

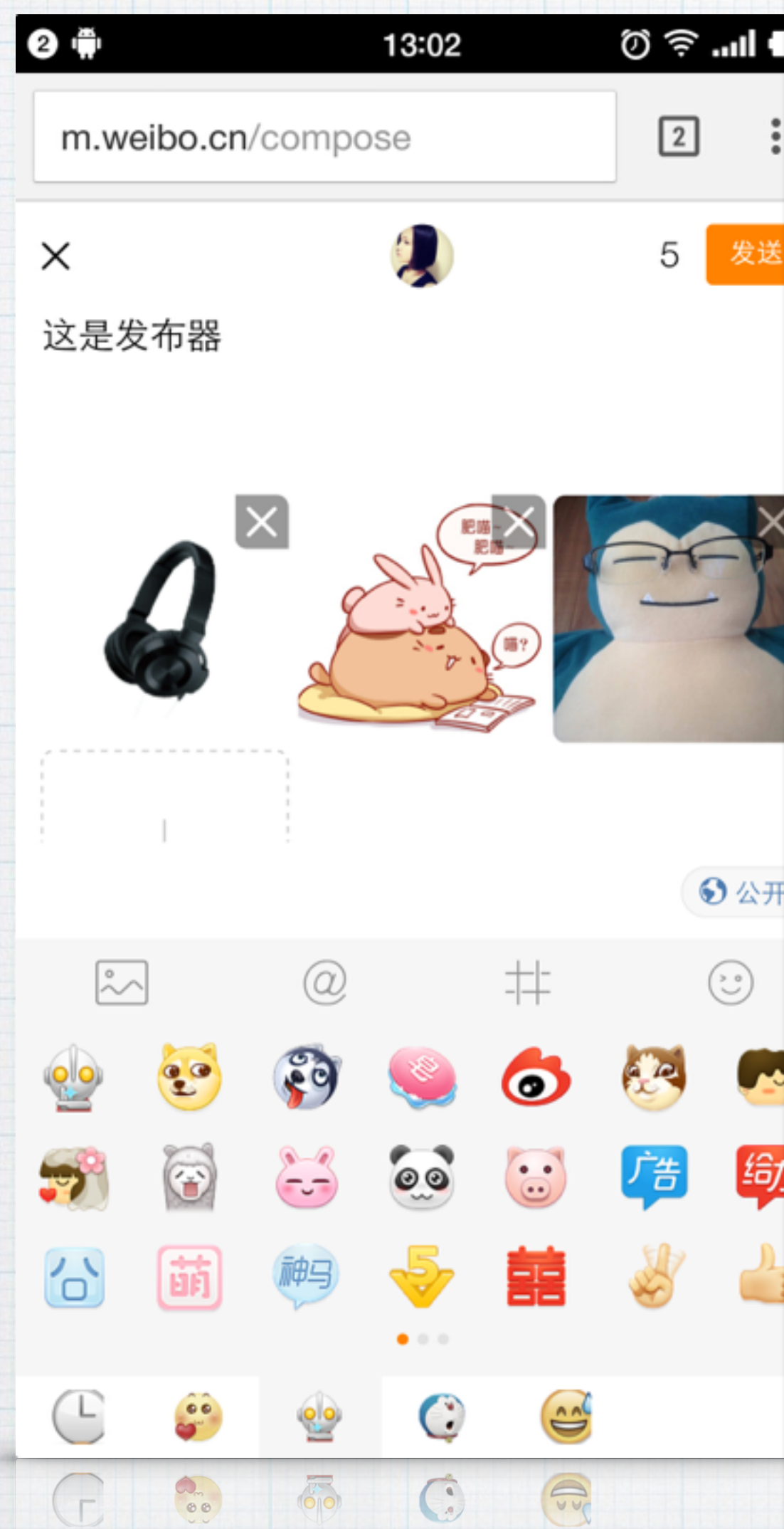
基于 MVVM 的微博前端优化方案



韦元悦

@韦字只念第二声

2017.5 微软大厦



- 无限滚动优化方案
- 拖放排序解决方案
- 通用模板组件优化方案

Part 1

无限滚动优化方案



安全 <https://www.zhihu.com/question/56236151/answer/148776856>

知乎新版的前端技术架构是什么？

1、流畅

首页timeline里答案展开卡飞。即使是已经缓冲的答案，折叠再展开也一样。

答案里倒是不卡，但遇到喜欢讲故事的，翻页给你翻到抓狂。敢不敢在左边 支持/反对 按钮下面，增加一个上下跳转？向上到本答案开头，向下到评论按钮即可。

别告诉我能用jk，不是程序猿的有几个知道jk？

另外，显示内容多了后（几十屏），所有操作都会开始卡。这点知乎和**微博**都没能避免。我就没看懂了，你在内存中只保留最近一两页，其他的卡片全部缓存起来不行么？进度条长度位置计算只需要一个卡片长度数组，我就问你这怎么会卡？

怕和浏览器滚动不好集成，那么做成滚动切页面那种模式切卡片不会么？卡片不足一页就多凑两张，超过一页就交给浏览器滚动，然后缓存前后卡片不行么？切换动画改成滚动风格，边缘滚动一次移动一格卡片，和连续滚动能有多大区别？而且这样卡片自动对齐屏幕，阅读体验好太多了。你乎移动版都会这么做，桌面版多凑几张卡片拼足一屏就那么难么？

感情前端做久了，native的那一套都忘光了么？

哦对了现在的前端根本不懂**native**。你们这些面向浏览器编程的人啊，是不会明白我们这种用32位程序做GB为单位bit为精度的数据处理的苦的。

▲ 78



● 54 条评论

➤ 分享

★ 收藏

♥ 感谢



收起 ^

▼ 87



● 24 条评论

➤ 分享

★ 收藏

♥ 感谢



收起 v

浏览器滚动不好集成，那么做成滚动切页面那种模式，过一页就交给浏览器滚动，然后缓存前后卡片不行么？切换动画一格卡片，和连续滚动有很大区别？而且这样卡片自动对齐屏幕都会这么做，桌面端做久了，native的滚动就那么容易么？



