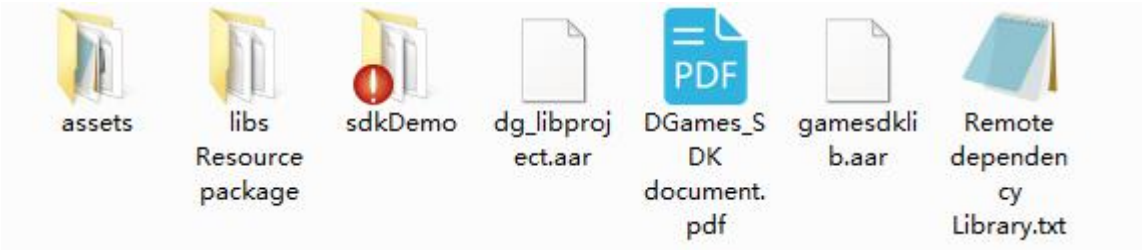
# DGames_SDK Access document

## Catalog

# 1.IDE To configure：

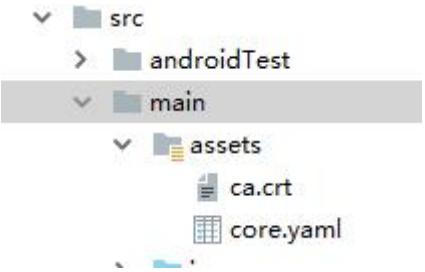
Decompress the compressed access document we provide，You can see the following in the root directory:assets、libs resource package、sdkDemo.zip、aar、DGames_SDKAccess documents and remote dependency library files：




**step：**

1.1 Import assets folder

Take assets folder import to Android Studio project，Location as shown below：




1.2 Import aar package

In the Android Studio project, two AAR packages are directly placed under the LIBS package，Add in build gradle of the project

```
Repositories{

 flatDir{

dirs 'libs'}}
```

Add two lines(If your Android studio is 3.0 ago, please change the following implementation to compile)

```
implementation(name: 'dg_libproject', ext: 'aar')
implementation(name: 'gamesdklib', ext: 'aar')
```

1.3 Import remote dependency Libraries (Be careful：Or copy the LIBS file directly to the corresponding location of the project, (Remote dependency library and LIBS resource bundle) Only one of them can be selected, and the remote dependency library is used below)

Copy and import remote dependency libraries from remote dependent library files: The following diagram:

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:27.1.1'
    implementation(name: 'dg_libproject', ext: 'aar')
    implementation(name: 'gamesdklib', ext: 'aar')
    implementation('org.web3j:core:3.3.1-android')
    implementation 'com.journeyapps:zxing-android-embedded:3.3.0@aar'
    implementation 'com.google.zxing:core:3.3.0'
    implementation 'com.squareup.okhttp3:okhttp:3.10.0'
    implementation 'com.android.support:multidex:1.0.3'
    implementation 'com.android.support.constraint:constraint-layout:1.1.2'
}
```

1.4 so File adaptation

Add the following code in build.gradle under app to fit the model:

```
ndk{

        abiFilters" armeabi-v7a" ," x86" ," armeabi"


        }
```

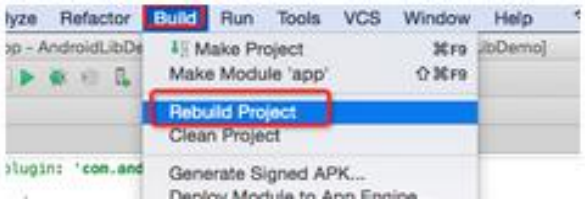
The following diagram:

```
android {
    compileSdkVersion 26
    defaultConfig {
        applicationId "com.dgames.sdkdemo"
        minSdkVersion 15
        targetSdkVersion 26
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
        multiDexEnabled true

        ndk {
            abiFilters"armeabi-v7a", "x86", "armeabi"
        }
    }
```

1.5 Recompile project:

Select from toolbar："Build"-->"rebuild project"




kindly reminder:There may be some minor differences in configuration due to the different versions of gradle, but Baidu can refer to relevant online configuration documents to introduce AAR packages into the project。

Note: If the file configuration is complete, the project reports an error: The number of Ex files referenced exceeds the method limit of 65536, please refer to the URL：https://www.aliyun.com/jiaocheng/804289.html

The above IDE configuration is complete, and the client configuration is started

# 2.client configuration： current version: v1.0.0

2.1：Permission and initialization sdk（Must be connected）

