

前言

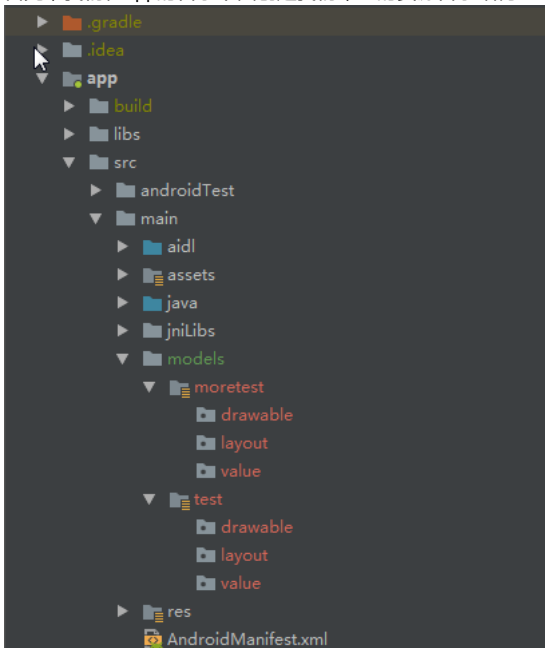
本次调研的目的在于为项目的分包做准备，将资源文件按模块划分，方便在之后的分包过程中能按模块添加/修改相关资源（主要包括drawable,layout和value三类文件）。

资源分包

资源分包是指分离各个模块的资源文件，这样需要我们按模块划分res，我们在使用AS的时候是不能对res目录进行分包，所以考虑通过Gradle的配置来指定资源目录。

调研发现，我们可以通过gradle配置中sourceSet来指定哪些源文件要被编译，所以进行了如下操作。

首先，我们在app的目录下面创建我们希望的资源目录结构：



从上图可以看到，在models下面，我们创建了moretest和test两个资源文件目录，下面包括了drawable、layout、value三个文件夹（这里的目录结构可以与res下面的一样，比如包括hdpi-drawable等）。

然后在app目录下创建config.gradle文件，并添加下面代码：

```
ext{
    srcDirs = [
        'src/main/models/moretest',
        'src/main/models/test',
        'src/main/res',
    ]
}
```

srcDirs中就是我们设置的自定义模块的路径，该路径是相对于app文件夹的路径，其中如果你的原资源文件夹还在使用，就需要设置“src/main/res”路径，否则可以不设置这个路径。

接下来还需要修改app目录下面的build.gradle文件：

```
apply from: "config.gradle"

android {
    sourceSets{
        main{
            res.srcDirs = project.ext.srcDirs
        }
    }
}
```

我们也可以直接给res.srcDirs赋值，但是在协作开发的过程中，这样不好。一般在合作开发中，我们的build.gradle文件的变动是很少的，本地运行的项目中依赖了很多个性化的东西（比如数据库管理或者增量编译工具），而这些东西可能影响到其他人的工程，所以是禁止上传到代码库的，而我们的模块添加又是一个

