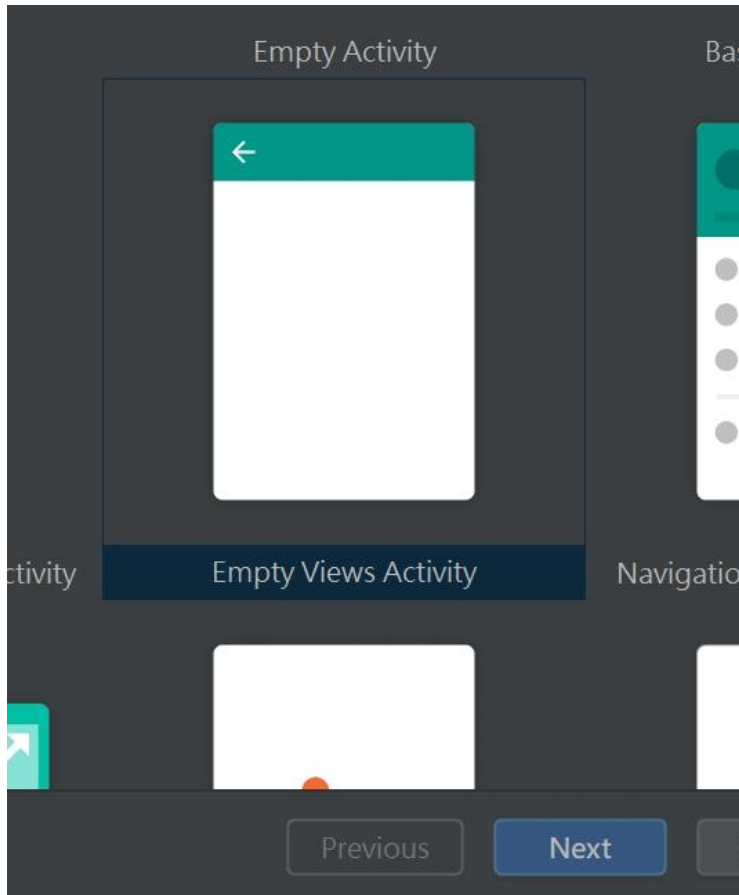


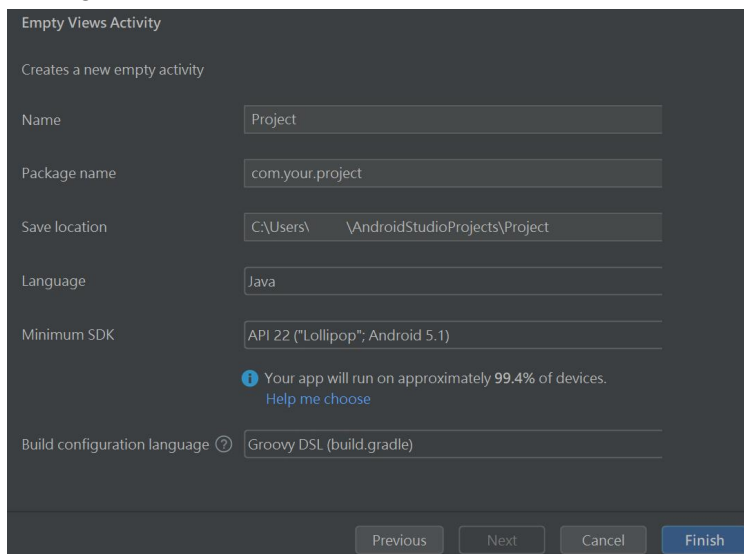
# Connect Tool Tutorial

The following are the initial steps

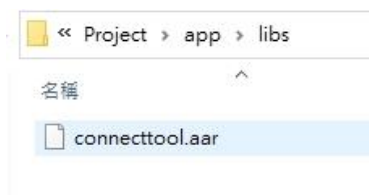
1. Create empty project



2. Settings

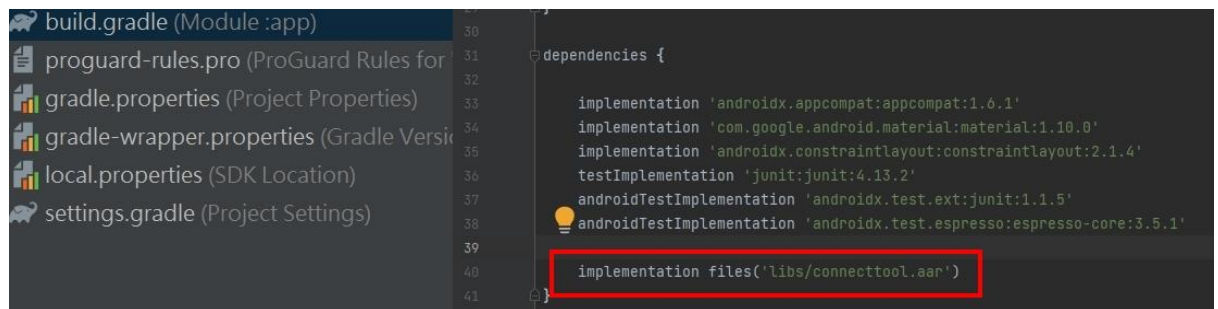


3. **Copy AAR File:** Copy the AAR library file into the 'libs' directory you created.



