****

**Turkcell Login SDK**

**Android Integration Document**

Author: Semih Gökyer

Document Status : Draft

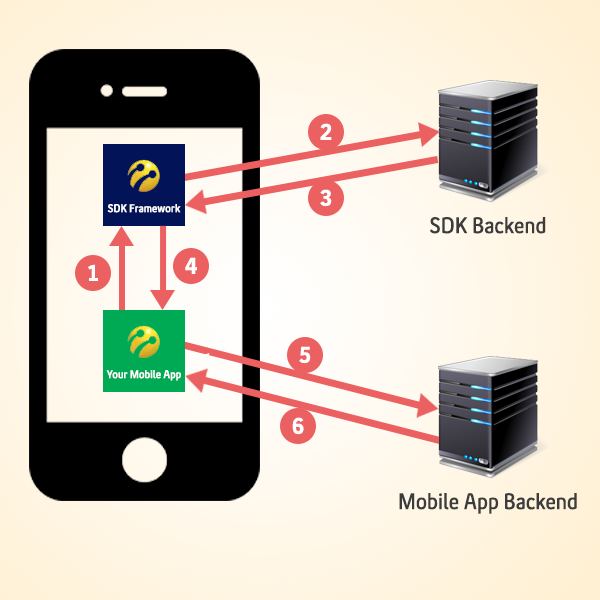
Document Version : 0.2

1. Prerequirements

Minimum deployment target for Login SDK is Anroid 10(GINGERBREAD)

If your app supports lower versions than 10 you can’t integrate your app.

1. Flow



1. Your app start LoginSDKMainActivity to start login process with following mandatory extras:

Intent intent = new Intent(this, com.turkcell.loginsdk.activity.LoginSDKMainActivity.class);

intent.putExtra(LoginSdk.KEY\_APP\_ID,"123");//Application unique id

intent.putExtra(LoginSdk.KEY\_DLL, "true");//If dll is true,Disable light login(skip 3g login)

intent.putExtra(LoginSdk.KEY\_SHOW\_LOGIN\_PAGE, "true");// If true,first trying rememberme login or lightLogin then show login page

intent.putExtra(LoginSdk.KEY\_IS\_DIRECT\_LOGIN, "false");//If true,automatically login page is shown

1. SDK makes rest call to SDK Server\*
2. SDK Server returns login result to SDK Client\*
3. SDK returns login result and authtoken (if result is success) to onActivityResult:

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

String resultMessage = data.getStringExtra("authToken") == null ? data.getStringExtra("code") + "\_" + data.getStringExtra("message") : data.getStringExtra("authToken");

if (100 == requestCode) {

if (resultCode == Activity.RESULT\_OK) {

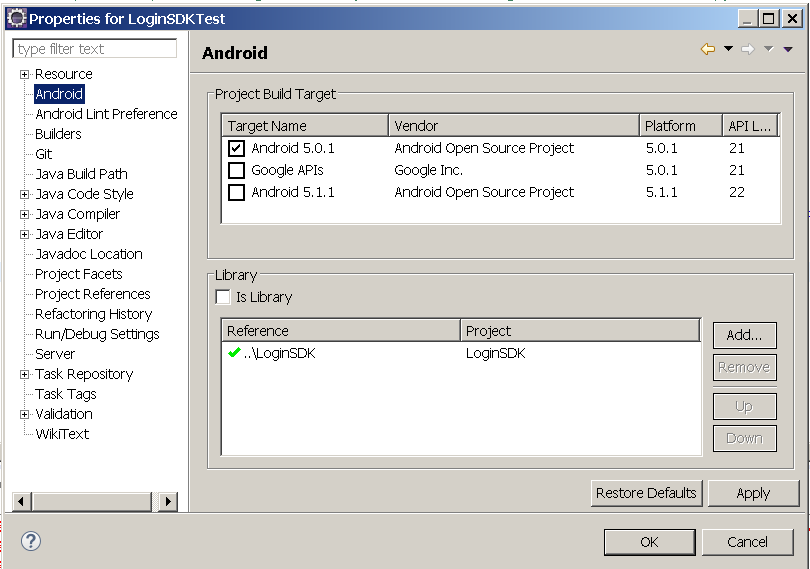
Toast.makeText(this, resultMessage, Toast.LENGTH\_LONG).show();

}

}

\*Added just for information. App developer shouldn’t care about these steps.

1. App sends token its backend to validate it
2. App backend calls validateAuthToken to retrieve msisdn, then starts session and performs app related tasks.
3. Integration
4. Integration for library project

Last version of LibraryProject didn’t support Turkcell All-Access Login Mechanism.If you want to integrate all-access login,You must convert your project to Android Studio gradle project.If not,You need to check out LibraryProject in github and import project(LoginSDK) into workspace.if you want to try example project,import also LoginSDKTest project and add library project to main project.

1. Integration for android studio

You need to check out AndroidStudioProject in github.There is an aar file in this example project.

