Element Face SDK

The Element Face SDK is an API library to create the biometrics models that can be used to authenticate users. This document contains information to integrate the Element Face SDK into an Android application by using Android Studio.

Version Support

- The Element Face SDK supports Android 5.0+ / API 21+ (Lollipop and up)
- Android Studio 3.1.3 with Gradle 4.4
- Last Target SDK Version: 27
- Last Compiled Support Library Version: 27.1.0
- Last Compiled Google Play Services Version: 15.0.1
- AWS Mobile SDK: 2.6.+
- Google Guava for Android: 23.5

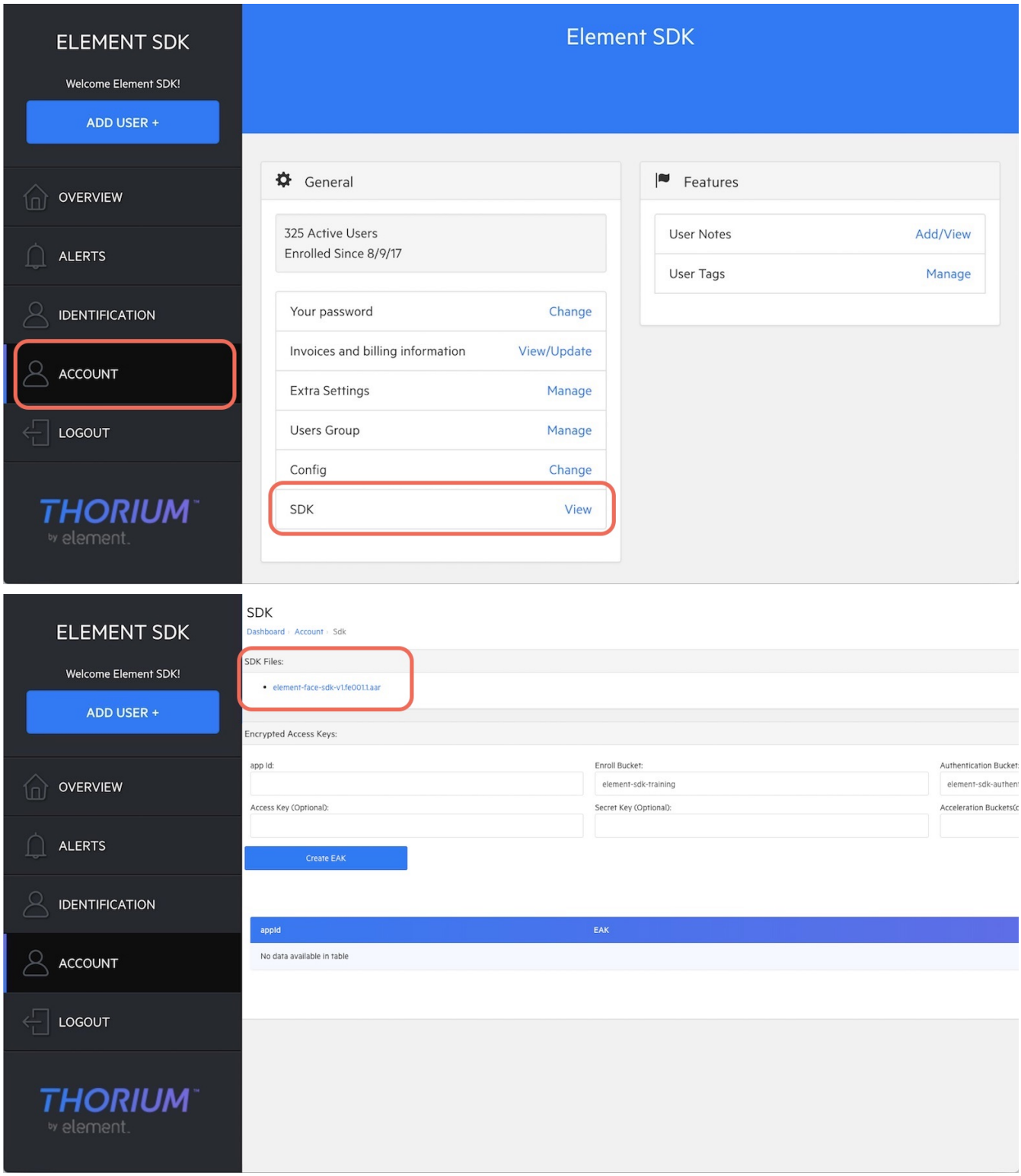
Prerequisites

Element Dashboard

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

AAR

The Element Face SDK is in the AAR format. The AAR file is available on Element Dashboard, under Account -> SDK -> SDK Files.

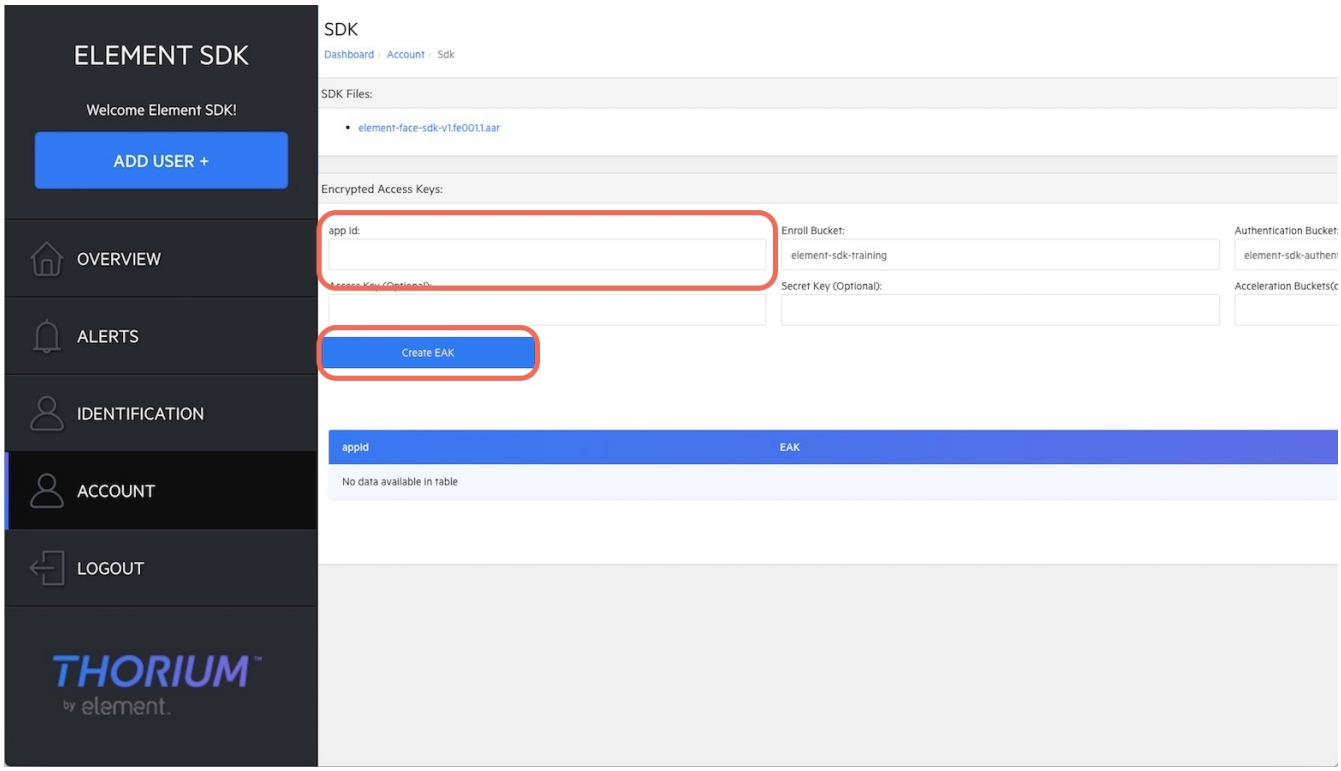


Register the Application Id (App Id) to Element and obtain the Encrypted Access Key (EAK)

The Element Face 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.



