KP模板接入指南

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本号** | **修改人** | **修改内容** | **日期** | **备注** |
| V1.0 | 李鹏辉 | 首次编写 | 2017.10.17 | 简版 |
|  |  |  |  |  |
|  |  |  |  |  |

1. 快速入门
2. 安装Android studio；
3. 下载KPMould源码；
4. 配置开发。
5. 安装Android studio

下载Android studio软件并安装。

1. 下载KPMould源码

从gitlab上同步下KPMould源码。可以通过以下两个地址同步源码：

1. HTTP方式地址：<http://gitlab.kawakp.com/android_app/KPMould.git>
2. SSH方式地址：[git@gitlab.kawakp.com:android\_app/KPMould.git](mailto:git@gitlab.kawakp.com:android_app/KPMould.git)

例：

git clone <http://gitlab.kawakp.com/android_app/KPMould.git>



1. 基于模板开发应用

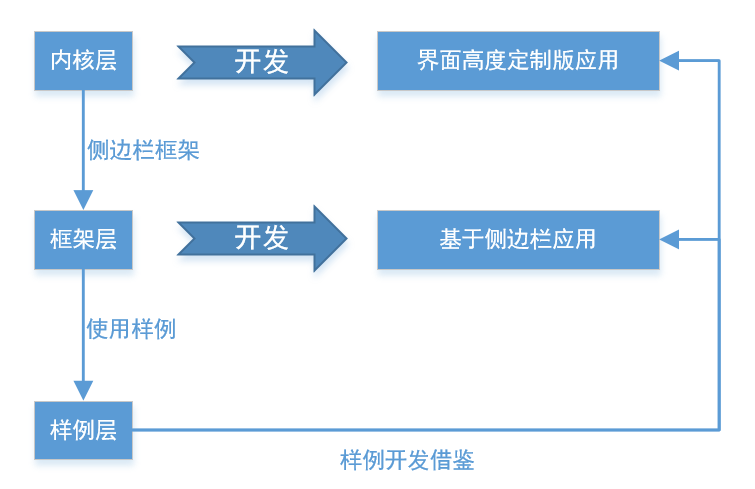


图4-1 KP模板结构图

KP模板结构图如图4-1所示。由图可以看出KP模板分三层：

* 内核层（git 的master分支）。该层没有封装如何UI层的东西，所有需求高度定制的应用都可以基于该层进行开发。
* 框架层（git 的 side分支）。该层除了内核外还分装了侧边栏框架，所有可以复用侧边栏的应用都可以基于该层进行开发。
* 样例层（git的demo分支）。该层主要是对内核层和框架层的应用，包括一些使用样例（MVP、折线图、GIF、数据库使用）。

这三层只有内核层和框架层是开放出来开发Android应用程序的。且这两层开发略有不同，下面对这两层的快速开发一一进行说明。

1. 基于内核层开发
2. 准备工作：完成第三大点、第四大点所有内容；
3. 开发配置：基于该模板进行开发需要配置一些插件。
4. 首先模板中第三方库是统一管理的，统一管理文件在Project -> config -> dependencies.gradle文件中，如图4-2所示。



图4-2 第三方库管理文件地址

1. Project里的build.gradle配置：分别设置kotlin、android\_plugin、lambda、realm的版本，然后引入这几个插件和第三方库配置文件，如图4-3所示。



图4-3 Project配置

1. Kernel lib中build.gradle配置：里面的所有配置请不要随意修改，避免编译出错或无法使用。
2. application中build.gradle配置：由于模板使用了databinding、lambda、realm、kotlin插件，所以需要引入相关插件和兼容包，如图4-4、图4-5所示。