了现在的前端根本不懂**native**。你们这些面向浏览器编程的人啊，做GB为单位bit为精度的数据处理的苦的。



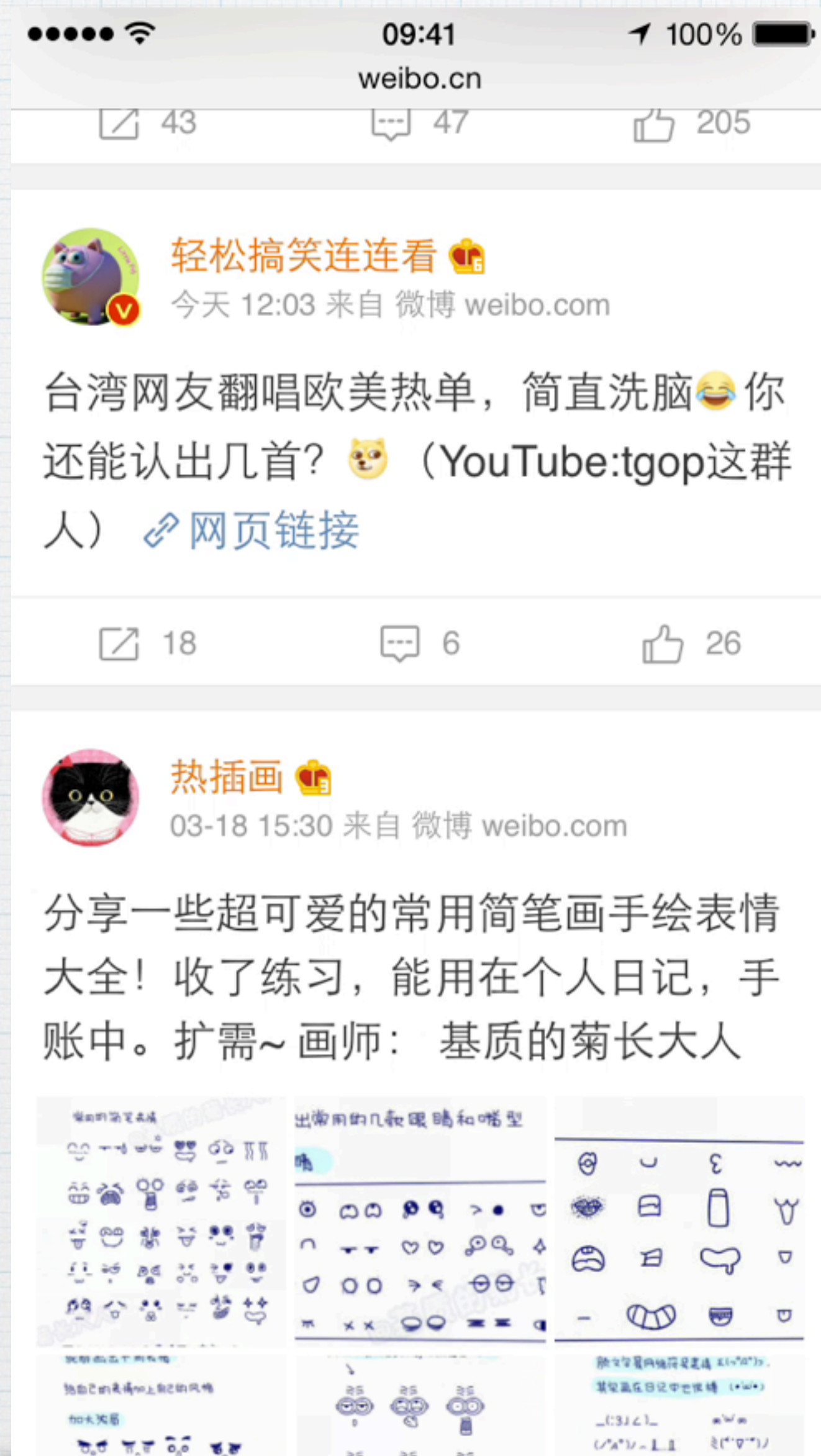
54 条评论

分享

★ 收藏

♥ 感谢





旧版面对的问题：

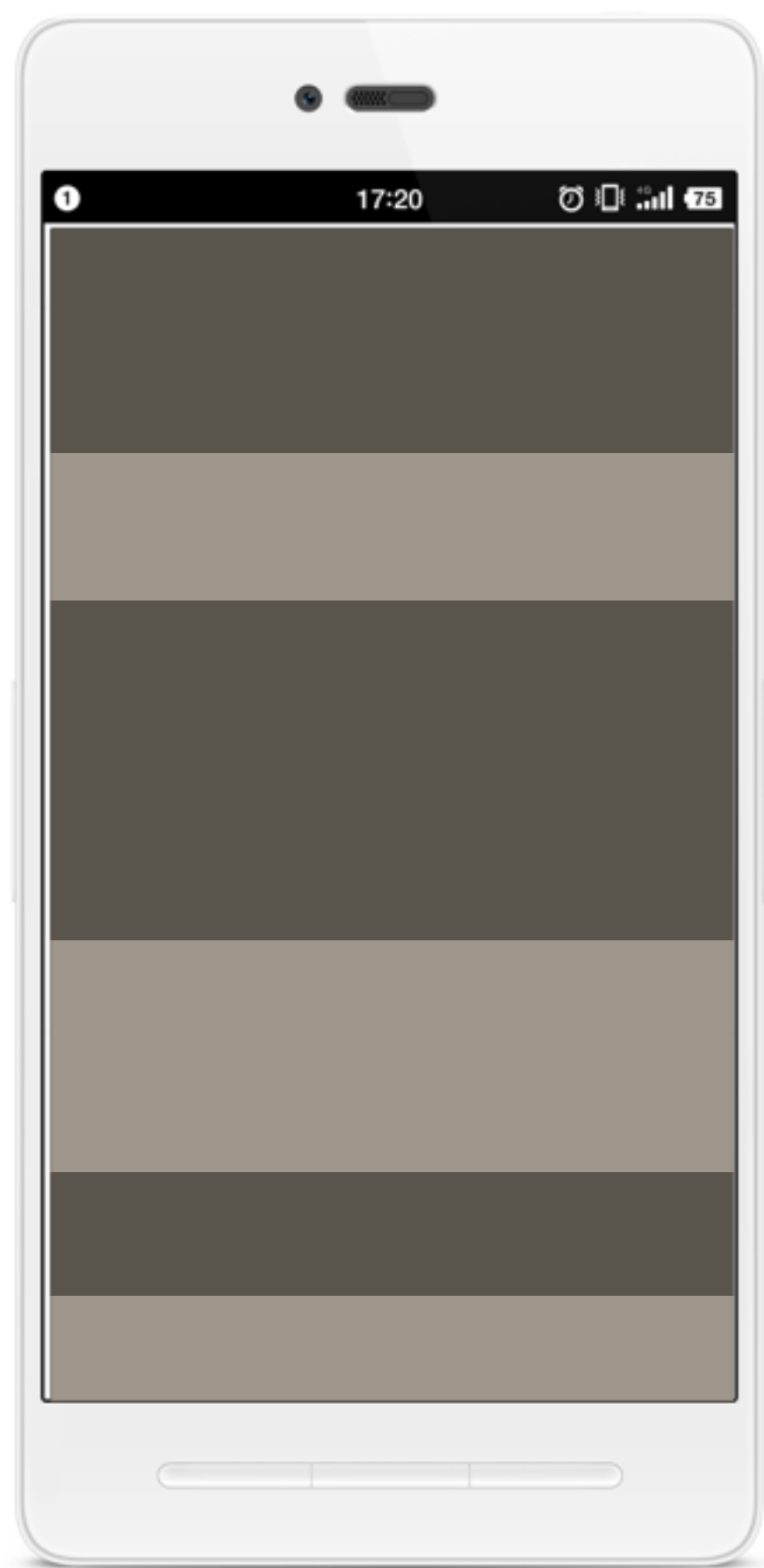
- 手动点击翻页，破坏连贯性
- DOM过多：卡顿、白屏、崩溃
- 跳转页面后不能返回上一次阅读位置

