
WOTD
51CTO
全球
World Of Tech 2017
2017年12月1日-2日 • 深圳中洲万豪酒店
软件开发技术峰会
DEVELOPMENT

WOTD

51CTO

基于Gradle插件的SDK产品集成方案

余勋杰

游族Mob云平台　技术副总监

Mob开发者平台

- SDK被下载300万次，服务超过36万个App
- MAU超过8亿，DAU超过2亿
- 累计覆盖超过7.4亿独立设备，国内覆盖率超过95%
- 累计5年无重大事故

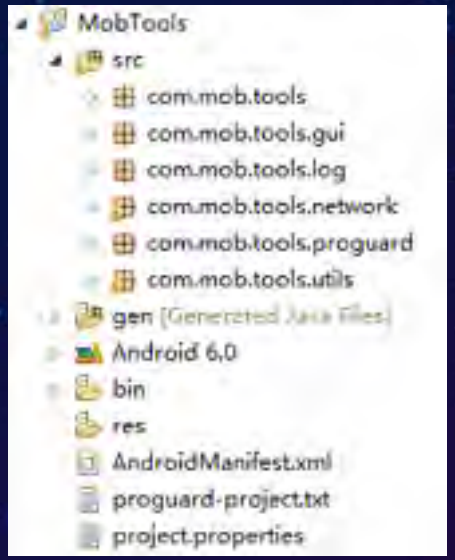
一切源自于一个小纠结

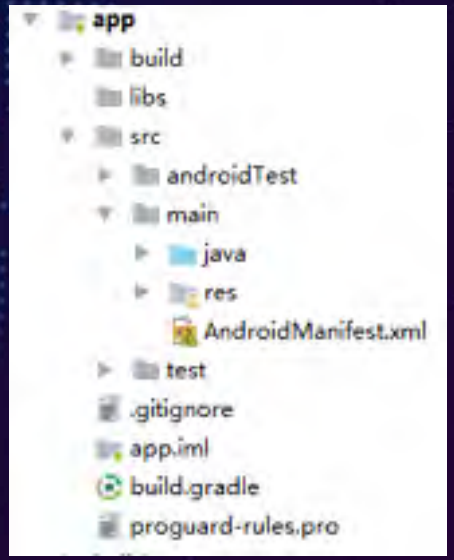
默认的Gradle脚本

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 26
    buildToolsVersion "27.0.0"
    defaultConfig {
        applicationId "com.example.yuxj.myapplication"
        minSdkVersion 15
        targetSdkVersion 26
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}
```

默认脚本的问题

- 项目目录结构不一致





- 编译参数硬编码

Mob内部项目中的通用Gradle脚本

```
sourceSets {
    main {
        manifest.srcFile 'AndroidManifest.xml'
        assets.srcDirs = ['assets']
        java.srcDirs = ['src']
        aidl.srcDirs = ['src']
        res.srcDirs = ['res']
        jniLibs.srcDirs = ['libs']
    }
}

buildTypes {
    release {
        minifyEnabled !isLibrary(projectDir)
        proguardFiles GlobalVariables.proguardFile
    }
}

lintOptions {
    checkReleaseBuilds false
```

Mob通用脚本的使用方法

```
apply plugin: 'com.android.library'
apply from: '../ASProj-CloudStorage/EclipseModule.gradle'

dependencies {
    compile project(':MobCommons')
}
```

```
apply plugin: 'com.android.application'
apply from: '../ASProj-CloudStorage/EclipseModule.gradle'

dependencies {
    compile project(':CloudStorage')
}
```

Mob通用脚本的问题

- 每个项目都需要复制这个文件
- 引用通用脚本时每个项目的路径不一样
- 遇到脚本修改时，只能通过手动替换来实现多项目同步

英雄登场

什么是Gradle插件

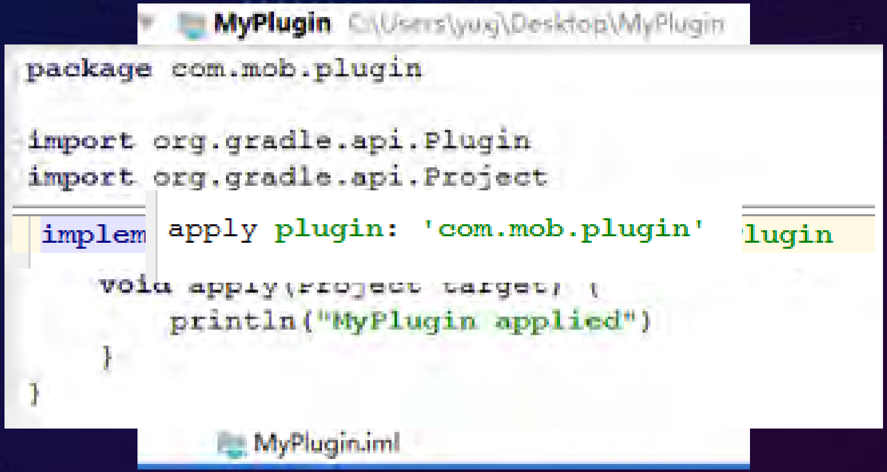
- 是一
- 可以 加方法
- 能监听脚本的各个执行过程，并作出反应
- 其实

```
apply plugin: 'com.android.library'
apply from: '../ASProj-CloudStorage/EclipseModule.gradle'

dependencies {
    compile project(':MobCommons')
}
```

```
apply plugin: 'com.android.application'
apply from: '../ASProj-CloudStorage/EclipseModule.gradle'

dependencies {
    compile project(':CloudStorage')
}
```

WOTD 51CTO

最简单的Gradle插件



添加Groo[illegible]能

由通用脚本演变而来的Gradle插件

