

MVP for Android

作者: [jzw](#)

原文链接: <https://ld246.com/article/1477446774376>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```
<p>Made with Remarkable! <link rel="stylesheet" href="http://cdnjs.cloudflare.com/ajax/libs/highlight.js/8.1/styles/github.min.css" /> </p>
<style type="text/css"><!--
  body,table tr{background-color:#fff}table tr td,table tr th{border:1px solid #ccc;text-align:left
padding:6px 13px;margin:0}pre code,table,table tr{padding:0}hr,pre code{background:0 0}bo
y{font:16px Helvetica,Arial,sans-serif;line-height:1.4;color:#333;word-wrap:break-word;paddin
:10px 15px}strong,table tr th{font-weight:700}h1{font-size:2em;margin:.67em 0;text-align:cent
r}h2{font-size:1.75em}h3{font-size:1.5em}h4{font-size:1.25em}h1,h2,h3,h4,h5,h6{font-weight:7
0;position:relative;margin-top:15px;margin-bottom:15px;line-height:1.1}h1,h2{border-bottom
1px solid #eee}hr{height:0;margin:15px 0;overflow:hidden;border:0;border-bottom:1px solid
ddd}a{color:#4183C4}a.absent{color:#c00}ol,ul{padding-left:15px;margin-left:5px}ol{list-style-t
pe:lower-roman}table tr{border-top:1px solid #ccc;margin:0}table tr:nth-child(2n){background
color:#aaa}table tr td :first-child,table tr th :first-child{margin-top:0}table tr td:last-child,table t
h :last-child{margin-bottom:0}img{max-width:100%}blockquote{padding:0 15px;border-left:
px solid #ccc}code,tt{margin:0 2px;padding:0 5px;white-space:nowrap;border:1px solid #eaea
a;background-color:#f8f8f8;border-radius:3px}pre code{margin:0;white-space:pre;border:none
.highlight pre,pre{background-color:#f8f8f8;border:1px solid #ccc;font-size:13px;line-height:1
px;overflow:auto;padding:6px 10px;border-radius:3px}
--></style>
```

```
<h1 id="mvp-for-android">MVP for Android</h1>
<p>受够了一个文件两千行代码，找一个修改点需要半天，修改逻辑思前想后半天。随着代码复杂，
辑性和耦合性要求提升，该用一些框架模式了。在Android中mvp是一个不错的选择。mvvc太过超
。mvc与下雨天的Android不是很配。接下来介绍一下mvp。</p>
<h2 id="mvp">MVP</h2>
<blockquote>
<p>所谓MVP(Model-View-Presenter)模式。是将APP的结构分为三层</p>
</blockquote>
<h3 id="view-ui">view - UI显示层</h3>
<ol>
<li>UI交互： 监听，点击效果和UI设计属性</li>
<li>UI被改变： 被presenter控制 UI改变数据或者状态属性</li>
<li>UI通知： ， 通知presenter 处理事务 <br /> 注意. View层不存储数据，不与Model层交互。 </li>
</ol>
<h3 id="presenter-">presenter - 逻辑处理层</h3>
<ol>
<li>获取model层的数据</li>
<li>处理逻辑业务</li>
<li>响应UI层的命令</li>
<li>发布修改UI层命令,接受类似Eventbus的广播订阅。 </li>
<li>存储临时相关数据 <br /> 注意. Presenter内不出现View引用。 </li>
</ol>
<h3 id="model-">model - 数据层</h3>
<blockquote>
<p>在看谷歌给的例子的时候是一个本地数据，而且数据还在测试包里面看的头都大了。</p>
</blockquote>
<ol>
<li>从网络，数据库，文件，传感器，第三方等数据源 <code> 读写数据 </code> 。 </li>
<li>对数据的临时存储,管理，协调上层数据请求 <br /> 3 . 将数据转化成好用的格式（如： bean Arr
ylist map等） </li>
</ol>
```

<p>文不如表表不如图直接看盗来的图。model层是如此的清晰。</p>

<p></p>

<p>能力有限待续……</p>

<script type="text/javascript" src="http://cdnjs.cloudflare.com/ajax/libs/highlight.js/8.1/highlight.min.js">// <![CDATA[

//]]> </script>

<script type="text/javascript">// <![CDATA[
hljs.initHighlightingOnLoad();

//]]> </script>

<script type="text/javascript" src="https://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS-MML_HTMLorMML">// <![CDATA[

//]]> </script>

<script type="text/javascript">// <![CDATA[
MathJax.Hub.Config({"showProcessingMessages" : false,"messageStyle" : "none","tex2jax": {
nlineMath: [["\$", "\$"]] });

//]]> </script>