Setup with Android Studio

Import the AAR

1. Open your project in Android Studio.
2. On the top menu bar, click File->New->New Module->Import .aar/.jar libraries->Next.
3. In the next window, choose the path to the AAR. And in the File Name field, and type in element-face-sdk in the Subproject Name field.
4. Click the Finish button and wait for Android Studio to finish building the project.

Add element-face-sdk and dependencies

1. Add element-face-sdk to your project in settings.gradle

```
include ':element-face-sdk'
```

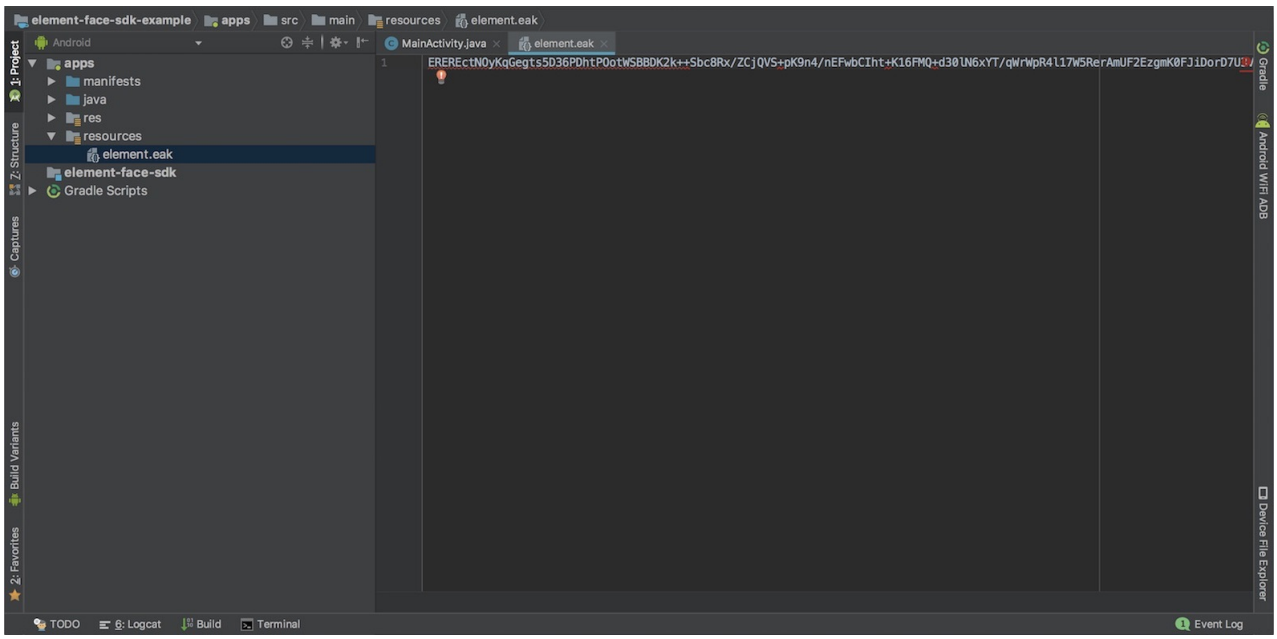
2. On the top menu bar, click File->Project Structure.

3. Select your app module under Modules on the left pane, click on the Dependencies tab, and click on the + button at the bottom of the window.
4. Choose the Module Dependency option in the popup, and select element-face-sdk.
5. Add the following dependencies to build.gradle in the app level.

```
dependencies {
  .....
  implementation 'com.android.support:appcompat-v7:27.1.0'
  implementation 'com.amazonaws:aws-android-sdk-core:2.6.+'
  implementation 'com.amazonaws:aws-android-sdk-s3:2.6.+'
  implementation 'com.google.android.gms:play-services-location:+'
  implementation 'com.google.guava:guava:23.5-android'
}
```

Include the EAK in the application

1. Create a resources directory, for example, [project dir]/[app dir]/[src]/[main]/resources.
2. Copy the element.eak file into the resources directory.