AndroidStudioProject/app/libs/app-debug.aar

Add this aar file into libs folder that is in main project and add these following dependencies into build.gradle file.

compile(name: 'app-debug', ext: 'aar')

compile 'com.mcxiaoke.volley:library:1.0.17@aar'

compile 'com.android.support:support-v4:23.0.0'

compile 'com.android.support:appcompat-v7:23.0.0'

compile 'com.android.support:design:23.0.0'

compile 'com.google.code.gson:gson:2.4'

1. Configuration for your project

You need add following permission to manifest.

<**uses-permission android:name="android.permission.INTERNET"** />  
<**uses-permission android:name="android.permission.READ\_PHONE\_STATE"** />  
<**uses-permission android:name="android.permission.GET\_ACCOUNTS"** />

Then,you need add following declaration to manifest(into application tag).

<activity android:name="com.turkcell.loginsdk.activity.LoginSDKMainActivity"></activity>

1. Configuration for proguard

LoginSDK use some library.You need to add these to your proguard to block crashing your release application.

-dontwarn android.support.\*\*

-dontwarn android.support.v7.\*\*

-keep class com.turkcell.loginsdk.service.response.\*\* {\*;}

-keep class com.google.gson.\*\* { \*; }

-dontwarn android.\*\*

-dontwarn com.android.\*\*

-dontwarn com.google.android.\*\*

-keep class android.support.v4.app.\*\* { \*; }

-keep interface android.support.v4.app.\*\* { \*; }

-keep class android.support.v7.app.\*\* { \*; }

-keep interface android.support.v7.app.\*\* { \*; }

-keep class com.android.volley.\*\* { \*; }

-keep interface com.android.volley.\*\* { \*; }

-dontwarn com.android.volley.toolbox.\*\*

-keepattributes Signature

-keepattributes \*Annotation\*

-keep class sun.misc.Unsafe { \*; }

-keep class com.google.gson.stream.\*\* { \*; }

-keep class com.google.gson.FieldNamingStrategy { \*; }

SDK Functionality

There is only one Activity in LoginSDK(LoginSDKMainActivity) but 2 functionality in this activity.(Login,Logout)

intent.putExtra("dll", "false");//If dll is true,Disable light login(skip 3g login)

intent.putExtra("loginForce", "false");//if loginForce is true,client try 3g login.if it fails, open wifi login screen

* 1. Login

Calling login intent

Intent intent = new Intent(this, com.turkcell.loginsdk.activity.LoginSDKMainActivity.class);

intent.putExtra( LoginSdk.KEY\_ENV, "PROD");//Default environment is test

intent.putExtra(LoginSdk.KEY\_APP\_ID,"123");//Application unique id

intent.putExtra(LoginSdk.KEY\_DLL, "true");//If dll is true,Disable light login(skip 3g login)

intent.putExtra(LoginSdk.KEY\_SHOW\_LOGIN\_PAGE, "true");// If true,first trying rememberme login or lightLogin then show login page

intent.putExtra(LoginSdk.KEY\_IS\_DIRECT\_LOGIN, "false");//If true,automatically login page is shown

startActivityForResult(intent, 100);

Result login intent

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

String resultMessage = data.getStringExtra("authToken") == null ? data.getStringExtra("code") + "\_" + data.getStringExtra("message") : data.getStringExtra("authToken");

if (100 == requestCode) {

if (resultCode == Activity.RESULT\_OK) {

Toast.makeText(this, resultMessage, Toast.LENGTH\_LONG).show();

}

}

* 1. Logout

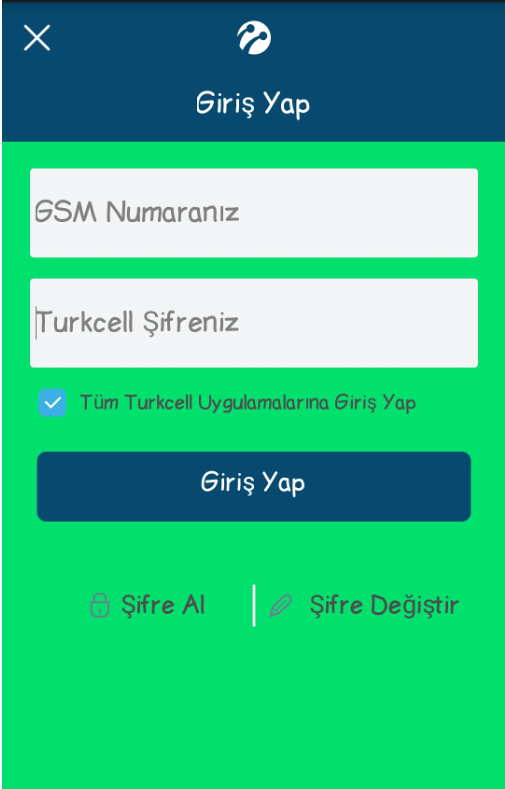
Calling logout

LoginSdk.logOut(“123”, “PROD”, MainActivity.this, new LoginSdkCallBack<LogoutResponse>() {  
 @Override  
 public void onSuccess(LogoutResponse value) {  
 Toast.makeText(MainActivity.this, "logout Success", Toast.LENGTH\_SHORT).show();  
 }  
  
