

# TouchSense® SDK for Mobile Apps

## Quick Start Guide

Tactile effects create more immersive and personal experiences for end users across a variety of mobile applications. The TouchSense® Software Development Kit for Mobile Apps (TS SDK) enables developers to bring those experiences to their Android applications. For example, TS SDK enables:

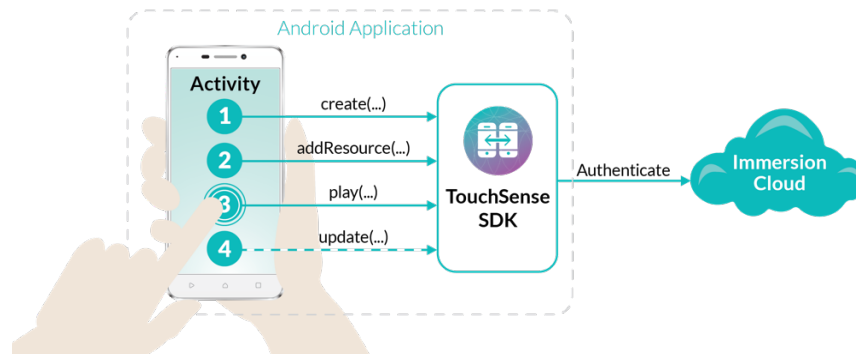
- Game developers to enhance their mobile gameplay with interactive tactile effects
- Multimedia applications to synchronize tactile effects with media content (e.g. video)

This document provides basic, high-level instructions for integrating TS SDK in an Android application. For more detailed steps on using the SDK in your Java, native, Unity, or Cocos2d-x environments, please refer to the main user guide (TouchSenseSDK\_User\_Guide.pdf).

In order to setup your (Java) environment,

- Place **TouchSenseSDK-xyz.jar** and **libTouchSenseSDK.so** files in your project's **libs** and **libs/armeabi** directories respectively. If the **libs** and **libs/armeabi** directories don't exist in your project, you will need to create them.
- Add **VIBRATE** and **INTERNET** permissions to **AndroidManifest.xml** file.
- Add these instructions into the app's **build.gradle** file:

```
android.sourceSets.main.jniLibs.srcDirs = [ 'libs' ]
dependencies {compile fileTree (include: [ 'TouchSenseSDK*.jar' ], dir: 'libs')}
```



- 1) create (...)**  
Retrieve an instance of the SDK by supplying your Immersion-provided credentials to the API. The credentials are used to authenticate your license with Immersion's cloud, and if successful, a valid instance of the SDK will be returned. See section [3.1 Initializing the Haptic Media Player](#) of the main user guide for more details.
- 2) addResource (...)**  
Using a valid instance of the SDK, load haptic media files (**.hapt** files) from the supplied haptic effect library by specifying each effect's URI. If successful, the API will return a unique resource ID for each URI supplied, which is used to identify each individual **.hapt** file. See section [3.2 Adding Haptic Resources](#) of the main user guide for more details.
- 3) play (...)**  
Play a haptic resource by specifying its corresponding resource ID from Step 2. This API will return a unique effect ID, which can be used to perform additional playback operations, such as stop, seek, mute, and more. See section [3.3 Playing Haptic Effects](#) of the main user guide for more details.
- 4) update (...)**  
Optionally synchronize the playback of the haptic effect with a time source by supplying a time stamp to this API every 1 second. See section [3.5 Synchronization](#) of the main user guide for more details.

For more information on how to execute the above steps or use the other available APIs to control haptic feedback in your native, Unity, or Cocos2d-x applications, please consult the respective sections in the main user guide.

For further assistance, please ask your Immersion sales representative or send an email to [content@immersion.com](mailto:content@immersion.com).