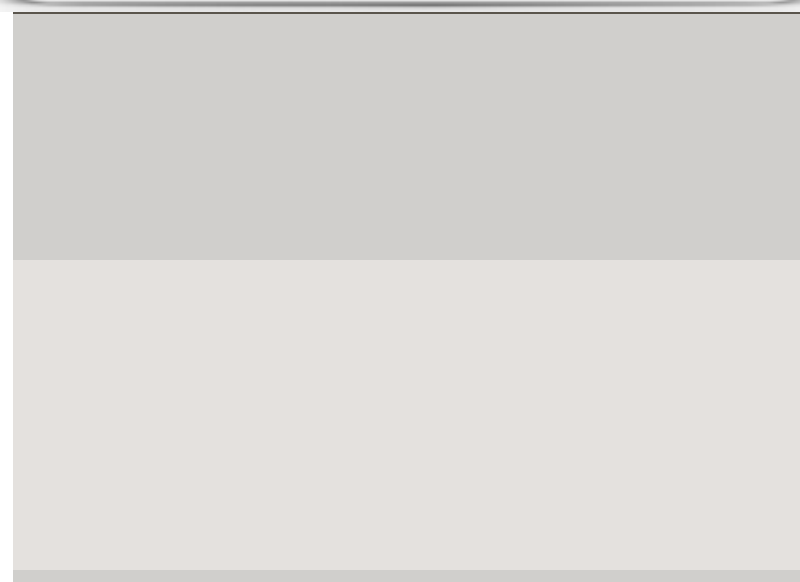
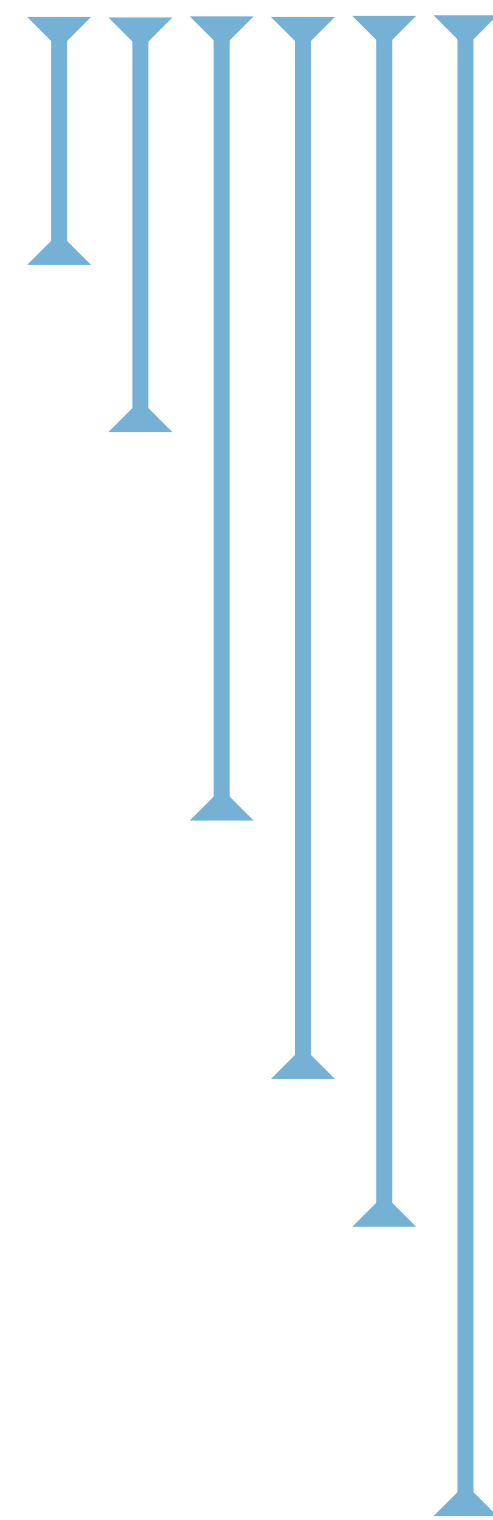
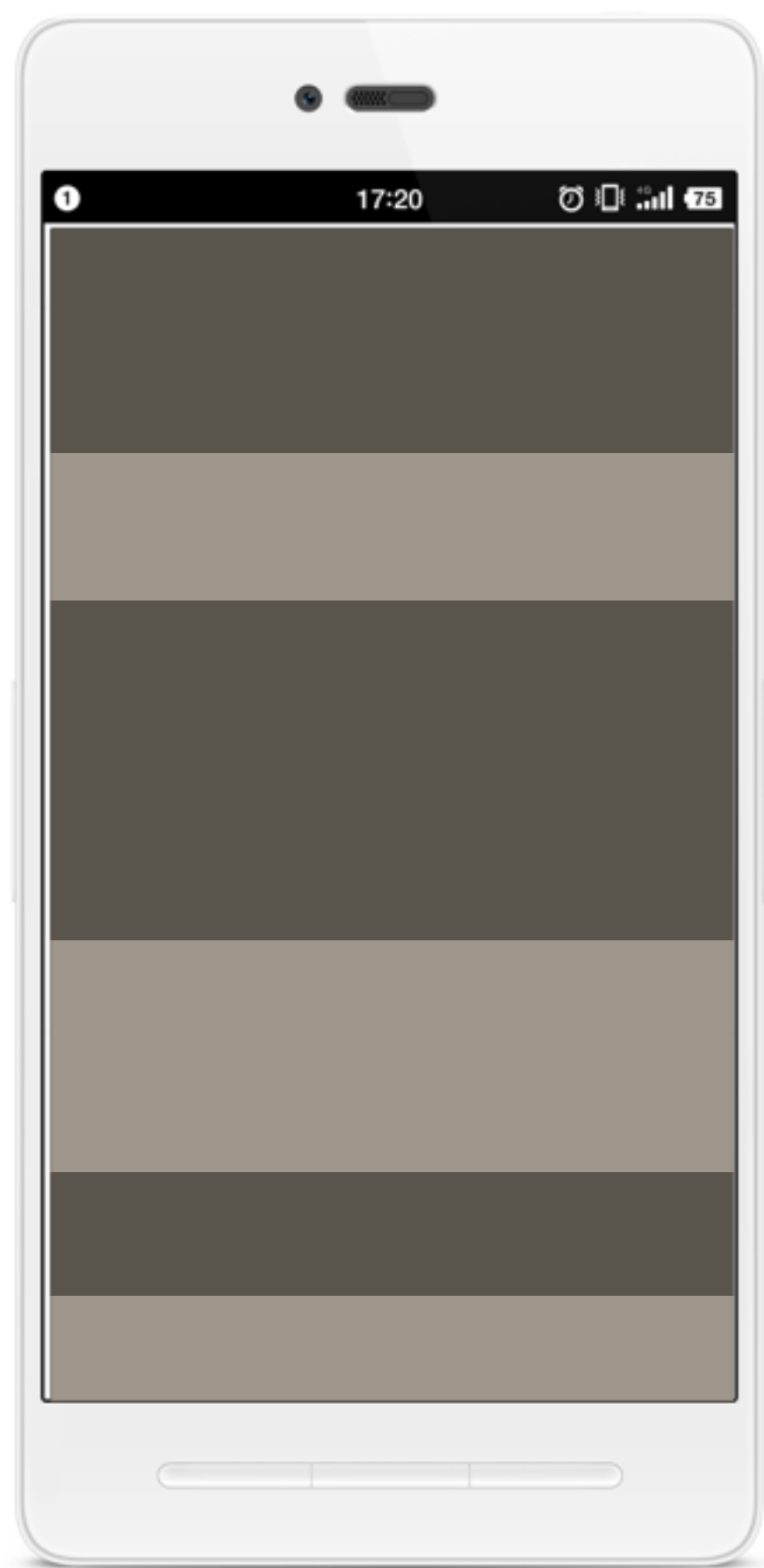
Android: RecyclerView