4. **Update Gradle Configuration:** Open the app-level build.gradle file in your project's module. Add the following code snippet within the 'dependencies' block:

```
implementation files('libs/connecttool.aar')
```



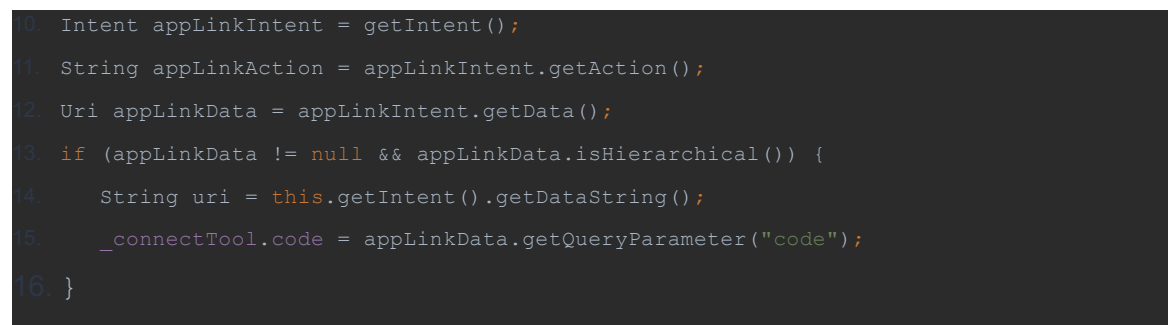
5. **Sync Gradle..**
6. **Open \app\src\main\AndroidManifest.xml to add:**

```
<uses-permission android:name="android.permission.INTERNET" />
<intent-filter>
    <action android:name="android.intent.action.VIEW" />
    <category android:name="android.intent.category.DEFAULT" />
    <category android:name="android.intent.category.BROWSABLE" />
    <data android:scheme="{ Get from redirect_uri's scheme }}" android:host="connectlink" />
</intent-filter>
```

7. **Importing Classes:**

```
import com.r17dame.connecttool.ConnectBasic;
import com.r17dame.connecttool.ConnectTool;
```

8. Create ConnectTool \_connectTool in MainActivity.
9. Copy to your **MainActivity.onCreate()** for deepLink :



## 17. Fill in relevant information


```
// Init tool
_connectTool = new ConnectTool( context: this, _state: "", _requestNumber: "", _redirect_uri: "", _RSAstr: "");
_connectTool.connectBasic = new ConnectBasic( _client_id: "", _X_Developer_Id: "", _client_secret: "", _Game_id: "", _referralCode: "");
```

The following are descriptions of the main functions.

SendRegisterData	
Intent	Complete registration
Step	<ol style="list-style-type: none"><li>1. Fill in registration information <code>_connectTool.CreateAccountInitData(String _email, String _password);</code></li><li>2. Passwords must have at least one <b>uppercase letter</b>/<b>lowercase letter</b>/<b>symbol</b>.</li><li>3. Password length must be 6 or more.</li><li>4. Get boolean result</li></ol> <pre>_connectTool.SendRegisterData(new ConnectCallback() {     @Override     public void callbackCheck(boolean value) {         Log.v(TAG, "RegisterData callback : " + value);     } });</pre>

SendLoginData	
Intent	After the user completes registration, execute <b>SendLoginData</b> .
Step	<ol style="list-style-type: none"><li>1. Fill in registration information <code>_connectTool.CreateAccountInitData(String _email, String _password);</code></li><li>2. Get boolean result</li></ol>

	<pre> _connectTool.SendLoginData(new ConnectCallback() {     @Override     public void callbackCheck(boolean value) {         Log.v(TAG, "LoginData callback : " + value);     } }); </pre>

OpenAuthorizeURL	
Intent	Get "code"
Step	<p><b>1. Fill in registration information</b></p> <pre> _connectTool.OpenAuthorizeURL(); </pre> <p><b>2. Authorize URL =&gt;</b>  <a href="https://gamar18portal.azurewebsites.net/connect/Authorize?response_type=code&amp;client_id=YourClientId&amp;redirect_uri=YourRedirect_uri&amp;scope=game+offline_access&amp;state=YourState">https://gamar18portal.azurewebsites.net/connect/Authorize?response_type=code&amp;client_id=YourClientId&amp;redirect_uri=YourRedirect_uri&amp;scope=game+offline_access&amp;state=YourState</a></p> <p><b>3. Open the browser to log in.</b></p> <p><b>4. Complete login</b></p>  <p><b>4. After logging in, deep link back to app.</b></p>

