

Android 组件化实战

版本号: 2019/2/2-1(0:14)

- [Android 组件化实战](#)
 - [项目模块的拆分](#)
 - [项目模块的详细拆分](#)
 - [lib_base](#)
 - [AS3.3需要使用api](#)
 - [module](#)
 - [module_remind](#)
 - [Module如何能独立运行?](#)
 - [module_home](#)
 - [报错/故障](#)
 - [No resource found that matches the given name](#)
 - [You need to use a Theme.AppCompat theme](#)
 - [应用安装后具有多个图标](#)
 - [ARouter::There is no route match the path](#)
 - [Fragment前后覆盖多层，都可以看见。](#)
 - [参考资料](#)

不论述组件化的基本功能。

利用ARouter框架来实现项目的组件化。

项目模块的拆分

1、项目拆分为三大部分

1. 第一部分：app，也就是壳工程。
2. 第二部分：通用library，这里拆分为两个。
 1. lib_base: 通用模块，一些封装好的通用接口。
 1. lib_res: 资源库。放置通用图片、assets、xml等通用的资源文件。
3. 第三部分: 组件，根据业务进行划分。
 1. module_remind: 直播提醒、订阅、预约等。
 1. module_home: 首页相关业务

项目模块的详细拆分

1、项目最外层下的gradle.properties来控制整个项目编译和独立测试的开关

1. 需要通过开关，来进行各个模块、通用库的独立测试

#是否需要测试通用lib库，true-开启独立测试 false-不开启

isTesting_lib_base = true

#是否测试直播提醒、追剧预约、新片预约等相关功能。

isTesting_module_remind = true

lib_base

1、lib_base的作用和需要配置的内容

1. lib_base 所有Module需要用的 基本库，
2. 用于提供基本的接口、通用工具等内容。
3. 引入并且配置ARouter(路由框架)

2、lib_base的build.gradle

1-通过开关来选择是application模式，还是library模式。

```
// 开启测试，就采用application进行独立运行
if(isTesting_lib_base.toBoolean()){
    apply plugin: 'com.android.application'
}else{
    // 没有开启测试就作为library
    apply plugin: 'com.android.library'
}
```

2-需要和主app选择一致的API版本和库的版本

```

android {
    // 需要和app一致
    compileSdkVersion 23
    buildToolsVersion '25.0.0'

    // 需要和app一致
    defaultConfig {
        minSdkVersion 17
        targetSdkVersion 28
        //...
    }
}

dependencies {
    // 需要和app一致
    compile 'com.android.support:appcompat-v7:24.0.0'
    //...
}

```

AS3.3需要使用api

3、Android Studio 3.3中lib引用的基础库需要使用api才能被其他module引用这些基本库

```

// 例如
api 'androidx.appcompat:appcompat:1.1.0-alpha01'
api 'androidx.constraintlayout:constraintlayout:1.1.3'

```

module

module_remind

1、module_remind需要如何配置？

1. 配置build.gradle
2. 配置AndroidManifest
3. 创建TestActivity，用于独立测试时运行。

2、module的build.gradle

1-通过开关来选择是application模式，还是library模式。

```

// 开启测试，就采用application进行独立运行
if (isTesting_module_remind.toBoolean()) {
    apply plugin: 'com.android.application'
} else {
    // 没有开启测试就作为library
    apply plugin: 'com.android.library'
}

```

2-参考lib_base,需要和主app一致。

```
compileSdkVersion 23
buildToolsVersion '25.0.0'

minSdkVersion 17
targetSdkVersion 28

compile 'com.android.support:appcompat-v7:24.0.0'
```

Module如何能独立运行?

Module的独立运行，需要去额外配置。

1、Module能独立运行需要配置哪几个东西?

1. 创建TestActivity，用于测试盒独立运行。
2. 配置AndroidManifest，让TestActivity作为启动的Activity
3. gradle.build需要正确配置开关、API版本

2、新建TestActivity

```
package com.zte.ipstvclient;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

import com.zte.ipstvclient.module_remind.R;

public class TestActivity extends AppCompatActivity {

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

3、AndroidManifest配置Activity和Theme

1. Activity设置为LAUNCHER activity
2. Theme要指定为 @style/Theme.AppCompat.Light.NoActionBar，否则会崩溃。

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.zte.iptvclient.module_remind">
    <application
        android:allowBackup="true"
        android:label="@string/app_name"
        android:supportsRtl="true">
        <activity android:name=".TestActivity"
            android:theme="@style/Theme.AppCompat.Light.NoActionBar">
            <intent-filter>
                <category android:name="android.intent.category.LAUNCHER"/>
                <action android:name="android.intent.action.VIEW"/>
                <action android:name="android.intent.action.MAIN"/>
            </intent-filter>
        </activity>
    </application>
</manifest>
```

4、build.gradle的配置

```
// 开启测试，就采用application进行独立运行
if (isTesting_module_remind.toBoolean()) {
    apply plugin: 'com.android.application'
} else {
    // 没有开启测试就作为library
    apply plugin: 'com.android.library'
}

android {
    compileSdkVersion 23
    buildToolsVersion '25.0.0'

    defaultConfig {
        minSdkVersion 17
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:24.0.0'
    compile 'com.android.support.constraint:constraint-layout:1.0.2'
    testCompile 'junit:junit:4.12'
    compile project(':lib_res')
}
```

module_home

报错/故障

No resource found that matches the given name

错误信息：

Error:(3) Error retrieving parent for item: No resource found that matches the given name 'android:TextAppe

原因:

新建的Module里面采用了高版本的API, 如下:

```
compile 'com.android.support:appcompat-v7:28.+'
```

解决办法:

选择低API版本的库

```
compile 'com.android.support:appcompat-v7:24.0.0'
```

You need to use a Theme.AppCompat theme

错误信息:

Caused by: java.lang.IllegalStateException: You need to use a Theme.AppCompat theme (or descendant) with th

应用安装后具有多个图标

1、应用安装后app+4个lib, 正常情况是1个图标, 结果有5个图标

1-需要将AndroidManifest分为两个

2-src/main/debug 下放置module独立编译时的AndroidManifest

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.hao.lib_base">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".TestActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>

```

3-src/main/ 下放置统一构建使用的AndroidManifest(移除LAUNCHER相关部分)

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.hao.lib_base">

    <application android:theme="@style/AppTheme">
        <activity android:name=".TestActivity">
        </activity>
    </application>

</manifest>

```

4-修改build.gradle, 选择合适的AndroidManifest.xml(记住debug目录不允许为 src/debug 会导致 exclude失效, 我将其放置到main文件夹中 src/main/debug 才能有效地清除)

```

sourceSets {
    main {
        if (isTestingLibBase.toBoolean()) {
            manifest.srcFile 'src/main/debug/AndroidManifest.xml'
        } else {
            manifest.srcFile 'src/main/AndroidManifest.xml'
            java {
                //全部Module一起编译的时候剔除debug目录
                exclude 'src/main/debug/'
            }
        }
    }
}

```


ARouter::There is no route match the path

解决办法:

1. 不同Module的一级目录(group)不能相同

Fragment前后覆盖多层，都可以看见。

解决办法:

1. 系统默认Fragment背景为透明
2. add多个Fragment后会依次按顺序展示，需要给Fragment主动设置背景色。

参考资料

1. [使用阿里ARouter路由实现组件化（模块化）开发流程](#)
2. [ARouter组件化项目-源码参考](#)