

# element-android-examples

---

*Element SDK Library* is a standalone Android application that enables the creation of biometrics models that can be used to authenticate users. *Element Orbit* is the Android library providing interfaces to communicate with *Element SDK Library*. This document contains information to integrate *Element Orbit* into an Android application by using Android Studio.

## Setup with Android Studio

### Import *Element Orbit*

1. Download the latest [element-orbit-release.aar](#) file.
2. Open your project in Android Studio.
3. On the top menu bar, click *File->New->New Module->Import .aar/.jar libraries->Next*.
4. In the next window, choose the path to *element-orbit.aar* in the *File Name* field, and type *element-orbit* in the *Subproject name* field.
5. Click the *Finish* button and wait for Android Studio to finish building the project.

### Refer to *Element Orbit* as a project dependency

1. On the top menu bar, click *File->Project Structure*.
2. Select your app module under *Modules* on the left pane, switch to the *Dependencies* tab, and click on the + button at the bottom of the window.
3. Choose the *Module dependency* option in the popup, and select *element-orbit*.

## Integrating *Element Orbit* into your application

- In *AndroidManifest.xml*, declare your api key in a meta-data tag and declare *OrbitReceiver* inside the *application* node.

```
<manifest>
    .....
    <application>
        .....
        <meta-data
            android:name="com.element.ApiKey"
            android:value="[YOUR_API_KEY]" />
        .....
        <receiver
```

```

        android:name="com.element.orbit.OrbitReceiver"
        android:enabled="true"
        android:exported="true">
        <intent-filter>
            <action android:name="com.element.orbit.ACTION_RESPONSE" />
        </intent-filter>
    </receiver>
    .....
</application>
</manifest>

```

- In the app level build.gradle, add the gson library as a dependency.

```

dependencies {
    .....
    compile 'com.google.code.gson:gson:2.2.4'
}

```

## Implementation of the key classes

### ***RequestManager***

*RequestManager* defines a set of handy functions to send out requests to *Element SDK Library*. After sending out requests via *RequestManager*, *OrbitListener* will get called as soon as hearing back from *Element SDK Library*.

- `syncUserData()`: request to fully sync user data from the SDK Library
- `addNewUser()`: request to add a new user
- `trainModel()`: request to start training all user models
- `authenticateUser()`: requests to authenticate the user

### ***OrbitService & OrbitListener***

*OrbitService* is the main gateway to the communicate with *Element SDK Library*. Initialize *OrbitService* when an Activity starts with a proper *OrbitListener* to receive data and information receiving back from *Element SDK Library*.

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    ...

    OrbitService.registerCallback(this);
}

```

- `onAllUserData()`: get called when all user data is received
- `onUpdatedUserData()`: get called when user data is updated
- `onAuthenticated()`: get called when the authentication result is received
- `onMessage()`: get called when any message is returned

## **Data Classes**

Data classes are the informative packets that are delivered between *Element SDK Library* & *Element Orbit*.

- *Config*: contains the parameters, status, and settings of *Element SDK Library*
- *Authentication*: contains the authentication information, status and results
- *UserData*: contains the user information including `userId`, names, etc.

## **Update Config**

*Element Orbit* can be configured by passing a *Config* data class. To get Config Update:

1. `RequestManager.updateConfig()`: register a Config
2. `onUpdateConfig()`: receive config updates

## **Notes**

- *Element SDK Library* is required to be installed on an Android device. It is available on [Google Play](#).
- *Element SDK Library* is currently in beta release. Please contact Element for access.