INSTANT GAMES PLATFORM SDK ACCESS MANUAL

|  |  |  |  |
| --- | --- | --- | --- |
| Made by | Chenxian Cao | Date |  |
| Review |  | Date |  |
| Approval |  | Date |  |

REVISION RECORD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Revised Version | Revised  Section | Revised Description | Author |
|  |  |  |  |  |
|  |  |  |  |  |

INTRODUCTION

Introduction …………………………………………………………………………………………………………..5

1. Introduction to COCOSSDK …………………………………………………………………………………5

2. Readers ………………………………………………………………………………………………………………5

3. Document Description ………………………………………………………………………………………..5

4. Keyword …………………………………………………………………………………………………………….6

Access Process ………………………………………………………………………………………………………..6

Abbreviations list: < Description of the abbreviations used in this article, with a full English name and an explanation in Chinese for each abbreviation>

| Abbreviation | English Name | Chinese Explanation |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

INTRODUCTION

1. **COCOS SDK Introduction**

## The Instant Games Platform COCOS SDK is a rapid development kit developed by Chukong Technology for third-party applications. Currently, the COCOS SDK is supported to run H5 PC games and Runtime games. By docking the COCOS SDK, you can quickly integrate your app into Instant Games.

1. **Target Audiences**

The target audience for this document includes the following:

* Demand Management Personnel

Demand managers read this document to understand whether the system design meets the business needs, including non-functional requirements.

* Designer:

Designers read this document to understand the overall architecture of the system and to guide subsystem design and module design.

* Developer

Developers read this document to understand the overall architecture of the system and as a guiding document for development.

* Tester

Testers read this document to understand the overall architecture of the system and can be used to verify that the system development is in line with the business scenario.

## 3.Documentation

This document mainly describes CocosPlay's access process, parameter definition, and provides demo integration examples so that developers can quickly integrate and release. For overseas users, we have applied the plug-in mode of AppBundle in order to enable users to have a pleasant game experience with cocos mini game engine at a lower access cost. For more information, refer to the official document of Android App Bundle https://developer.android.com/platform/technology/app-bundle.

Android Studio has a minimum requirement of 3.2 to access App Bundle.

**4.Keyword**

Host: APP Accessing SDK

**Access Process**

1. Add the following dependencies to the host build. gradle

implementation fileTree(include: ['\*.jar'], dir: 'libs')

implementation fileTree(dir: 'libs/', include: ['\*.aar'])

api 'com.android.support:multidex:1.0.3'

implementation 'com.android.support:appcompat-v7:26.1.0'

api 'com.android.support:recyclerview-v7:26.1.0'

api 'com.squareup.okhttp3:okhttp:3.8.1'

api 'com.squareup.retrofit2:retrofit:2.2.0'

api 'com.github.bumptech.glide:glide:3.8.0'

api 'com.github.CymChad:BaseRecyclerViewAdapterHelper:2.9.18'

api 'com.squareup.retrofit2:converter-gson:2.4.0'

api 'com.squareup.retrofit2:adapter-rxjava2:2.4.0'

api 'io.reactivex.rxjava2:rxjava:2.1.12'

api 'io.reactivex.rxjava2:rxandroid:2.0.2'

implementation 'com.github.xiaohaibin:XBanner:1.6.9'

implementation 'com.google.android.play:core:1.6.1'

api 'com.google.android.gms:play-services-ads:16.0.0'

2. Accessing cocoslib.aar and lib-rt-frame.aar under host LIBS

3. The host's myapplication can be modified in two ways

1. public class MyApplication extends SplitCompatApplication{}
2. public class MyApplication extends Application {

@Override

protected void attachBaseContext(Context context) {

super.attachBaseContext(context);

SplitCompat.*install*(this);

MultiDex.*install*(this);

}

}

4. Import project cocos\_game\_jar (this project only contains cocos so file, because so file is large, so as feature, when users enter the platform, download the so file corresponding to the mobile phone CPU architecture dynamically, which achieves the purpose of reducing the host package). Add the following content in the build.gradle

bundle {

language {

enableSplit = false

}

density {

enableSplit = false

}

abi {

enableSplit = true

}

}

dynamicFeatures = [":cocos\_game\_jar"]

repositories {

flatDir {

dirs 'libs'

}

}

sourceSets {

main {

jniLibs.srcDirs = ['libs']

}

}

5. Add to the strings. XML of the host

<string name="title\_dynamic\_feature">Cocos Play</string>

6. The host needs to call the initialization method before opening the page:

CocosConfig config = new CocosConfig()

.setChannelId(channelId)

.setDeviceId(deviceId)

.setModelName(modelName)

.build();

CKGameSDK.init(config);

Parameter Description:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Implication | Type | If Empty | Remarks |
| channelId | Game ID | string | Nonempty | Manage background applications, such as 160718 |
| deviceId | Game ID | string | Nonempty | Unique Identification of Current Equipment |
| modelName | Cocos Feature Name | string | Nonempty | By default, "cocos\_game\_jar" is used. If the host needs to change its name, it needs to be passed in accordingly. |

7. During the testing, the host must set up the method of using test advertisement after init to prevent the risk of blocking the advertisement account caused by malicious advertisement brushing. You can remove it before you go online. When testing a formal advertisement, try not to click on the download page when there is an advertisement display.

