

Bison的博客

成功就是比你更天才的人，还比你更努力!

目录视图

个人资料



bsince

关注

发私信

访问：82次

积分：20

等级：

BLOG > 1

排名：千里之外

原创：2篇

转载：0篇

译文：0篇

评论：0条

文章搜索

文章分类

Android进阶

(2)

文章存档

2015年08月

(2)

阅读排行

Android MVP 探究与总结

(45)

Android MVP实战——MVP...

(40)

评论排行

Android MVP 探究与总结

(0)

Android MVP实战——MVP...

(0)

推荐文章

* HDFS如何检测并删除多余副本块

* Project Perfect让Swift在服务器端跑起来 - 让Perfect更Rails (五)

* 数据库性能优化之SQL语句优化

* Animation动画详解(七)——ObjectAnimator基本使用

* 机器学习系列(7)_机器学习路线图（附资料）

* 大数据三种典型云服务模式

Android MVP 探究与总结

标签：[android](#) [MVP](#)

2015-08-11 18:2845人阅读

分类：Android进阶 (1)

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录(?)

[+]

导读：在Android的开发过程中，Activity承担了大量的工作。如果对整个项目十分了解，并且在开发过程中有意识地抽离出了一点，还稍微好一点，但实际上我们经常会遇到这样一些情况：维护并迭代已有的商业项目（可能前几期并不是由你开发）;需求变动，Ctrl+到处查找？）...

如果按照MVC的项目结构来说，Activity应该是属于View这一层。而实质上，它既承担了View，更多地却包含一些Controller的职责，与维护来说极度不利，耦合度太高了

那么有没有什么解决方案呢？答案当然就是本文所讲述的——MVP了

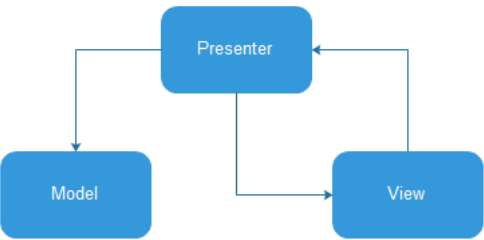
什么是MVP？

虽然说MVP已经出来很有一阵了，网上关于讲解这方面的文章也不少，但实际上真动手写起来时，还是有一些不顺畅。这没有更练，多练（重要的事情说三遍）。

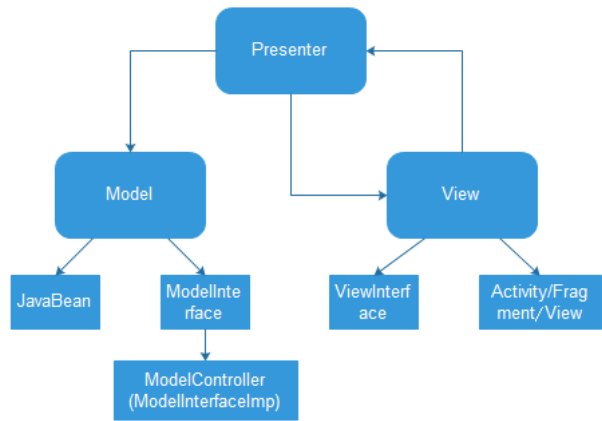
什么是MVP呢？

MVP是Model, View和Presenter的简称。是MVC模式的演化版。MVP模式把显示逻辑和从业务逻辑层中分离出来，理想替换过程中，可以实现相完全相同的逻辑代码。

下面看一张图



从上图可以看出，Presenter层成了Model与View之间的桥梁，Model与View之间不再直接进行交互。而Presenter持有Model逻辑地处理。将这个模式代入到Android中，我们可以将他丰富成这样(如下图)



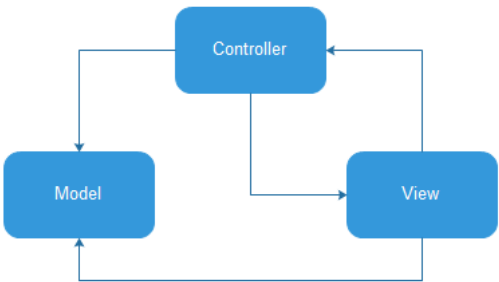
- Model层：包含必要的JavaBean对象，这个不用多说。其次，需要抽离出一个ModelInterface出来，以用于对数据流的需要一个实现类。
- View层：同样需要抽离出来一个ViewInterface，用于实现UI的交互及刷新。而我们的Activity，Fragment或者View其实实现类。所以一般在Activity或fragment中实现ViewInterface接口。
- Presenter层。presenter英文意思是主持人，在MVP中它是负责逻辑处理并统筹View与Model的一个层。一般来说View情况，当然这只是一般情况，并不排除复杂界面中Presenter有多个的情景。上面说到Presenter持有View与Model的一个有的是ViewInterface与ModelInterface的引用。

