SDK
 - библиотеки для различных платформ необходимо поместить в каталог «`app/src/main/jniLibs/`» с сохранением структуры каталогов

Настройка зависимостей

В правила сборки (`grandle`) в раздел «`dependencies`» необходимо добавить следующие строки:

- «`compile fileTree(include: ['*.jar', '*.so'], dir: 'libs')`» для включения в сборку нативных библиотек SDK;
- «`compile 'com.google.zxing:core:3.2.0'`» для включения в сборку сканера штрих-кодов `zxing`;
- «`compile 'cz.msebera.android:httpclient:4.3.5'`» для включения в сборку `http`-клиента, выполняющего автоматическую синхронизацию времени с серверами времени.

Пример файла `build.grandle`

```
apply plugin: 'com.android.application'
apply plugin: 'io.fabric'

android {
    compileSdkVersion 23
    buildToolsVersion "23.0.2"

    defaultConfig {
        applicationId = "com.safetech.paycontrol"
        minSdkVersion 11
        targetSdkVersion 23
        versionCode = 6
        versionName = "2.1"
    }
    buildTypes {
        release {
            minifyEnabled = false
            proguardFiles += file('proguard-rules.txt')
        }
    }
    productFlavors {
    }
    lintOptions {
        abortOnError false
    }
    allprojects {
        repositories {
            jcenter()
        }
    }
}

dependencies {
    compile fileTree(include: ['*.jar', '*.so'], dir: 'libs')
    compile 'com.android.support:recyclerview-v7:23.1.1'
    compile 'com.android.support:appcompat-v7:23.1.1'
    compile 'com.google.zxing:core:3.2.0'
    compile 'cz.msebera.android:httpclient:4.3.5'
}
```

Manifest.xml

1. Необходимо добавить разрешения для работы приложения с камерой мобильного телефона:

```
<uses-permission android:name="android.permission.CAMERA"/>
<uses-feature android:name="android.hardware.camera"/>
<uses-feature android:name="android.hardware.camera.autofocus"
    android:required="false"/>
```

2. Необходимо добавить Activity для работы с QR-сканером

```
<activity android:name="com.safetech.paycontrol.sdk.PCQRScannerActivity"/>
```

Инициализация библиотек

PayControlApplication

Наследование собственного Application от PayControlApplication

Простой способ работы с SDK – это создать Application и наследником от PayControlApplication.

```
public class MyApplication extends PayControlApplication {
    @Override
    public void onCreate() {
        super.onCreate();
    }
}
```

В Manifest.xml добавить следующую запись:

```
<application
    android:name=".MyApplication"
    ...
```

Самостоятельное добавление PayControl

Если Application уже наследуется от какого-либо Application, то необходимо добавить PayControl самостоятельно:

```
public class MyApplication extends Application {
    public PayControl payControl;
    @Override
    public void onCreate() {
        super.onCreate();
        payControl = new PayControl(this);
    }
    public PayControl getPayControl()
    {
        return payControl;
    }
    @Override
    public void onTerminate() {
        payControl.destroy();
        super.onTerminate();
    }
}
```

Activity

Быстрый доступ к методам

Для быстрого доступа к библиотеке необходимо добавить в Activity обращение к PayControl:

```
public PayControl getPayControl()
{
    return (MyApplication)getApplication().getPayControl();
}
```

Pause и Resume

Для корректной работы библиотеки необходимо добавить pause() и resume() в методы onPause() и onResume() для Activity:

```
@Override
protected void onPause() {
    super.onPause();
    getPayControl().pause();
}

@Override
protected void onResume() {
    super.onResume();
    getPayControl().resume();
}
```

Описание функций и методов

Описание функций и методов приведено в директории API Description. Начальная страница – index.html

Локализация сообщений

Для локализации сообщений на экране QR-сканера, необходимо передать локализованные строки в параметре localizationMsgs в методе

```
public void scanQR(String textToShow, String localizationMsgs, Activity activity)
```

Правила формирования строки localizationMsgs:

1. Каждый параметр состоит из ключа и значения, разделенными символом равенства «=»: key=value
2. Каждая пара «ключ=значение» разделена друг от друга переносом строки «\n»: «key1=value1\nkey2=value2»
3. В качестве ключа используется значение имени из перечисления com.safetech.paycontrol.sdk.PCQRScannerActivity.LocalizationMsgEnum
4. Если в строке localizationMsgs отсутствует какое-либо значение, то строка сообщения будет взята по умолчанию из состава библиотеки (язык – русский).
5. Если localizationMsgs является пустой строкой или «null», то будут использованы значения всех строк по умолчанию.

Например:

```
String localizationMsg =  
PCQRScannerActivity.LocalizationMsgEnum.MSG_CANCEL.name() + "=" + "Cancel" + "\n"  
+  
PCQRScannerActivity.LocalizationMsgEnum.MSG_OK.name() + "=" + "OK";
```

Автоматическая синхронизация времени

При инициализации библиотек PayControl выполняется попытка автоматического определения корректного времени через Интернет путем запроса к серверам времени компании Google.

При успешном получении значения времени вычисляется смещение полученного значения от времени, установленного в смартфоне Android. Данное смещение используется в дальнейшем для расчета кода подтверждения.

Полученное смещение запоминается в хранилище приложения и используется в дальнейшем в случае, если при следующей попытке автоматической синхронизации получить значение времени не удалось. Например, если смартфон находится в офлайне.

Функции автоматической синхронизации никак не затрагивают настройки времени смартфона.