GetConnectToken_Coroutine	
Intent	Get ConnectToken.
Step	<ol style="list-style-type: none"> <li>connectTool.code is required.</li> <li>connectTool.code can be obtained through onDeepLinkActivated .</li> <li>After completing the OpenAuthorizeURL call: <pre> _connectTool.GetConnectToken_Coroutine(new ConnectTokenCall() {     @Override     public void callbackConnectToken(ConnectToken value) {         Log.v(TAG, "ConnectToken callback : " + value.access_token);     } }); </pre> </li> <li>Get ConnectToken. <pre> &lt;-- 200 OK https://qamar18portal.azurewebsites.net/connect/token (717ms) Content-Type: application/json;charset=UTF-8 Date: Fri, 17 Nov 2023 01:12:51 GMT Server: Microsoft-IIS/10.0 Cache-Control: no-store Expires: Thu, 01 Jan 1970 00:00:00 GMT Pragma: no-cache Set-Cookie: ARRAffinity=7a4eb409c3ffb246d105c1240b993b7fdf38fd864eb053c1b Set-Cookie: ARRAffinitySameSite=7a4eb409c3ffb246d105c1240b993b7fdf38fd864 Transfer-Encoding: chunked Vary: Accept-Encoding X-Powered-By: ASP.NET {   "access_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6IldGMlRHVC03Q1ZIVEdSWVZQZ   "token_type": "Bearer",   "expires_in": 3599,   "refresh_token": "CfDJ8IrwLQ0muyhLsIPqYgNcA1kCwkBCykHWP6Tv6l3Ed1_2tCK6H } &lt;-- END HTTP (2513-byte body) </pre> </li> </ol>

GetRefreshToken_Coroutine	
Intent	Get ConnectToken.

Step	<ol style="list-style-type: none"> <li>1. connectTool.refresh_token` is required.</li> <li>2. Call GetRefreshToken_Coroutine.</li> </ol> <pre> connectTool.GetRefreshToken_Coroutine(new ConnectTokenCall() {     @Override     public void callbackConnectToken(ConnectToken value) {         Log.v(TAG, "RefreshToken callback : " + value.access_token);     } }); </pre> <ol style="list-style-type: none"> <li>3. Get ConnectToken.</li> </ol> <pre> &lt;-- 200 OK https://qamar18portal.azurewebsites.net/connect/token (455ms) Content-Type: application/json;charset=UTF-8 Date: Fri, 17 Nov 2023 01:13:58 GMT Server: Microsoft-IIS/10.0 Cache-Control: no-store Expires: Thu, 01 Jan 1970 00:00:00 GMT Pragma: no-cache Set-Cookie: ARRAffinity=7a4eb409c3ffb246d105c1240b993b7fdf38fd864eb053c1b2 Set-Cookie: ARRAffinitySameSite=7a4eb409c3ffb246d105c1240b993b7fdf38fd864e Transfer-Encoding: chunked Vary: Accept-Encoding X-Powered-By: ASP.NET {   "access_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6IldGMLRHVC03Q1ZIVEdSWVVZQzI   "token_type": "Bearer",   "expires_in": 3600,   "refresh_token": "CfDJ8IrwLQ0muyhLsIPqYgNcA1k0w0g2kLJFBA4igPf-eq5YKr5ceF } &lt;-- END HTTP (2513-byte body) </pre>
------	--

GetMe_Coroutine	
Intent	Get User's <b>MeInfo</b> .
Step	<ol style="list-style-type: none"> <li>1. After GetConnectToken_Coroutine and get ConnectTokenModel</li> <li>2. `connectTool.access_token` is required.</li> <li>3. Call GetMe_Coroutine function</li> </ol> <pre> try {     connectTool.GetMe_Coroutine(new MeCallback() {         @Override         public void callbackMeInfo(MeInfo value) {             Log.v(TAG, "MeInfo callback : " + value.status);         }     }); } catch (NoSuchAlgorithmException e) {     throw new RuntimeException(e); } </pre>

```
}
```

#### 4. Get User's MeInfo.

```
<-- 200 OK https://r18gameapi.azurewebsites.net/api/Me?RequestNumber=ebe4d
Content-Type: application/json; charset=utf-8
Date: Fri, 17 Nov 2023 01:25:35 GMT
Server: Microsoft-IIS/10.0
Set-Cookie: ARRAffinity=7a4eb409c3ffb246d105c1240b993b7fdf38fd864eb053c1b1
Set-Cookie: ARRAffinitySameSite=7a4eb409c3ffb246d105c1240b993b7fdf38fd864e
Transfer-Encoding: chunked
Vary: Accept-Encoding
X-Signature: E9zyfSTnbKRuHyP4RK2U15pSUzdjJXre+r+g3QNs8SL88LhsaAUL52dTXtj5-
X-Powered-By: ASP.NET
{"data":{"email":"","nickName":null,"avatarUrl":null}}
<-- END HTTP (186-byte body)
```