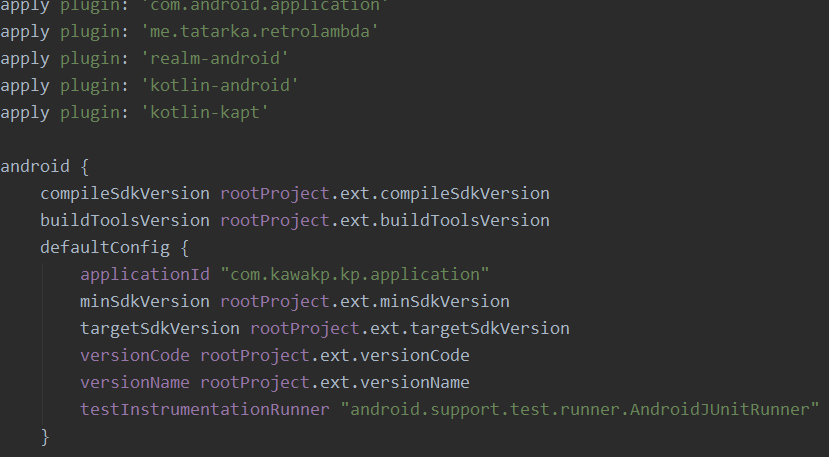


图4-4 插件引入配置



图4-5 插件兼容配置

1. AndroidManifest配置：

* Application中配置KpApplication或者继承于KpApplication的类；
* 建议所有activity配置下面这条配置，防止设备状态改变重新创建activity导致的问题。配置如图4-6所示。

android:configChanges="keyboardHidden|orientation|screenSize|locale|screenLayout|mcc|mnc"

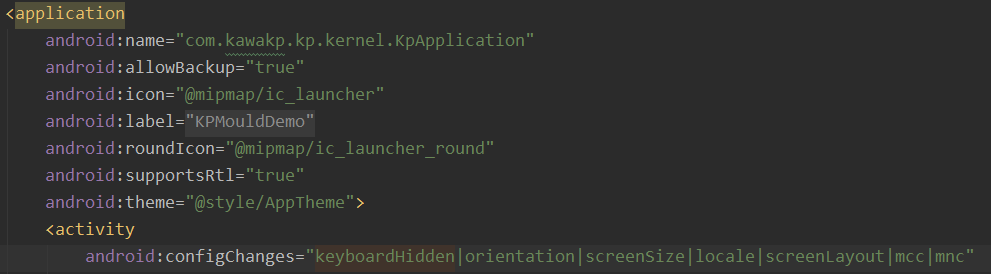


图4-6 AndroidManifest配置

1. 应用开发：开发中所有应用都基于BaseActivity、BaseFragment开发，且都是采用MVP框架开发。开发基类和MVP结构样例如图4-7所示。

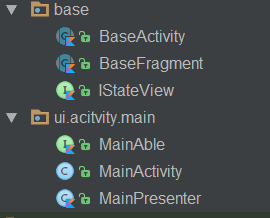


图4-7 基类和样例

图4-7中base文件夹中是需要使用的基类，base中的基类实现了view和视图状态切换相结合。ui文件夹下是一个MVP框架的使用样例。MainAble是需要view来实现接口，该接口继承IView，如图4-8所示；