```
*Config.LOGIN.................Already logged in

*Config.UNLOGIN...............Not logged in

*Config.LOGINOUT..............Exit logon

*DGameManager.init(this,config)

*config.setLandscape.......The vertical and horizontal screen is set,true
                        as the horizontal screen,false as the vertical screen

*config.setAppId.............Provided by us appId

*config.setApibrowse..........Browser address to CP

*config.setFloatGravity.......Initial display position of suspended ball

*config.setDecimals...........decimals

*config.setDgameDebug.........Set up sub chain currency and dgame recharge
 address and ETH URL address true as test address false as formal address

*config.setLanguage...........en English   cn chinese

*@param     amount                 //Amount of payment
*@param     orderId                //Order number
*@param     toAddress               //Custom Payment Address
*@param     comment+System.currentTimeMillis //Payment information(Special
 note: to avoid repeated transactions, time stamps must be added to the payment
 information)

*@param     tokenId........transaction identifier
*@param     equip_info.....Details of game equipment (Including: ID, name,
                                        price, picture, notes,
                                        etc.Splicing in JSON form)
*@param     comment_erc........Game ID, server class ID, game device ID (Note:
                                        String separated by commas)
```

```
PermissionUtils                    //Permission utility class(Warm reminder:
first instantiate a tool class object.)
setPermission();                  //Access method of permission (These four
permissions are required in SDK, and the compiled version of Android studio
 is 23 or more)
getConfig();          //Gets the config object for the SDKinitialization call

QueryGameAmount(MainActivity.this);     // Current user chain currency query
                                    method (please refer to sdkdemo)
QueryDgasAmount();              //Current user DGAs query method (please refer
                                      to sdkdemo)
transErc(MainActivity.this, orderId, tokenId, equip_info,comment) //ERC_721 trans

transErcAddress(MainActivity.this, orderId,toAddress,tokenId,equip_info,comment)
                                      //ERC_721 Custom Transfer Address
gameQueryAssetErc(tokenId)            //ERC_721 Query asset attribution function

subStringRecharge();      //Directly adjusting DGAs or dgame recharge sub chain
                                      currency method
```

Call in main Activity:

```
  PermissionUtils permissionUtils=new PermissionUtils();

  Public void getConfig(){
  config=new Config();
  config.setAppId(“V431rSWOMpq3xGGJYSQTGH5ox1MBiXjJRw”);
  config.setApibrowse(“http://192.168.2.32:801”);
  config.setFloatGravity(FloatGravity.TOP_CENTER);
  config.setDecimals(“100000000”);
  config.setLanguage(“cn”);
  config.setDgameDebug(true);
  config.setLandscape(true);

  }
  perms = new String[]{
              Manifest.permission.CAMERA,
              Manifest.permission.WRITE_EXTERNAL_STORAGE,
              Manifest.permission.READ_PHONE_STATE,
              Manifest.permission.ACCESS_FINE_LOCATION
                 };
  private void setPermission(){

if(android.os.Build.VERSION.SDK_INT>=android.os.Build.VERSION.CODES.M){
        permissionUtils.requestRunPermission(perms,new PermissionListener()){

        @Override
        public void onGranted(){
                DGameManager.init(this,config);
                }
```

```
        @Override
        public void onDenied(List<String> deniedPermission){

            for(int i=0;i<deniedPermission.size();i++){

                showRequestRunPermission=ActivityCompat.shouldsShowRequestRunPe
rmissionRationale(this,.get(i));
        }
        if(showRequestRunPermission){

              setPermission();

          }else{
                showMissingPermissionDialog();
                }
                }
            });
        }else{
                DGameManager.init(this,config);
                }