iOS: UITableViewCell

Vue.js + SPA

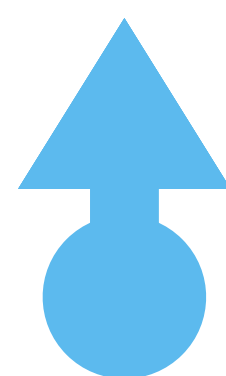
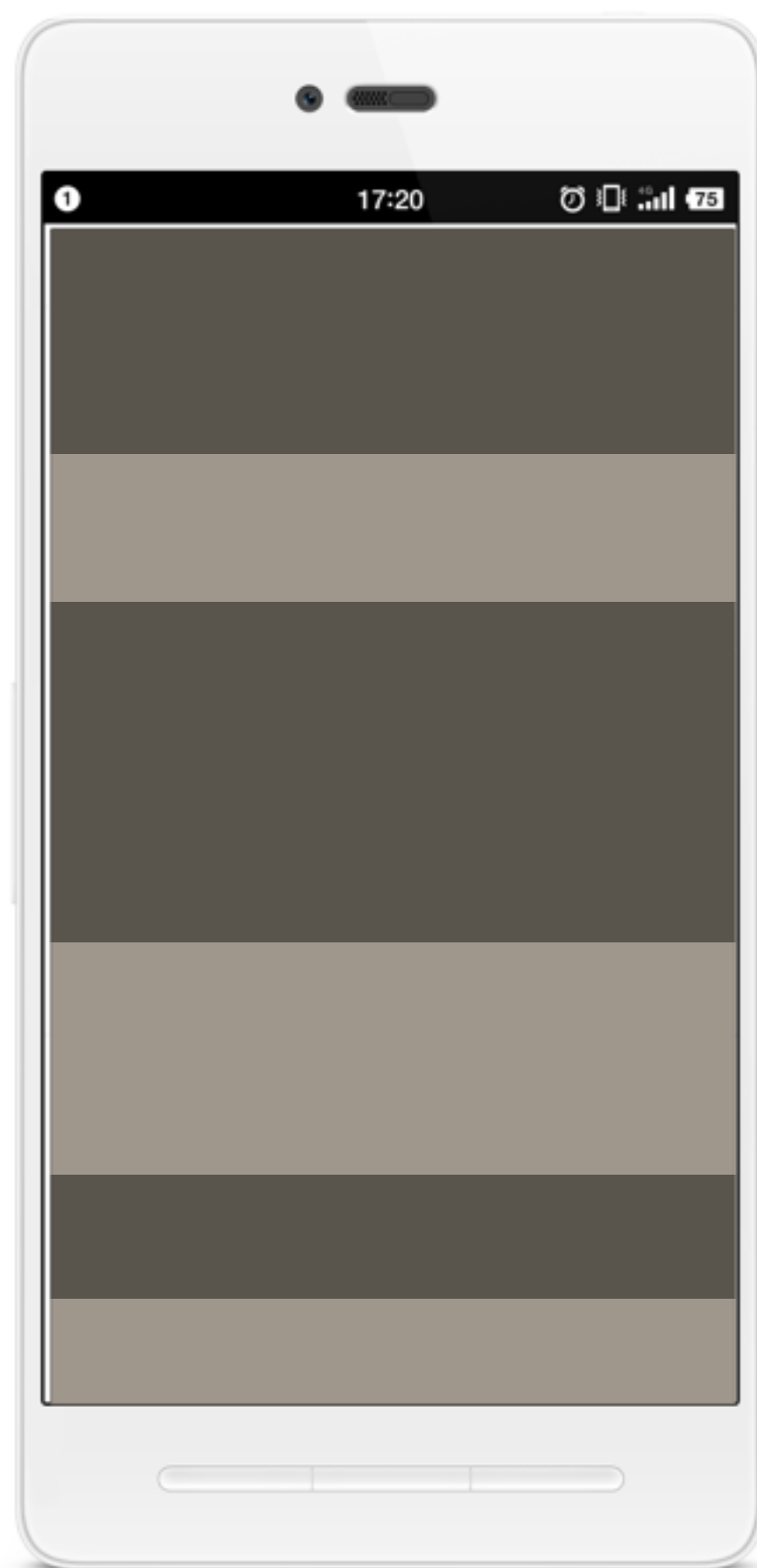
更多的可能性，更高的效率，更专注于 Model 层