```
class EclipseModuleConfig {
    Project project
    String buildTool
    int compileSdkVersion
    Project appModule
    String proguardFile
    int targetSdkVersion
    int minSdkVersion
    String applicationId
    int versionCode
    String versionName

    // 初始化编译参数
    void init(Project proj) {
        this.project = proj

        // 使用本机最新版本的build tools

        // 解析application模块中的project.properties，确定编译时采用的SDK level
```

SDK集成和Maven库依赖

时下流行的SDK集成流程

- 在“开放平台”页面下载客户端SDK压缩包
- 阅读艰涩的“集成文档”，然后复制文件、修改配置
- 但还是会编译不过，重头检查集成步骤
- 耐着性子对比了三遍集成文档，终于完成集成
- 立誓从此不再碰这个模块

基于Gradle的Maven库依赖流程

- 在根脚本，或依赖链上的所有项目中加入Maven仓库地址
- 在依赖链底端模块“dependencies”中填写依赖库名称、版本

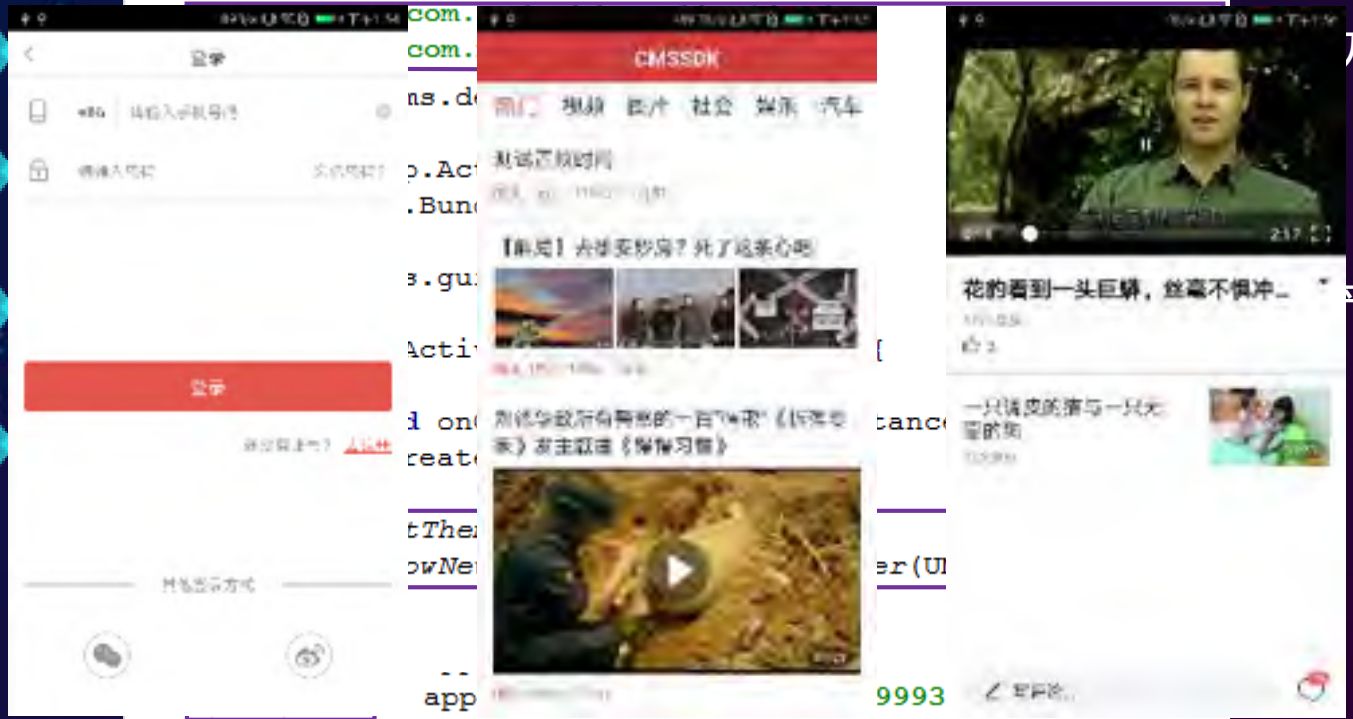
```
dependencies {
    compile 'com.android.support:support-v4:27.0.1'
    compile 'com.mob:Jimu:+@aar'
}
```

- 根据各类“教程”调用方法实现功能

两类集成方式的对比

- SDK类集成方式纯手工操作、步骤繁多，容易出错
- Maven库依赖方式属于半自动操作，步骤很少，容易使用
- SDK类集成方式为离线集成，版本更新麻烦
- Maven库依赖需要网络环境，可以实现自动更新版本
- SDK集成方式功能丰富，Maven集成方式功能单一

MobSDK的集成方式



方案

产品

MobSDK做了什么

- 使用Maven库依赖的方式，引入Mob各个产品线的SDK
- 依照各产品线要求，完成AndroidManifest.xml文件的配置
- 为个别产品线做特殊的处理，如产生ShareSDK.xml文件
- 自动加入MobSDK的初始化代码

SDK+

MobData

- 依托Mob开发者服务平台扎实基础
- 多方合作，打通线上线下数据生态
- 800+ 兴趣标签，全方位了解目标用户
- 客户覆盖各行各业

数据反哺，互利共生

- SDK+大数据用户标签
- 多维度用户组画像，洞察业务特质，智能趋势预测
- SDK+用户行为预测
- 海量数据建模，千人千面精准画像，精准营销

Thank you!