MainPresenter是实现一些业务逻辑等，并引入view接口；

MainActivity是实现view的相关接口，便于MainPresenter使用接口方法。

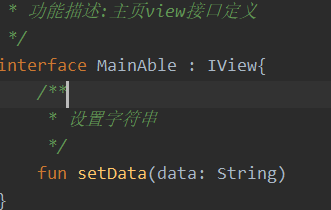


图4-8 MainAble接口

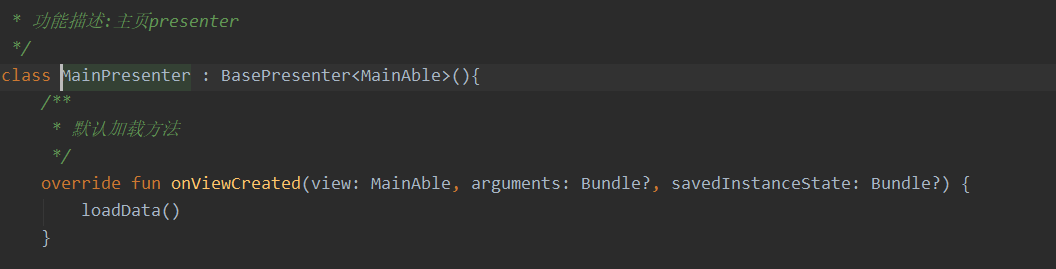


图4-9 MainPresenter类



图4-10 MainActivity类

需要实现一个新的界面，需要图4-8、图4-9、图4-10三个MVP文件，如果没用业务逻辑可以使用内核默认提供的EmptyPresenter代替Presenter。建议这三个文件放在同一个view的文件夹内，方便快速定位查看、修改。

图4-10中需要设置的三个地方是Presenter、ViewDataBinding、layoutId，这三部分就可以实现一个完整界面。

1. 基于框架层开发

基于该层开发所有配置、界面实现和基于内核层开发完全一样。区别只在与该层多了侧边栏管理类和奔溃管理类。如图4-11所示。

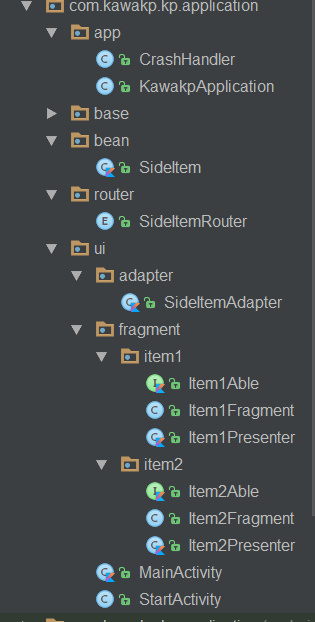


图4-11 侧边栏框架结构图

图4-11中SideItem、SideItemRouter、SideItemAdapter、MainActivity四个文件组成侧边栏框架。若需要添加一项侧边栏侧边，只需要实现一个类似于item1的MVP Fragment界面，然后在SideItemRouter配置图标、文字标题、Fragment就行。SideItemRouter配置文件如图4-12所示。

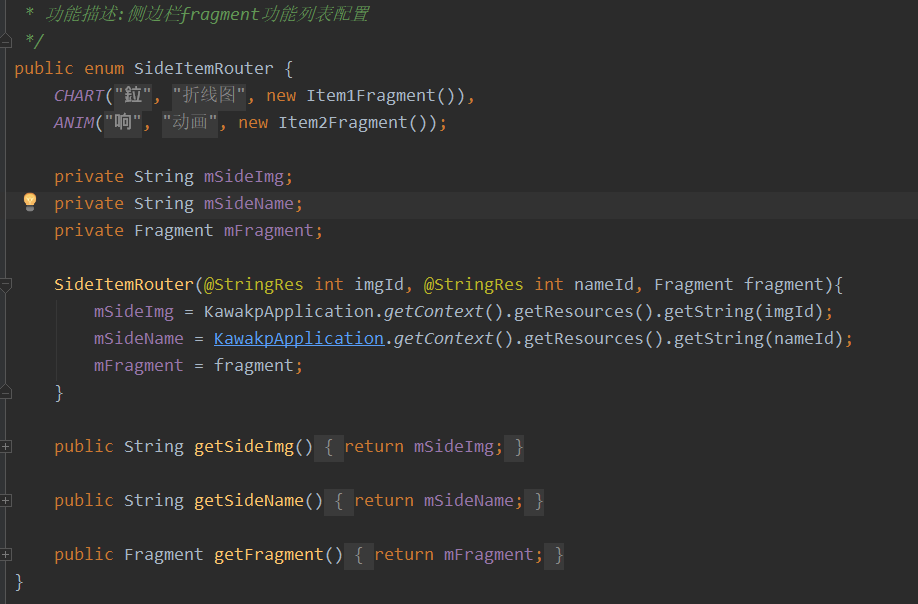


图4-12 SideItemRouter源码图

图4-12中CHART、ANIM就是折线图和动画两个侧边项，Item1Fragment和Item2Fragment是起对应的Fragment界面。