PRACTICAL-11

AIM: Creating an application that provides Single Sign-on (SSO) with Chrome CustomTabs via theAppAuth library, and optionally push managed configuration toprovide a user loginhint.https://codelabs.developers.google.com/codelabs/signin/index.html?index=..%2F..index#0

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayoutxmlns:android="http://schemas.android.com/
apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.example.app_1.signin_1.MainActivity">

<LinearLayout
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:orientation="vertical"
android:padding="2dip">
<com.google.android.gms.common.SignInButton
android:id="@+id/sign_in_button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:enabled="false" />

<Button
android:id="@+id/sign_out_button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Sign Out"
android:enabled="true" />

<Button
android:id="@+id/revoke_access_button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Revoke Access"
android:enabled="true" />

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/statuslabel"
android:text="Status"/>
</LinearLayout>

</android.support.constraint.ConstraintLayout>
```

mainactivity.java

```
package com.example.app_1.signin_1;

import android.app.PendingIntent;
import android.content.Intent;
```

```
import android.content.IntentSender;
import android.support.v4.app.FragmentActivity;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.SignInButton;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.common.api.GoogleApiClient.ConnectionCallbacks;
importcom.google.android.gms.common.api.GoogleApiClient.OnConnectionFailedListener;
importcom.google.android.gms.common.api.Scope;
import com.google.android.gms.plus.Plus;

public class MainActivityextends FragmentActivityimplements
ConnectionCallbacks, OnConnectionFailedListener,
View.OnClickListener {
private static final int SIGNED_IN = 0;
private static final int STATE_SIGNING_IN = 1;
private static final int STATE_IN_PROGRESS = 2;
private static final int RC_SIGN_IN = 0;
private GoogleApiClientmGoogleApiClient;
private int mSignInProgress;
private PendingIntentmSignInIntent;
private SignInButtonmSignInButton;
private Button mSignOutButton;
private Button mRevokeButton;
private TextViewmStatus;

    @Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);


// Get references to all of the UI views
mSignInButton= (SignInButton) findViewById(R.id.sign_in_button);
mSignOutButton= (Button) findViewById(R.id.sign_out_button);
mRevokeButton= (Button) findViewById(R.id.revoke_access_button);
mStatus= (TextView) findViewById(R.id.statuslabel);

// Add click listeners for the buttons
mSignInButton.setOnClickListener(this);
mSignOutButton.setOnClickListener(this);
mRevokeButton.setOnClickListener(this);

// Build a GoogleApiClient
mGoogleApiClient= buildGoogleApiClient();
    }
private GoogleApiClientbuildGoogleApiClient() {
return new GoogleApiClient.Builder(this)
                .addConnectionCallbacks(this)
                .addOnConnectionFailedListener(this)
                .addApi(Plus.API, Plus.PlusOptions.builder().build())
                .addScope(new Scope ("email"))
                .build();
    }
    @Override
protected void onStart() {
super.onStart();
mGoogleApiClient.connect();
```

```
    }

    @Override
protected void onStop() {
super.onStop();
mGoogleApiClient.disconnect();
    }
    @Override
public void onConnectionSuspended(int cause) {
mGoogleApiClient.connect();
    }
    @Override
public void onConnected(Bundle connectionHint) {
mSignInButton.setEnabled(false);
mSignOutButton.setEnabled(true);
mRevokeButton.setEnabled(true);

// Indicate that the sign in process is complete.
mSignInProgress= STATE_SIGNED_IN;

try {
            String emailAddress = Plus.AccountApi.getAccountName(mGoogleApiClient);
mStatus.setText(String.format("Signed In to My App as %s", emailAddress));
        }
catch(Exception ex){
            String exception = ex.getLocalizedMessage();
            String exceptionString = ex.toString();
// Note that you should log these errors in a 'real' app to aid in debugging
}
    }
    @Override
public void onConnectionFailed(ConnectionResult result) {
if (mSignInProgress!= STATE_IN_PROGRESS) {
mSignInIntent= result.getResolution();
if (mSignInProgress== STATE_SIGNING_IN) {
resolveSignInError();
            }
        }
// Will implement shortly
onSignedOut();
    }

private void resolveSignInError() {
if (mSignInIntent!= null) {
try {
mSignInProgress= STATE_IN_PROGRESS;
startIntentSenderForResult(mSignInIntent.getIntentSender(),
RC_SIGN_IN, null, 0, 0, 0);
            } catch (IntentSender.SendIntentException e) {
mSignInProgress= STATE_SIGNING_IN;
mGoogleApiClient.connect();
            }
        } else {
// You have a play services error -- inform the user
}
    }
    @Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
switch (requestCode) {
case RC_SIGN_IN:
if (resultCode == RESULT_OK) {
mSignInProgress= STATE_SIGNING_IN;
                } else {
```

```
mSignInProgress= SIGNED_IN;
                }

if (!mGoogleApiClient.isConnecting()) {
mGoogleApiClient.connect();
                }
break;
        }
    }
private void onSignedOut() {
// Update the UI to reflect that the user is signed out.
mSignInButton.setEnabled(true);
mSignOutButton.setEnabled(false);
mRevokeButton.setEnabled(false);

mStatus.setText("Signed out");
    } @Override
public void onClick(View v) {
if (!mGoogleApiClient.isConnecting()) {
switch (v.getId()) {
case R.id.sign_in_button:
mStatus.setText("Signing In");
resolveSignInError();
break;
case R.id.sign_out_button:
Plus.AccountApi.clearDefaultAccount(mGoogleApiClient);
mGoogleApiClient.disconnect();
mGoogleApiClient.connect();
break;
case R.id.revoke_access_button:
Plus.AccountApi.clearDefaultAccount(mGoogleApiClient);
Plus.AccountApi.revokeAccessAndDisconnect(mGoogleApiClient);
mGoogleApiClient= buildGoogleApiClient();
mGoogleApiClient.connect();
break;
            }
        }
    }
}
```

appgradle

```
apply plugin: 'com.android.application'

android {
compileSdkVersion 26
defaultConfig {
applicationId"com.example.app_1.signin_1"
minSdkVersion 22
targetSdkVersion 26
versionCode 1
versionName"1.0"
testInstrumentationRunner"android.support.test.runner.AndroidJUnitRunner"
}
buildTypes {
        release {
minifyEnabledfalse
proguardFilesgetDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
}
    }
}
```

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:26.1.0'
compile 'com.google.android.gms:play-services:7.3.0'
implementation 'com.android.support.constraint:constraint-layout:1.1.3'
testImplementation'junit:junit:4.12'
androidTestImplementation'com.android.support.test:runner:1.0.2'
androidTestImplementation'com.android.support.test.espresso:espresso-core:3.0.2'
}
```

OUTPUT:

Conclusion:

We successfully completed added google login