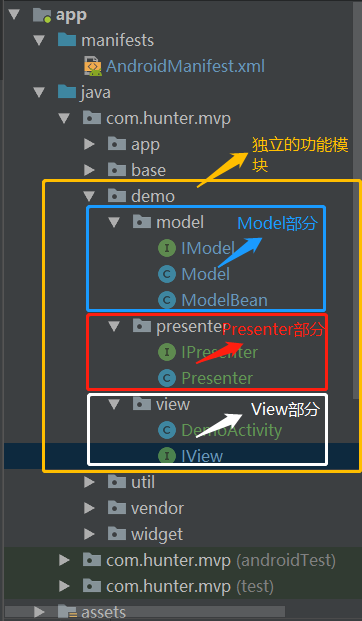
MVP Demo剖析

|  |  |  |
| --- | --- | --- |
| 武蛟 |  | 2019/01/07 |

### 目的

本文档通过剖析MVP Demo中的关键要素，来深入了解如何使用MVP框架

### 典型的MVP分包结构示意图



### 分包结构说明

1. demo。

一般是项目中围绕某个业务功能的一个模块，建议单独放在一个包中，以便后续考虑是否可以从项目中分离出去，独立成一个脱离项目的模块

1. model(角色**M**)

一般由bean和model组成，

bean是简单的POJO对象;

model是**P**用来从数据库或者网络中获取数据的，demo中用线程池来执行**M**对应的操作(分DiskIoPool,NetworkPool和MainThread).

1. presenter(角色**P**)

持有**M**和**V**的实例，通过**M**来拿到数据，交给**V**去渲染页面(实现**M**的回调方法，在回调方法中拿到数据交给**V**去渲染页面)

1. view(角色**V**)

在Activity或者Fragment实现**V**中的方法，**V**不包含业务逻辑，只负责将数据去渲染页面.

### MVP中接口方法的约束(建议供参考)

1. **M**中的方法的命名，可包含关键字insert，delete，update，query，方法通过回调方法来返回数据，回调方法的在**P**中具体实现，具体操作交给线程池来处理:
2. **P**中的方法命名，可包含关键字save，remove，update，get/load，实现**M**中的回调方法，在回调方法中将**M**返回的数据交给**V**，**P**通过getView()来获取**V**中定义好的接口方法
3. **V**中负得方法命名，可包含关键字remove，update，show/refresh，**V**只负责将数据渲染页面，避免Activity或Fragment臃肿

### MVP的流程(以P从M中获取数据在V显示为例).