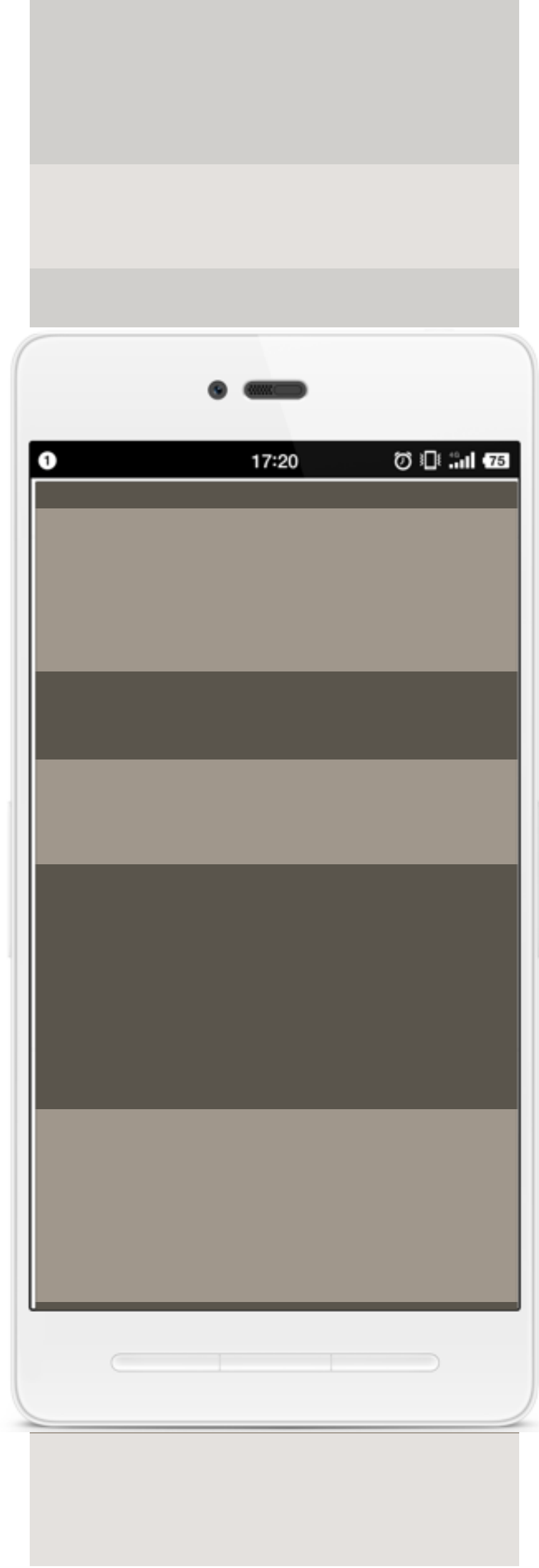




CurrentList

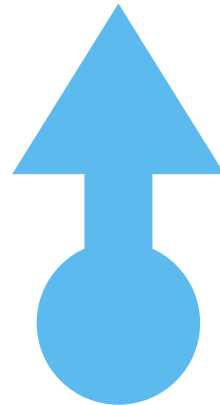
WholeList



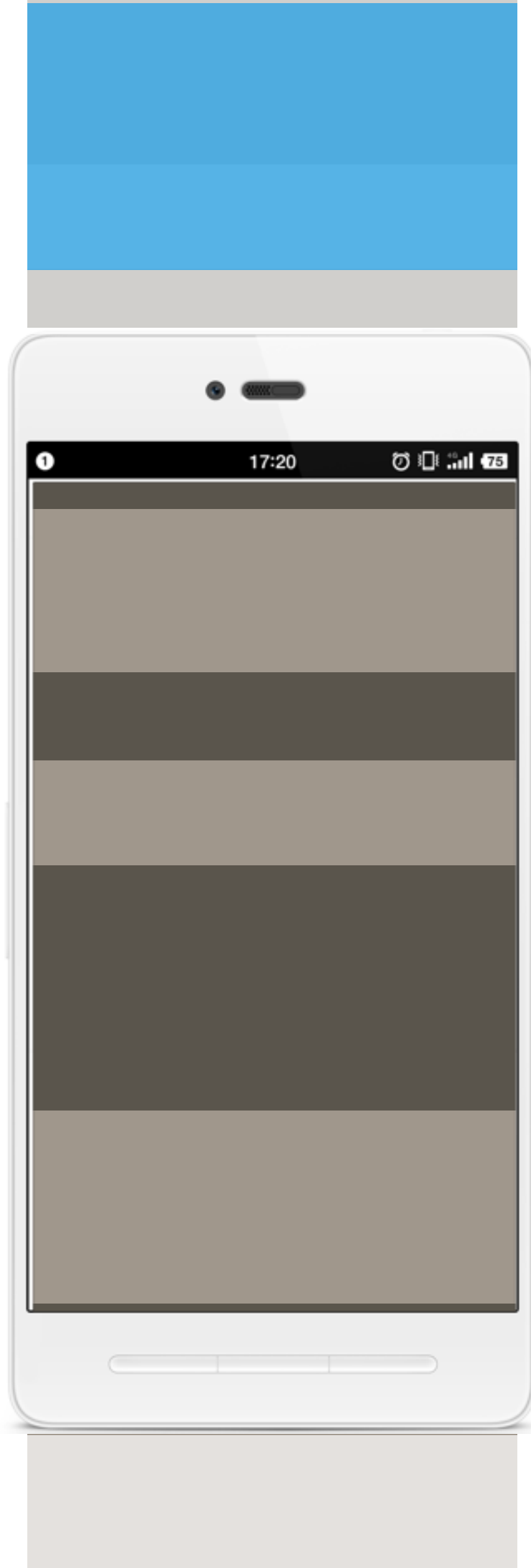


CurrentList

WholeList

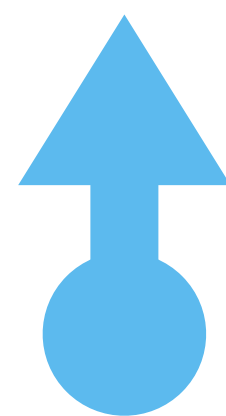


padding-top

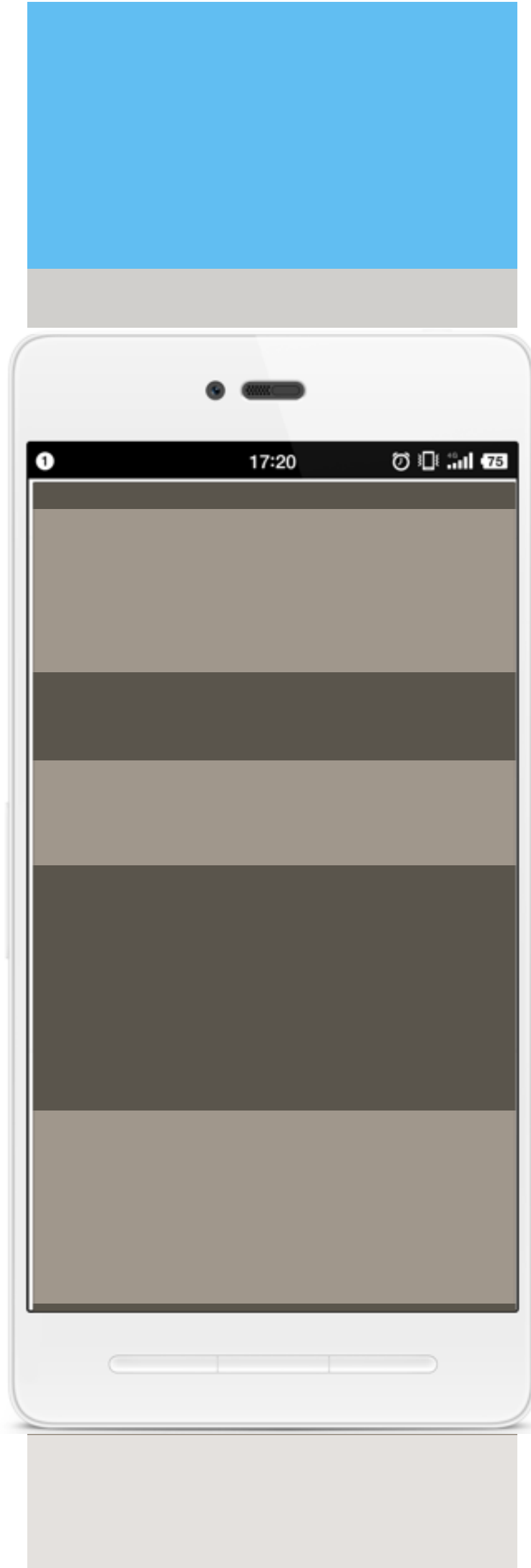


CurrentList