频繁的工作，所以，我们不能直接修改build.gradle，而是使用ext的方式。

模块化插件

为什么需要使用插件？

在上面资源分包的过程中，我们知道，我们的流程一般是这样的：

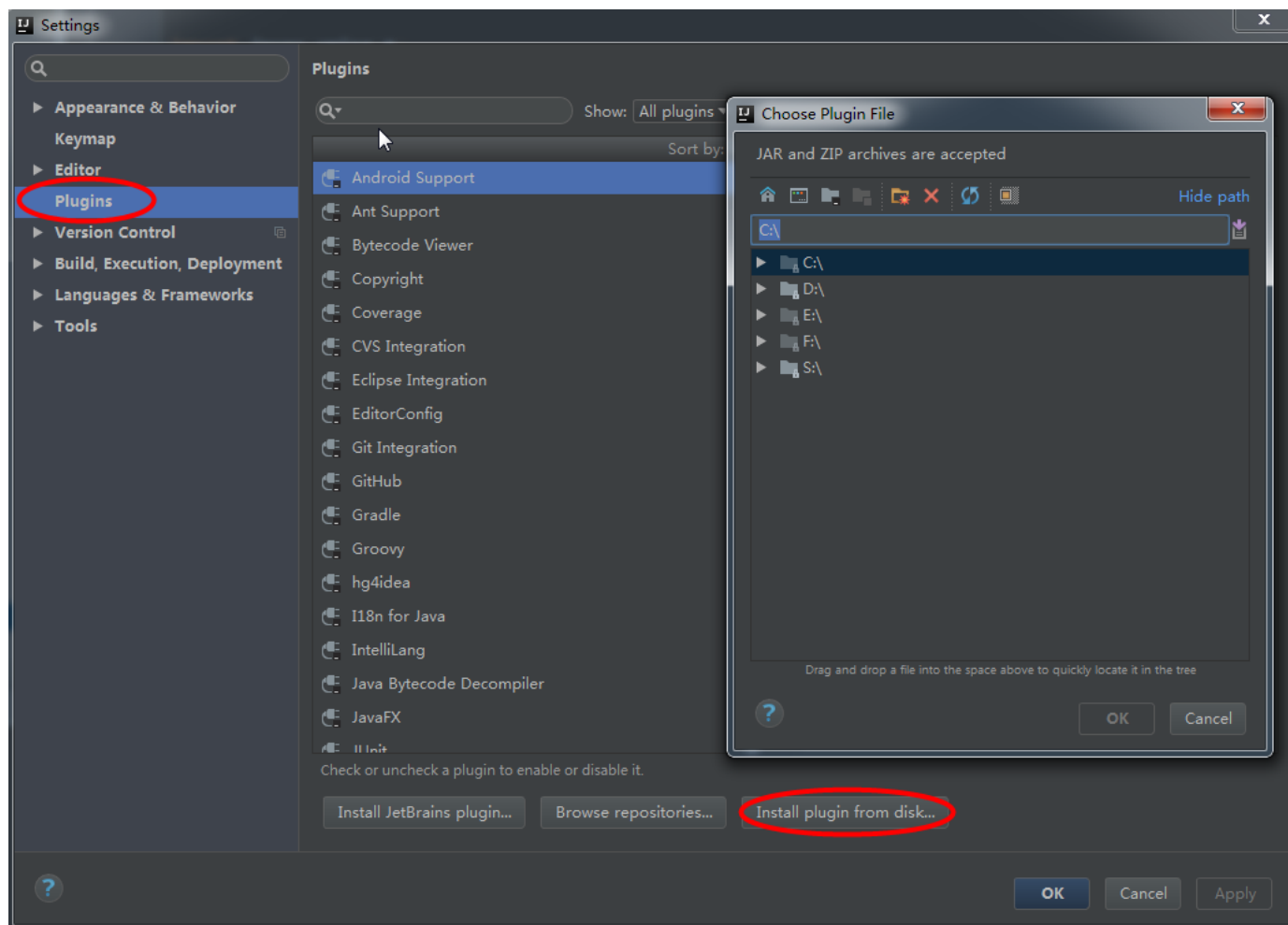
创建模块文件夹（test）→创建资源文件夹（drawable、layout、value）→修改config.gradle文件

这样的流程是没有毛病的，但是比较繁琐，另外还考虑到在创建模块的时候，可能还涉及到java包下面的模块目录添加，这就需要一些繁琐的文件夹创建操作，比较浪费时间，再考虑到这一步可能会因为命名而造成很多不必要的麻烦，所以考虑使用IEDA的插件来帮助完成新模块所有文件夹的创建及config.gradle的修改。

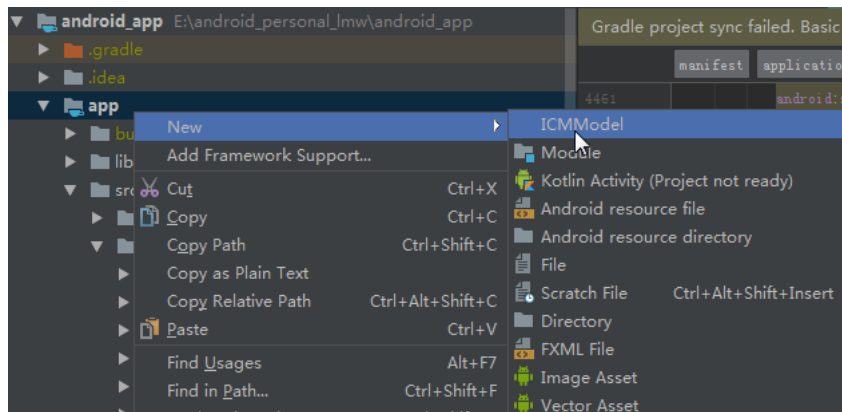
如何使用插件？

将jar包作为插件导入IDEA，流程如下：

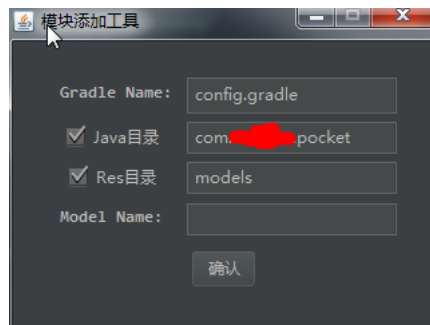
File → Settings → Plugins → Install Plugin From Disk → 选择jar包位置 → OK → restart