Using the Element Face SDK APIs

Initialize the Element Face SDK

1. Create a class extends android.app.Application, and initialize the Element Face SDK in onCreate

```
public class MainApplication extends Application {
  @Override
```

```
  public void onCreate() {
    super.onCreate();
    ElementFaceSDK.initSDK(MainApplication.this);
  }
}
```

2. Declare the Application in AndroidManifest.xml

```
<manifest>
  .....
  <application android:name=".MainApplication">
    .....
  </application>
</manifest>
```

Ask for user permissions

1. The Element Face SDK requires the permissions,
2. android.Manifest.permission.CAMERA
3. android.Manifest.permission.ACCESS_FINE_LOCATION
4. android.Manifest.permission.ACCESS_COARSE_LOCATION
5. The Element Face SDK provides `PermissionUtils.verifyPermissions(Activity activity, String... permissionsToVerified)` for requesting the permissions.

```
PermissionUtils.verifyPermissions(
  MainActivity.this,
  Manifest.permission.CAMERA,
  Manifest.permission.ACCESS_FINE_LOCATION,
  Manifest.permission.ACCESS_COARSE_LOCATION);
```

User enrollment

The Element Face SDK utilizes the ElementFaceEnrollActivity for user enrollment. It's based on the startActivityForResult) method.

1. Enroll an user and obtain the UserInfo. The UserInfo contains an unique userId (ElementId). The pair of the userId and the appId (context.getPackageName()) are mainly used in the Element Face SDK to inquiry the user information and status.

```
UserInfo userInfo = UserInfo.enrollUser(
  mainActivity.getBaseContext(),
  mainActivity.getPackageName(),
  fStr,
  lStr,
  extra);
```

2. Declare the request code

```
public static final int ENROLL_REQ_CODE = 12800;
```

3. Start the ElementFaceEnrollActivity

```
Intent intent = new Intent(MainActivity.this, ElementFaceEnrollActivity.cla
intent.putExtra(ElementFaceEnrollActivity.EXTRA_ELEMENT_USER_ID, userInfo.u
startActivityForResult(intent, ENROLL_REQ_CODE);
```

4. Override the onActivityResult) method to receive the enrollment results

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent dat
  if (requestCode == ENROLL_REQ_CODE) {
    if (resultCode == Activity.RESULT_OK) {
      showMessage(getString(R.string.enroll_completed));
    } else {
      showMessage(getString(R.string.enroll_cancelled));
    }
  }
}
```

User authentication

User authentication is similar to user enrollment, but it calls the ElementFaceAuthActivity.

1. Declare the request code

```
public static final int AUTH_REQ_CODE = 12801;
```

2. Start the ElementFaceAuthActivity

```
Intent intent = new Intent(MainActivity.this, ElementFaceAuthActivity.class
intent.putExtra(ElementFaceAuthActivity.EXTRA_ELEMENT_USER_ID, userInfo.use
startActivityForResult(intent, AUTH_REQ_CODE);
```

3. Override the onActivityResult) method to receive the authentication results

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent dat
  if (requestCode == AUTH_REQ_CODE) {
    if (resultCode == Activity.RESULT_OK) {
      String results = data.getStringExtra(EXTRA_AUTH_RESULTS);
      if (ElementFaceAuthActivity.USER_VERIFIED.equals(results)) {
          showMessage(getString(R.string.msg_verified));
      } else if (ElementFaceAuthActivity.USER_FAKE.equals(results)) {
          showMessage(getString(R.string.msg_fake));
      } else {
```

```
            showMessage(getString(R.string.msg_not_verified));
          }
        } else {
          showMessage(getString(R.string.auth_cancelled));
        }
      }
    }
```

Questions?

If you have questions, please contact devsupport@discoverelement.com.