element

element-android-examples

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

Prerequisites

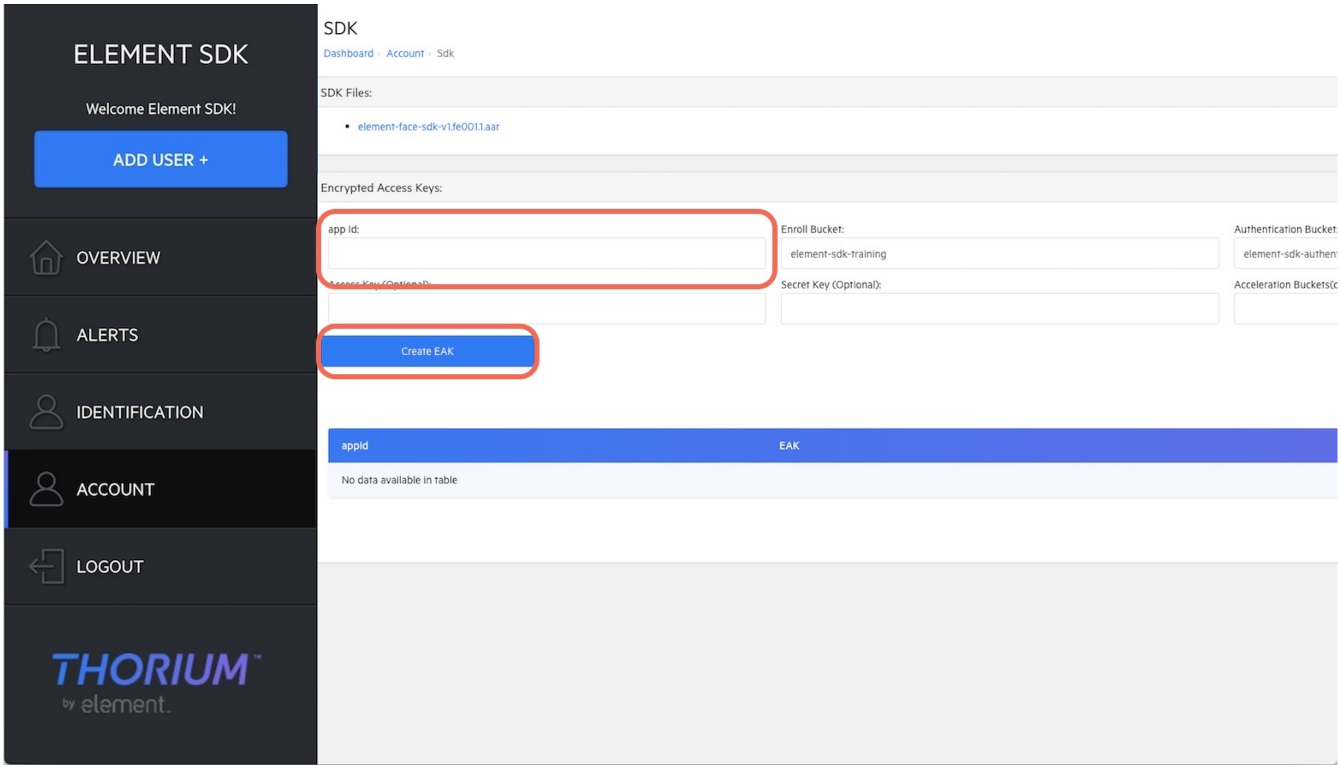
Element Dashboard

The Element Dashboard is here. An account is required to access the Element Dashboard.

Register the Application Id (App Id) to Element and obtain the Encrypted Access Key (EAK)

The Element SDK requires the Encrypted Access Key (EAK) file. The EAK carries encrypted information including the Application Id (App Id). The App Id in the EAK will need to match the id of the running application on an Android device. The EAK is also available on Element Dashboard, under Account -> SDK.

- Fill the App Id field with your application id, and click on Create EAK.
- Download the EAK file.



SDK

Dashboard › Account › Sdk

SDK Files:

- element-face-sdk-v1.fe001.1.aar

Encrypted Access Keys:

app Id:

Enroll Bucket: element-sdk-training

Authentication Bucket: element-sdk-authen

Access Key (Optional):

Secret Key (Optional):

Acceleration Buckets(c

appId	EAK
No data available in table	

Setup with Android Studio

Import *Orbit Lite*

1. Download the latest *orbitlite-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 _ orbitlite-release.aar_ in the *File Name* field, and type in *orbitlite* in the *Subproject name* field.
5. Click the *Finish* button and wait for Android Studio to finish building the project.

Refer to *Orbit Lite* 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 *Orbit Lite* into your application

- In AndroidManifest.xml, declare your EAK in a meta-data tag.

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

Implementation of the key classes

This is our *JavaDoc*

ElementSDKManager

ElementSDKManager defines a set of handy functions to send out requests to *Element SDK Library*.

- initElementSDK(Context context): request to initialize the Element SDK Library. If this is not called before the first SDK request, an "Unauthorized" error is presented. At a minimum, this method should be called in *onCreate()* of your application class, or when the app comes to the foreground.
- enrollNewUser(Activity activity, String userName, HashMap<String,String> extras): request to add a new user
- identifyUser(Activity activity, ArrayList elementUserIds): look through the list of user ids specified to find a user via their palm
- authenticateUser(Activity activity, String elementUserId): authenticate the specified user via their palm
- requestSyncState(Activity activity, ArrayList elementUserIds, final SyncStateListener listener): request the sync information for the users in the list. Pass an empty list to get sync state for all users
- requestServerSync(Context context): ask the SDK to refresh it's data from the server
- cursorToSyncState(Cursor cursor): convert a LoaderCallbacks Cursor to ElementSyncState and make the data accessible

Listeners

Used to receive results from *Element SDK Library*. *EnrollListener* is used for Enroll. *SearchListener* is used for Search. *AuthListener* is used for Authentication. *SyncStateListener* is used to receive SyncState results.

ElementSDKManager.getCursorLoader is available to listen to the Element ContentProvider using LoaderManager.LoaderCallbacks. This is preferable to requestSyncState() which only polls.

Call *ElementSDKManager.onActivityResult()* with an instance of one or all of these listeners to receive the appropriate callbacks after making a request.

```
public class MainActivity extends AppCompatActivity implements ElementSDKManager

...
    @Override
        protected void onActivityResult(int requestCode, int resultCode, Intent
            super.onActivityResult(requestCode, resultCode, data);
            ElementSDKManager.onActivityResult(requestCode, resultCode, data, th
        }
...
}
```

ElementSyncState

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

- *ElementSyncState*: contains the model information and status of each user in the *Element SDK Library*

Notes

- *Element SDK Library* is required to be installed on an Android device. It is available on HockeyApp.
- *Element SDK Library* is currently in beta release. Please contact Element for access.