CKGameSDK.setTestAd(context);

8. When the host pulls up the Cocos Mini Game platform, it calls the interface

CKGameSDK.startHomeActivity(context);

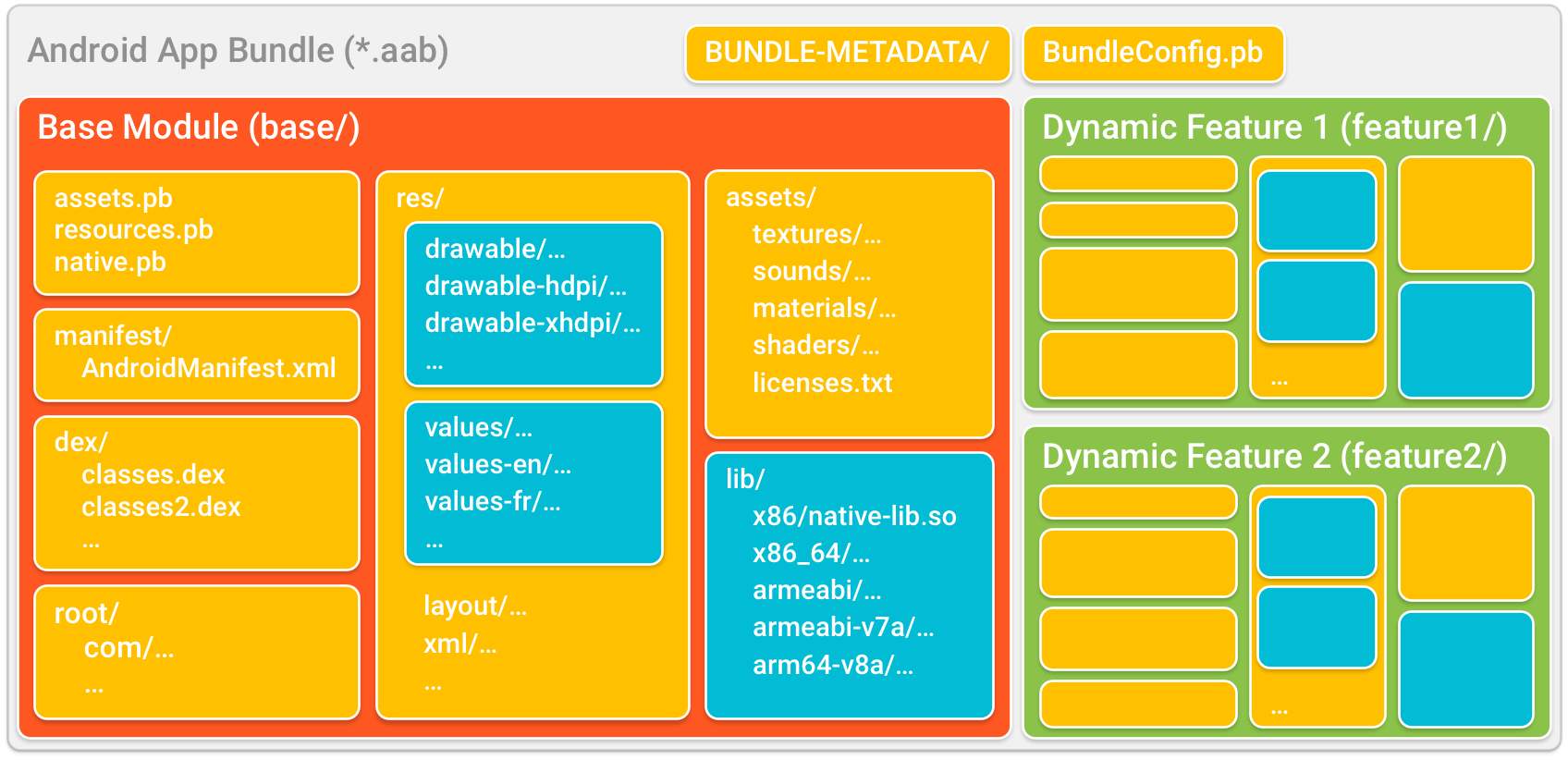
9. The host is pulling up a game and calling the interface

CKGameSDK.startGameActivity (context，gameId);

Parameter Description:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P**arameter** | **Implication** | **Type** | If Empty | Remarks |
| gameId | Game ID | string | Nonempty | Provide by Cocos |

10. Packaging process brief, after accessing App Bundle, because of features, the host need to generate tab packages uploaded to Google Play. The following is an official illustration.



Select the following graphic options when you export the package

