Ezetap Android Cordova SDK

CORDOVA INTEGRATION DOCUMENT

1. Overview

This document provides information about how Ezetap customers can integrate with Ezetap's payment API's. Ezetap provides multiple ways to for customers to easily integrate their existing systems with the Ezetap payments platform. Integration can be achieved natively or on a Hybrid app using Cordova on the client application using Ezetap's Android SDK.

1.1 SDK and Documentation

Developer portal

Login to Ezetap's developer portal for detailed API description & usage.

SDK

You can download Ezetap's payment SDK and sample app from our Github page

2. API Integration

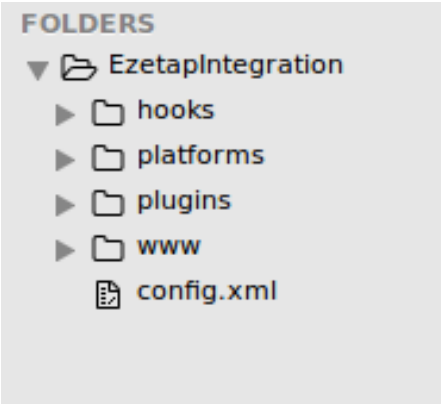
Assuming you are familiar on how Cordova works & the project structure of Cordova, this document will help you create an android plugin and use Ezetap API' s for making payments.

This integration involves two parts-

- Create Ezetap Cordova plugin for android
- API usage in your hybrid code.

2.1 Create Plugin

Your newly created or existing cordova project will look something like below, navigate to the plugins folder of the project & follow the below steps to create a plugin.

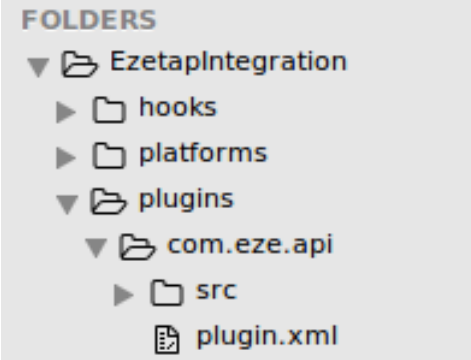


Step 1 -

Create a folder com.eze.api inside plugins folder of your existing or newly created app.

Step 2 -

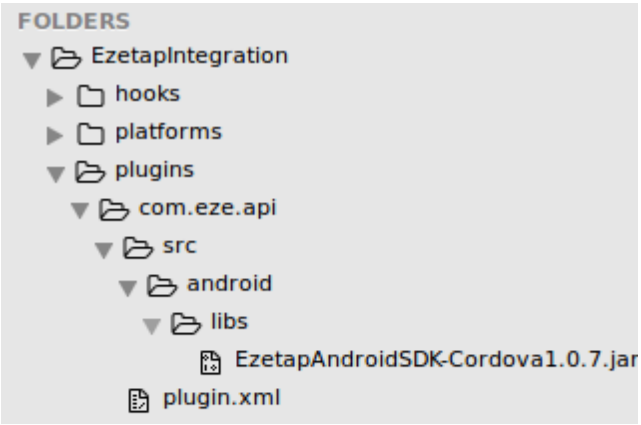
Create a new folder **src** and a file **plugin.xml** inside com.eze.api folder



Step 3 -

This plugin has a dependecy on Ezetap SDK (jar file) which is available in the releases folder of our Github page, clone the project make sure you are using the latest version of the SDK. Once our SDK is cloned follow the below steps-

- Create a new folder **android** inside src
- Create a new folder **libs** inside android
- Copy the latest **EzetapAndroidSDK-CordovaX.X.X.jar** from releases folder of our github repo & paste inside libs folder



Step 4 -

Creating a Provider to support service app installation on newer versions of Android.

- Create a new folder **xml** inside android folder.
- Create a file with name **provider_paths.xml** inside xml folder and paste the below code-

```
<?xml version="1.0" encoding="utf-8"?>
<paths xmlns:android="http://schemas.android.com/apk/res/android" >
        <files-path name="ezetap-download" />
        <external-path name="ezetap-download" />
</paths>
```

Step 5 -

Now that the dependency jar is setup inside the project, lets configure the plugin

Open plugin.xml which you added in **Step 2** and copy the below code in it, make sure you add your project name in the highlighted area & replace your application ID in the place of <Your application ID> highlighted in red. Your app ID/package name can be found in your app's manifest file under package tag or in the applicationId tag of your gradle file.

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin xmlns:android="http://schemas.android.com/apk/res/android"
xmlns="http://apache.org/cordova/ns/plugins/1.0" id="com.eze.api" version="1.0">
<name><<Your project name >></name>
<platform name="android">
        <source-file src="src/android/libs/<<Jar file copied in Step3>>.jar" target-dir="libs" />
        <source-file src="src/android/xml/provider_paths.xml" target-dir="res/xml" />
        <config-file target="res/xml/config.xml" parent="/*">
                <feature name="EzeAPIPlugin">
                        <param name="android-package" value="com.eze.api.EzeAPIPlugin"/>
                </feature>
        </config-file>
        <config-file target="AndroidManifest.xml" parent="/manifest">
                <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
        </config-file>
```

<config-file target="AndroidManifest.xml" parent="/manifest/application">

<activity android:name="com.eze.api.EzeAPIActivity"
android:screenOrientation="portrait"

android:configChanges="orientation|keyboardHidden|keyboard|screenSize|
locale"

android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen" />

</config-file>

<config-file target="AndroidManifest.xml" parent="/manifest/application">

<provider

android:name="com.ezetap.sdk.EzetapFileProvider"

android:authorities="<<Your app package >>.EzetapFileProvider"

android:exported="false"

android:grantUriPermissions="true" >

<meta-data

android:name="android.support.FILE_PROVIDER_PATHS"

android:resource="@xml/provider_paths" />

</provider>

</config-file>

</platform>

</plugin>

Step 6 -

Plugin succesfully added, you can use below cordova commands to use Ezetap SDK

- cordova plugin add cordova-plugin-android-support-v4-jar
- cordova platform add android
- cordova build android
- cordova run android

2.2 API Usage

After successful plugin creation, you can use Ezetap API's in your hybrid code.

Here is the sample-

```
var Request = {
    "demoAppKey": "",
    "prodAppKey": "",
    "merchantName": "",
    "userName":"",
    "currencyCode": "INR",
    "appMode": "DEMO",
    "captureSignature": "false",
    "prepareDevice" : "false"
};
 var ezeTapSuccessCallBack = function(response){
    console.log(JSON.stringify(response));
};
var ezeTapFailureCallBack = function(response){
        console.log(JSON.stringify(response));
};
cordova.exec(ezeTapSuccessCallBack,ezeTapFailureCallBack,"EzeAPIPlugin","initialize",
[Request]);
```

Note -

- For more API documentation check our developer's portal here
- To test the Ezetap plugin you built quickly you can copy paste the www folder of our sample project in your cordova project & run it