  //Permissions callback method
  @Override
  public void onRequestPermissionsResult(int requestCode, @NonNull
String[]permissions, @NonNull int[] grantResults) {

  super.onRequestPermissionsResult(requestCode, permissions, grantResults);

    permissionUtils.onRequestPermissionsResult(requestCode, permissions,
grantResults);

    }

Private void showMissingPermissionDialog(){

AlterDialog.Builder builder=new AlterDialog.Builder(this);

  builder.setTitle(“prompt”);

  builder.setMessage(“For your normal use of SDK,please open the permissions!”);

  builder.setPositiveButton(“go Set!”,new DialogInterface.OnClickListener(){

               Public void onClick(DialogInterface dialog,int which){

                 startAppSettings();
        }
```

```
});
}
    Builder.setCancelable(false);

    Builder.show();
}
  Private void startAppSettings(){

    Intent intent=new Intent(Setting.ACTION_APPLICATION_DETAILS_SETTINGS);

    intent.setData(Uri.parse(“package:”+getPackageName()));

    startActivityForResult(intent,REQUEST_CODE_SDK_RESULT_PERMISSIONS);

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

  if (requestCode==REQUEST_CODE_SDK_RESULT_PERMISSIONS){

        setPermission();

  }
```

### 2.2：Add life cycle callbacks (must be)

Add to onStart of main Activity

```
DGameManager.onStart();
```

Add to onResume of main Activity

```
DGameManager.showFloatintView();
```

Add to onPause of main Activity

```
DGameManager.hideFloatintView();
```

Add to onStop of main Activity

```
DGameManager.onStop();
```

Add to onDestory of main Activity

```
DGameManager.onDestory();
```

## 2.3：API provided by SDK

The following table，The following ways of calling are：DGameManager.Function name(...)，Please refer to the specific method of use sdkDemo，

| Function name | effect | Parameter & type | describe |
|---|---|---|---|
| **Login(context)** | **login** | **context** | Direct call can |
| **setSDKLoginCallback( new ILoginCallBack())** | **Login callback (Please refer to demo)** | ILoginCallBack | OnSuccess（message）<br>Callback is successful and Cp ends are obtained:<br>Signdata Validation parameters<br>uname<br>address<br>OnFailed（message） |
| **Pay（context,orderId,amount，comment）** | **Pay** | Context<br>orderId<br>Amount<br>Comment | Direct call can |
| **setSDKPayCallback (new IPayCallback)** | **Pay callback** | IPayCallback | onPaySuccess(jsonStr)<br>Callback is successful and Cp ends are obtained:<br>payOrderId txid<br>OnPayFail(message) |
| **PayAddress（context,orderId,toAddress,amount，comment）** | **Custom Payment Addres** | Context<br>orderId<br>toAddress<br>Amount<br>Comment | Direct call can |
| **openUserCenter（）** | **personal Center** | | Direct call can |
| **isLogin()** | **Whether or not to log in** | | Direct call can |
| **queryGameAmount( context)** | **Subchain balance** | context | CP end gets the sub chain balance. |
| **queryDgasAmount( new Icallback)** | **DGAs balance** | | CP ends get DGAs balance |

| | | | |
|---|---|---|---|
| **transErc(context,orderId,tokenId,equip_info,comment)** | **ERC_721trans** | Context<br>orderId<br>tokenId<br>Comment | Direct call can |
| **transErcAddress(context,orderId,toAddress,tokenId,equip_info,comment)** | **ERC_721 Custom trans** | Context<br>orderId<br>toAddress<br>tokenId<br>Comment | Direct call can |
| **setSDKTransErcCallback(new ITransErcCallback)** | **ERC_721**Transfer callback | | onTransErcSuccess（**str**）<br>Successful submission of transaction<br>onTransErcFail（**str**）<br>Failure to commit transaction |
| gameQueryAssetErc(tokenId) | **ERC_721Query asset attribution** | tokenId | Direct call to return the user address attributable to the asset. |
| rechargeSubChain(context) | Dgas,dgame<br>Recharge sub chain | Context | Direct call can |

Note: refer to the server to pay the callback interface document for payment callbacks。