MVP与MVC的区别？

让我们先回忆一下MVC的结构

- M（Model）实体层
- V（View）视图层
- C（Controller）控制层

相信对MVC，大家都是非常熟悉了，不再多言，一图以蔽之



通过与MVP结构图对比着看可以发现：

在MVC结构中，Model是可以直接被View操作的。其实这就是MVP与MVC最大的一个区别。

MVP有什么用？

或者可以这样问MVP有什么好处？为什么要使用MVP？

其实通过上面的几张图就不难看出

- MVP抽离了显示层和逻辑层，降低了耦合
- MVP结构层次更清晰，可以轻易实现View替换却不带动逻辑层变动。
- MVP逻辑层是针对接口编程，他更方便于进行单元测试。

其实，说这么多，都是因为显示层与逻辑层分离带来地便利。

如何使用MVP？

都说MVP抽离了层结构，降低了耦合度，那么它到离是如何抽离的？当然是通过接口了。

Presenter中通过对接口的操作，可以完全不关心View层与Model层的具体实现，只需要处理对应逻辑就可以了。而在View层中Fragment中，只需要持有一个Presenter的对象，以完成界面交互。这样咱们的View层就变得非常地薄了。所以，总结了一下几个方向思考：

- **创建实体JavaBean对象.**这个跟以往地MVC没有多少区别，不多说。
- **抽离出View的接口，即ViewInterface.**如何抽离？View层主要负责与用户地交互工作。那么落到具体上其实就是对UI的需要把我们原来项目中的事件全部抽离，如：点击，长按，滑动....,而所谓地UI控制，即提供对View的操作方法，如进度条与消失等等。
- ****抽离Model的接口，即ModelInterface.****ModelInterface需要有什么？说白了，其实就是对数据地读取与写入。比如表集合，或者是从本地文件中读取一段字符串等等。在Presenter因为操作接口地缘故，所以他根本不用关心数据地获取与里分析一下业务逻辑，将其抽离成接口
- **实现Presenter.**这个不难，它持有ModelInterface与ViewInterface,通过操作它们实现所有的逻辑处理。最后需要暴露出
- **实现ModelInterface与ViewInterface.** 这个肯定是必须地，ModelInterface的实现不用多说，而ViewInterface的实现或Fragment中调用Presenter暴露出来地方法而已。

以上,MVP理论上来说可以实现在多人协作项目中，让写UI的人与写逻辑的人分工而行，也更方便于在开发过程中进行单元测试。地实际应用我们放在下章再说，我们一起来一步步实现一个MVP模式的Android应用。

顶 0 踩 0

- [下一篇](#) Android MVP实战——MVP在Android中的简单应用

我的同类文章

Android进阶（ 1 ）		
•	Android MVP实战——MVP在Androi...	2015-08-12 阅读 40

主题推荐 android mvp c语言 ui mvc javabean bt x

猜你在找

威哥全套Android开发课程【基础…	MVC——Android App的设计架构MV…
基于PHP面向对象的自定义MVC框…	625 android开发mvp模式类似swi…
Android开源项目实践之UI篇	Leveraging NET Components and…
老郭全套iOS开发课程【UI技术】	《UI层的设计模式从ScriptCode …
Android底层技术：Linux驱动框…	Android中单元测试 Robolectric…

查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题	Hadoop	AWS	移动游戏	Java	Android	iOS	Swift	智能硬件	Docker	OpenStack	VP	
IE10	Eclipse	CRM	JavaScript	数据库	Ubuntu	NFC	WAP	jQuery	BI	HTML5	Spring	Apache
HTML	SDK	IIS	Fedora	XML	LBS	Unity	Splashtop	UML	components	Windows Mobile	Rails	
Cassandra	CloudStack	FTC	coremail	OPhone	CouchBase	云计算	iOS6	Rackspace	Web App	S		
Compuware	大数据	aptech	Perl	Tornado	Ruby	Hibernate	ThinkPHP	HBase	Pure	Solr	Android	
Cloud Foundry	Redis	Scala	Django	Bootstrap								