IDEA重启之后，我们选择需要添加模块的工程，右键，New，选择ICMModel，如下图所示：



选择以后，我们可以看到插件界面，如下图所示：



在界面中我们可以看大我们可以修改/选择四项内容，它的的含义如下：

- **Gradle Name**: 设置了ext内容的gradle文件的文件名。就是在“资源分包”中提到的`config.gradle`，这一项默认填充“`config.gradle`”。
- **Java目录**: 可以通过改变勾选状态来决定是否需要在Java目录下生成模块目录，后面为模块所在的包（格式如`xxx.xxx.xxx`）。在Java目录下生成的模块目录结构可见[附录1](#)。
- **Res目录**: 可以通过改变勾选状态来决定是否需要在Models目录下生成新的资源模块，后面为资源模块所在的包名。如上图的例子，资源目录的实际路径就是（`src/main/models/xxx`），在`models`目录下生成的模块目录结构可见[附录2](#)。
- **Model Name**: 模块命名。

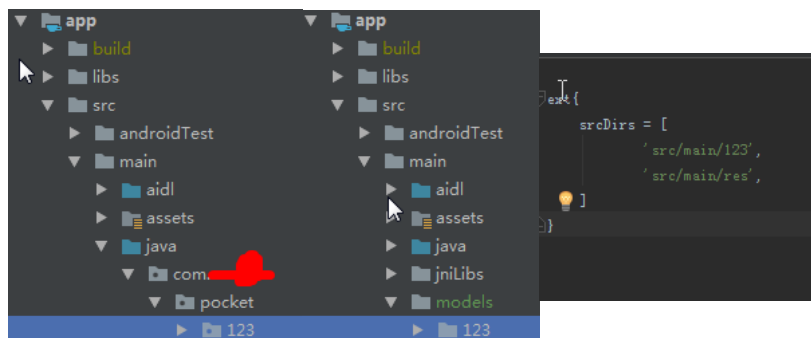
以上面的例子来说，在使用中有下面几点需要注意：

1. config.gradle 这个文件必须有用户先创建好并且与app下面的build.gradle关联，并且初始的config.gradle文件必须要包含以下代码（因为创建新模块时修改的依据是判断“[/n”这个符号）：

```
ext{
    srcDirs = [
    ]
}
```

2. models 这个目录需要手动创建（有强迫症，不予许直接在main目录下面添加资源模块）

输入了模块名后点击确认，工具会自动的添加模块目录以及修改gradle文件，比如我们模块命名为“123”，则添加内容有：



同时，工具会在app目录下生成一个`config.icm`文件，文件存储了一段json字符串，内容如下：

```
{"needJava":true,"needRes":true,"javaPkgName":"xxx.xxx.xxx","resPkgName":"models","extFileName":"config.gradle"}
```

这个文件记录了你上次操作时候的配置，会在你下次再创建模块时读取，避免每次都需要修改。

总结

本次调研并证实了Android资源文件分包的可行性，并开发了用于帮助模块化开发的辅助工具，目前分包方案及工具均满足实际需求并且能有效的提高开发人员的工作效率。

由于初次进行IDEA插件开发，并且没有找到详细的帮助文档，所以插件的开发比较粗略，另外为了方便直接使用的Java Swing 所以界面看起来不太美观，但是插件功能以及满足了现有需求，所以如果有大神将插件优化了或者有更好的插件，希望可以多交流交流（吐槽下IDEA Plugins的文档，找了丫好久都找不到）。

鸣谢

[简单的中文文档](#) ☺

附录一：Java模块结构

- activity
- adapter
- callback
- constants
- dao
- event
- fragment
- event
- net
- pojo
- proto
- req
- service
- util
- vo
- wedgit

附录二：Res模块结构

- layout
- drawable
- value