WholeList

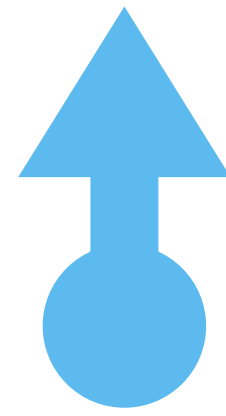


padding-top

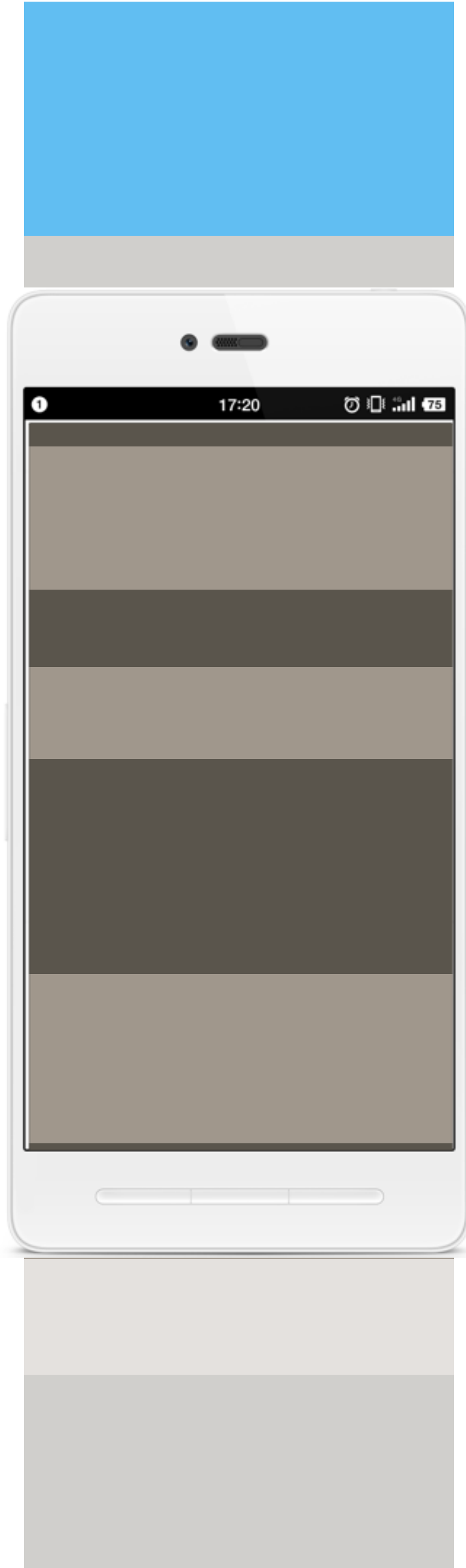


CurrentList

WholeList

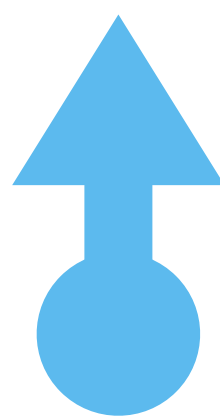


padding-top



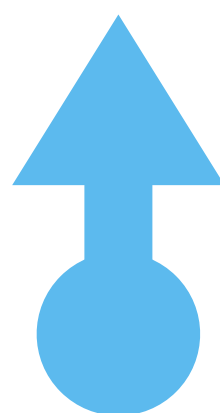
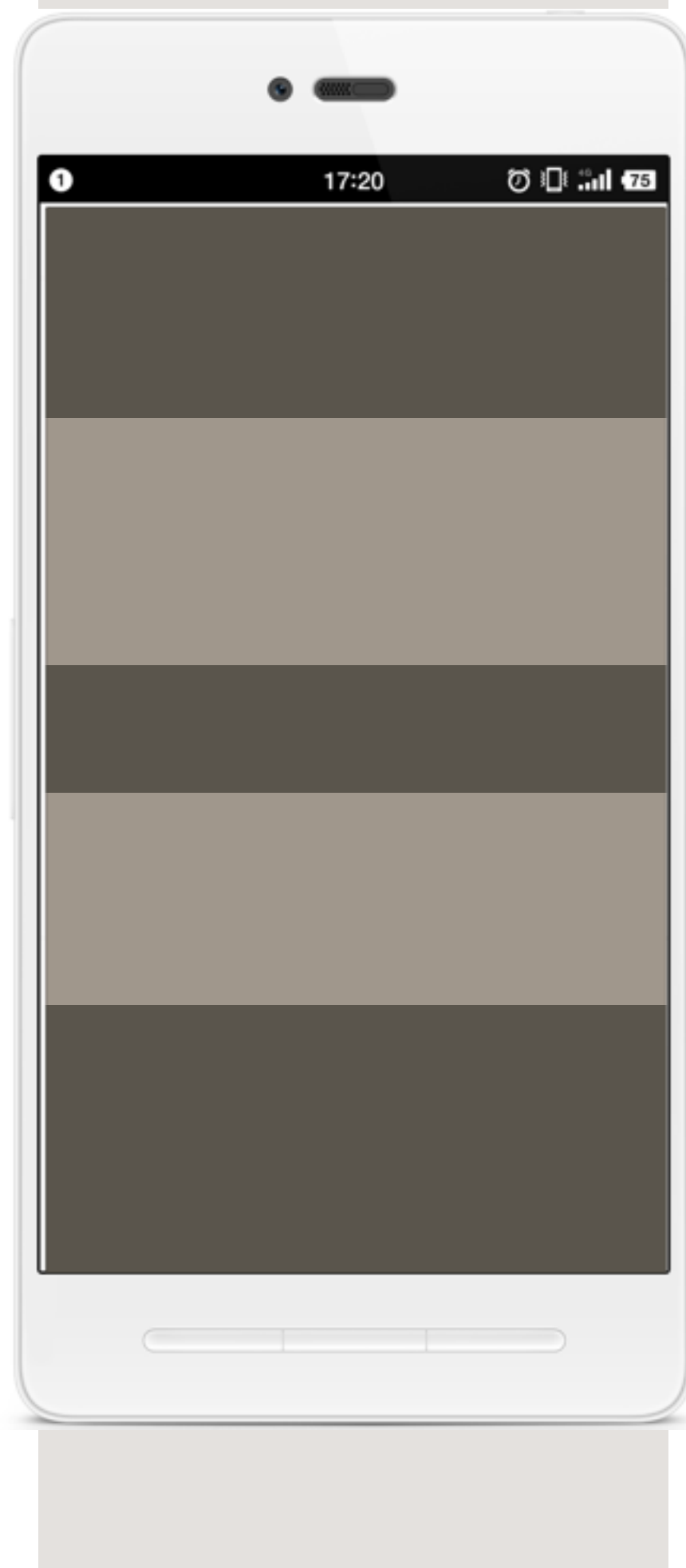
CurrentList