 @Override  
 public void onFail(Exception exception) {  
 Toast.makeText(MainActivity.this, "logout failed", Toast.LENGTH\_SHORT).show();  
  
 }  
});

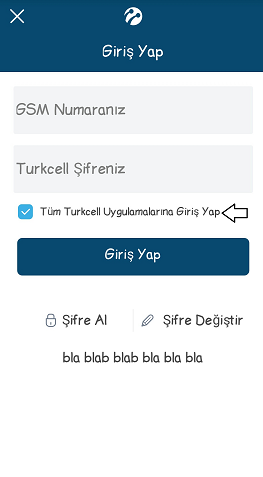
1. Customizing Theme

intent.putExtra(LoginSdk.***KEY\_FONT\_PATH***, **"fonts/LDFComicSansBold.ttf"**);

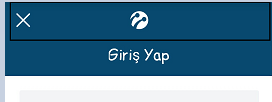
intent.putExtra(LoginSdk.***KEY\_PAGE\_BACKGROUND\_COLOR***, R.color.***white***);



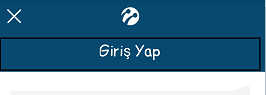
intent.putExtra(LoginSdk.***KEY\_PAGE\_TEXT\_COLOR***, R.color.***lsdk\_gray1***);



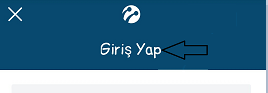
intent.putExtra(LoginSdk.***KEY\_HEADER\_BACKGROUND\_COLOR***, R.color.***lsdk\_defaultHeaderColor***);



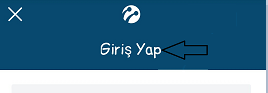
intent.putExtra(LoginSdk.***KEY\_HEADER\_TEXT\_BACKGROUND\_COLOR***, R.color.***lsdk\_defaultHeaderColor***);



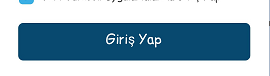
intent.putExtra(LoginSdk.***KEY\_HEADER\_TEXT\_COLOR***, R.color.***white***);



intent.putExtra(LoginSdk.***KEY\_HEADER\_TITLE***, **"Turkcell Platinum"**);



intent.putExtra(LoginSdk.***KEY\_POSITIVE\_BUTTON\_COLOR***, R.color.***lsdk\_defaultSignInButtonColor***);



intent.putExtra(LoginSdk.***KEY\_POSITIVE\_BUTTON\_TEXT\_COLOR***, R.color.***c\_363E4F***);



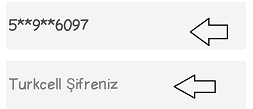
intent.putExtra(LoginSdk.***KEY\_NEGATICE\_BUTTON\_COLOR***, R.color.***c\_f4f5f8***);



intent.putExtra(LoginSdk.***KEY\_NEGATICE\_BUTTON\_TEXT\_COLOR***, R.color.***c\_363E4F***);



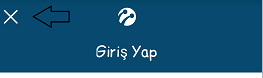
intent.putExtra(LoginSdk.***KEY\_INPUT\_BACKGROUND\_COLOR***, R.color.***lsdk\_gray\_background***);



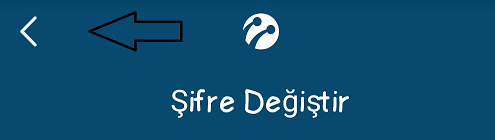
intent.putExtra(LoginSdk.***KEY\_SHOW\_FREE\_TEXT***, **"true"**);  
 intent.putExtra(LoginSdk.***KEY\_FREE\_TEXT***, **"** bla blab blab bla bla bla **"**);



intent.putExtra(LoginSdk.***KEY\_EXIT\_BUTTON\_DRAWABLE***, com.turkcell.loginsdktest.R.drawable.***a***);



intent.putExtra(LoginSdk.***KEY\_BACK\_BUTTON\_DRAWABLE***, com.turkcell.loginsdktest.R.drawable.***b***);



intent.putExtra(LoginSdk.***KEY\_CHECKHED\_CHECKBOX\_BUTTON\_DRAWABLE***, com.turkcell.loginsdktest.R.drawable.***custom\_icon***);



intent.putExtra(LoginSdk.***KEY\_TURKCELL\_LOGO\_DRAWABLE***, com.turkcell.loginsdktest.R.drawable.***d***);



intent.putExtra(LoginSdk.***KEY\_HINT\_TEXT\_COLOR***, R.color.***lsdk\_yellow1***);



intent.putExtra(LoginSdk.***KEY\_CAPTCHA\_BUTTON\_DRAWABLE***, R.drawable.***lsdk\_icon\_refresh***);