WholeList

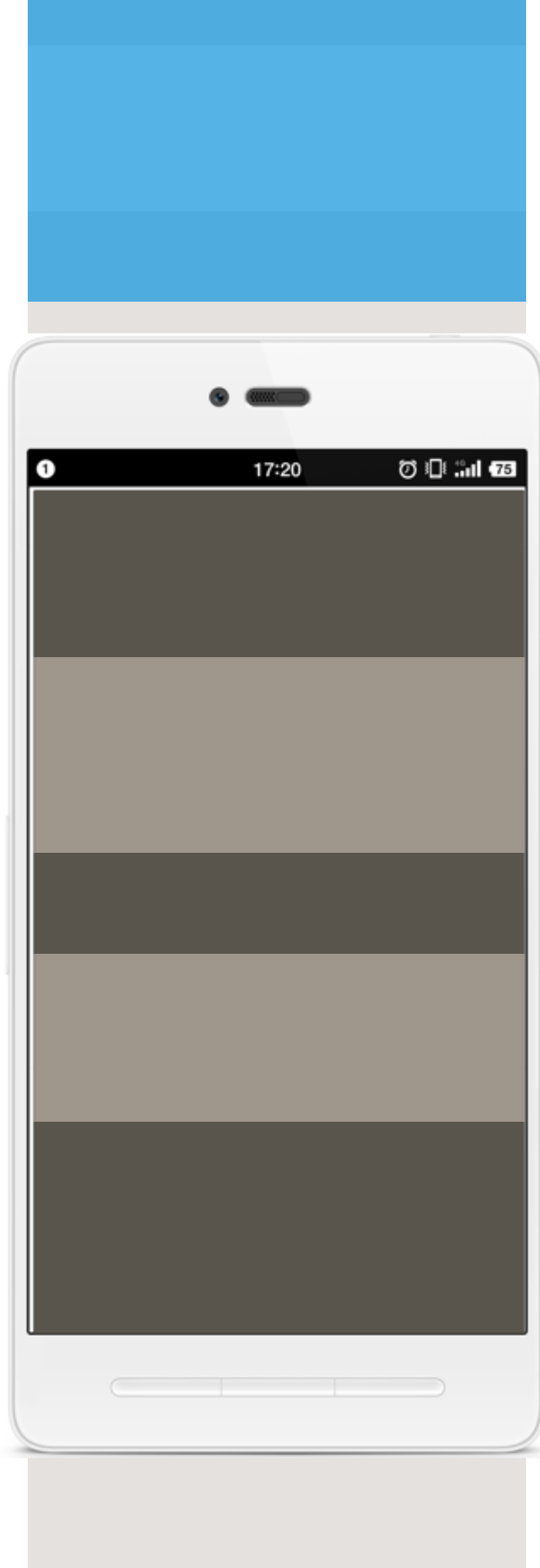


CurrentList

WholeList



padding-top

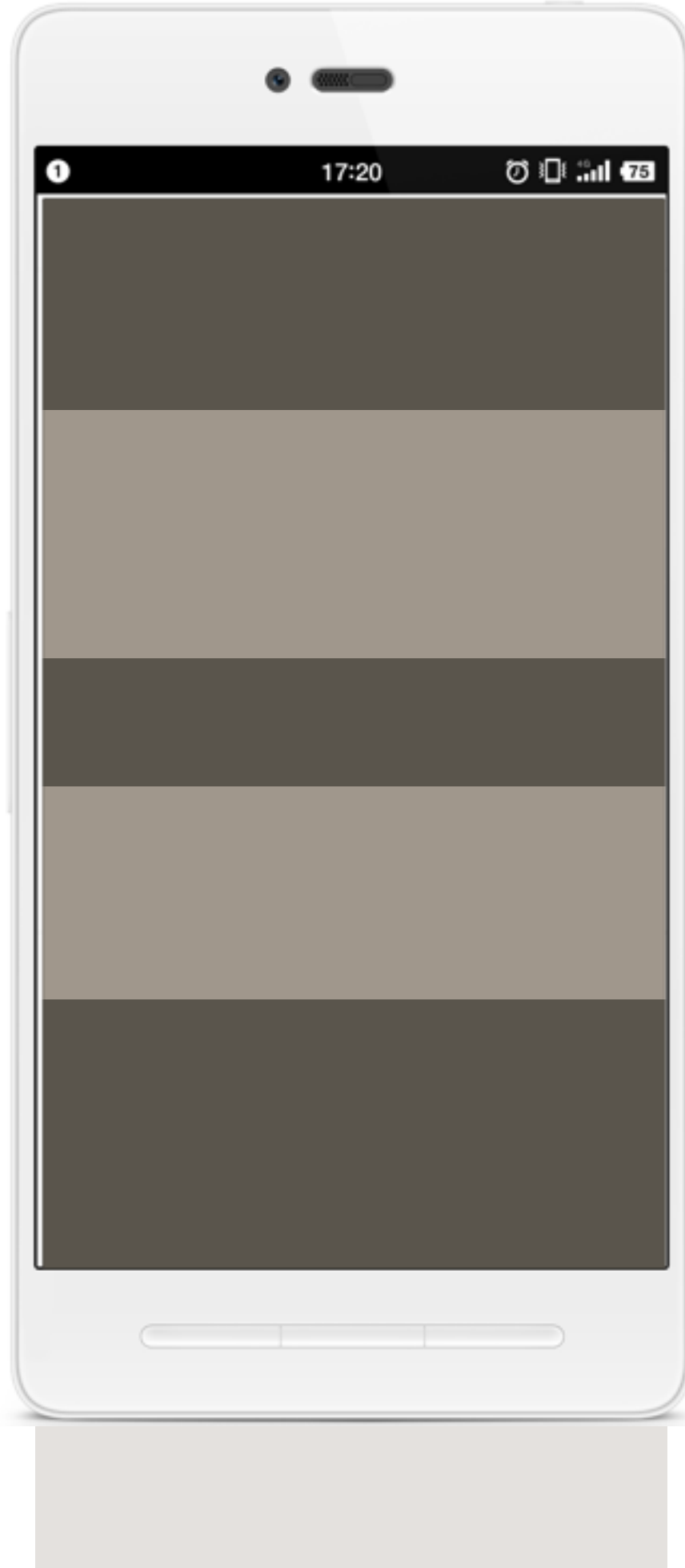


CurrentList

WholeList

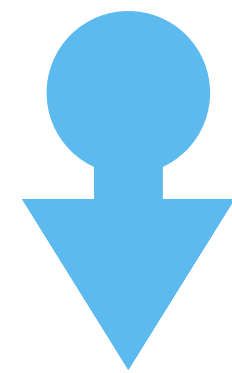


padding-top



CurrentList

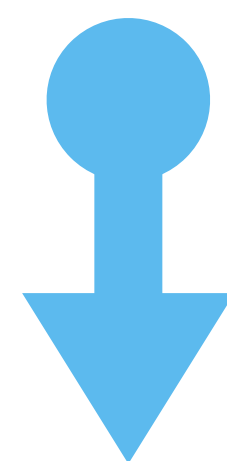
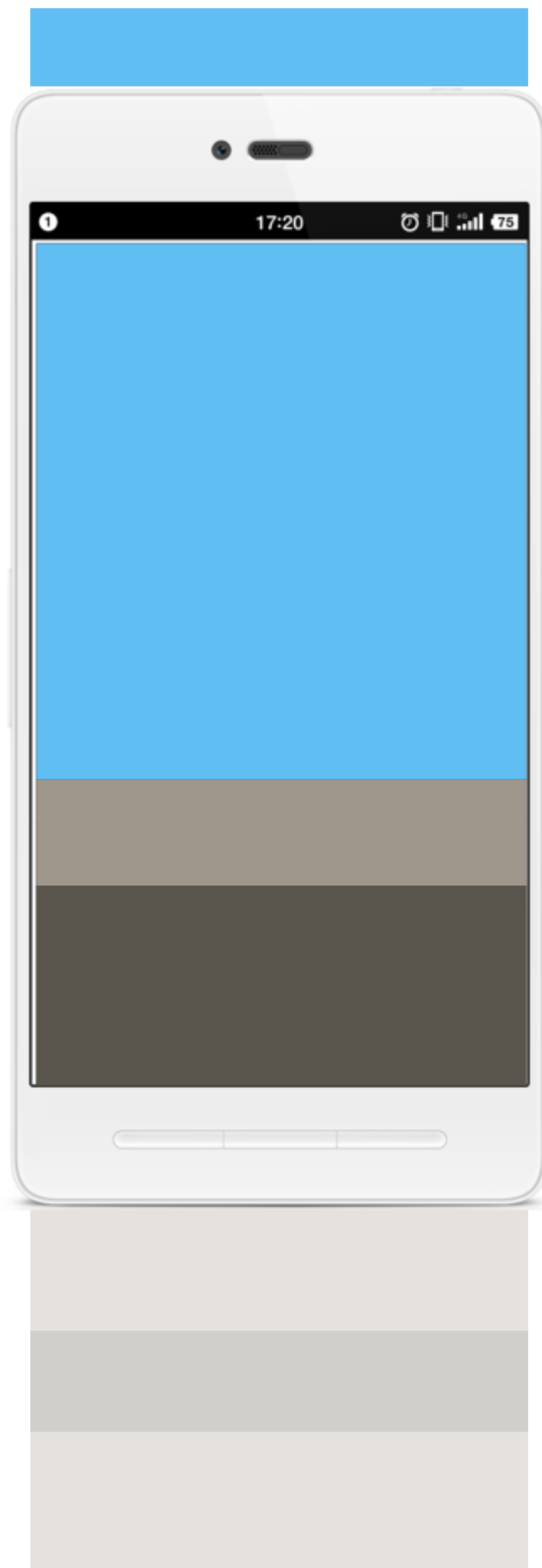
WholeList



CurrentList

WholeList

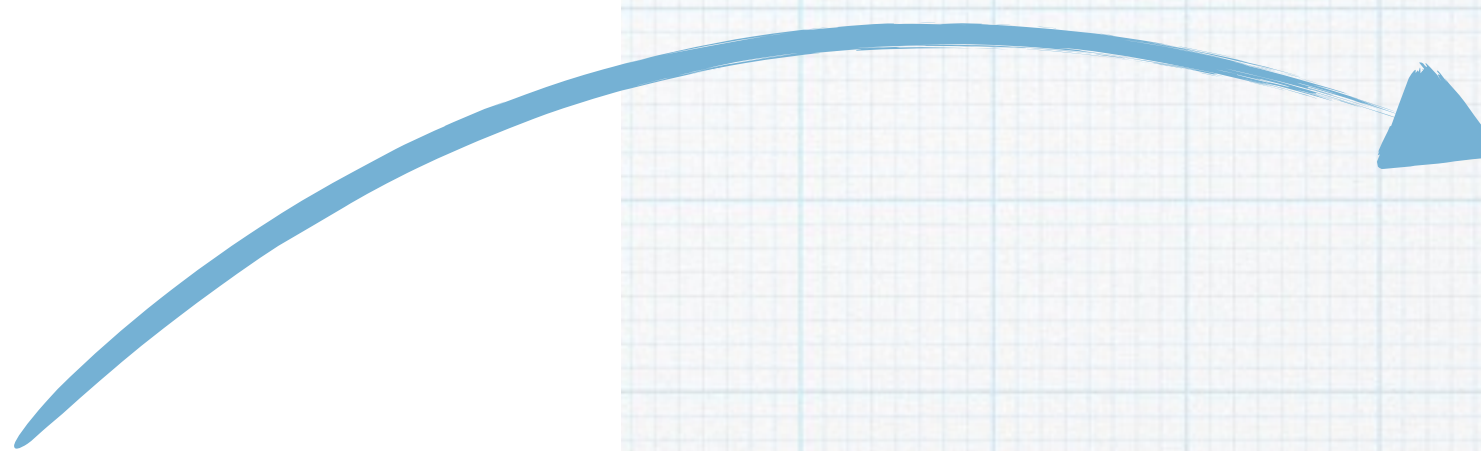
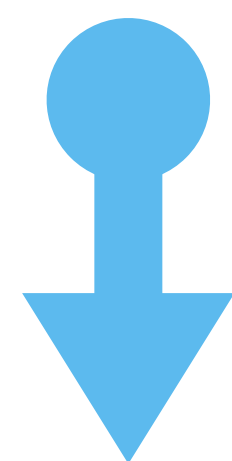
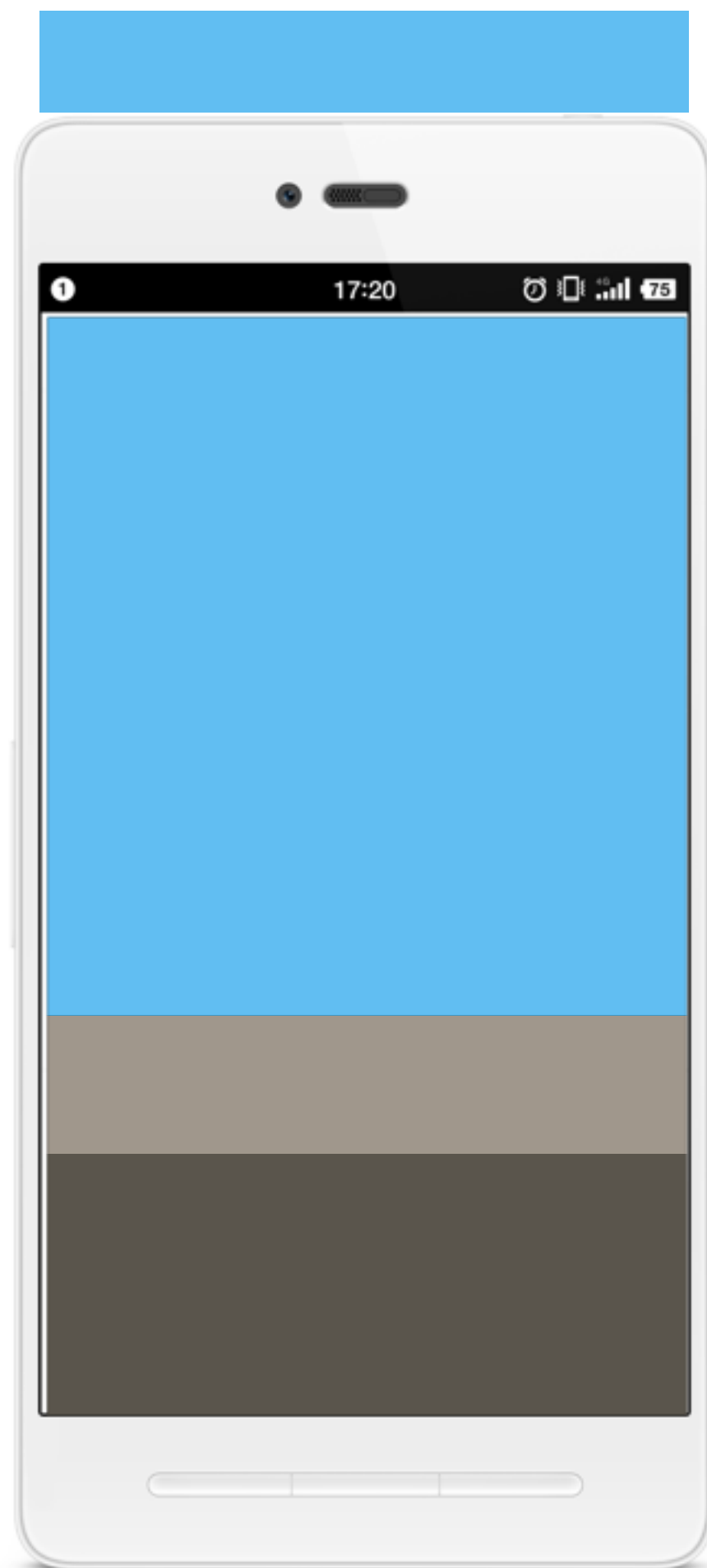
padding-top



CurrentList

WholeList

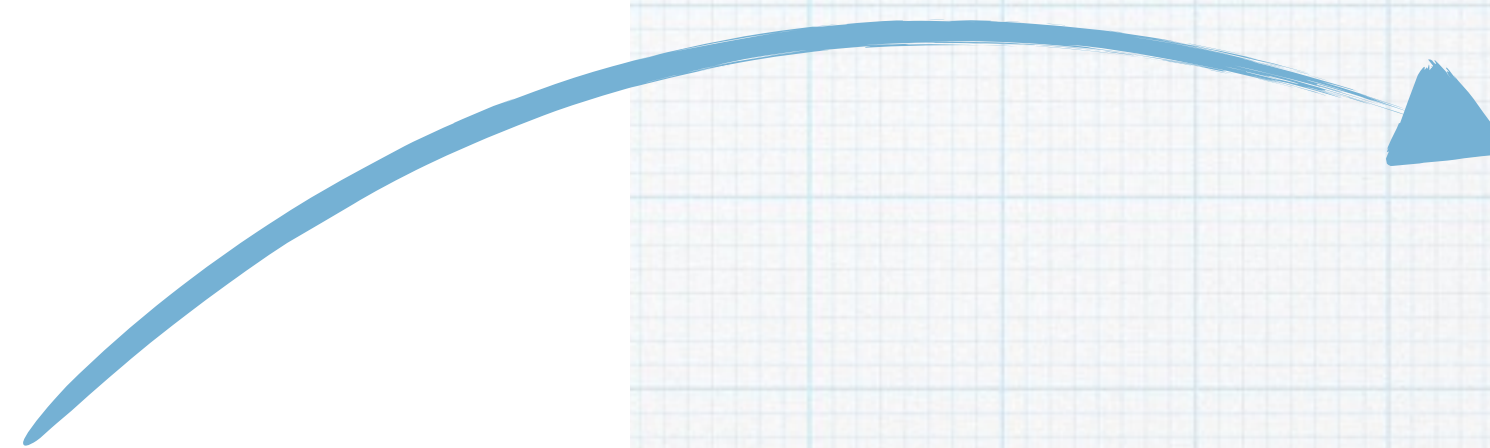
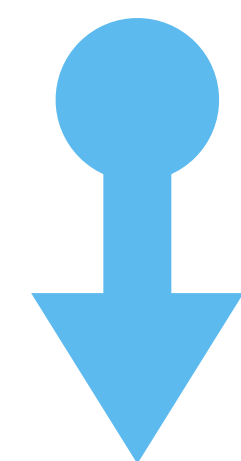
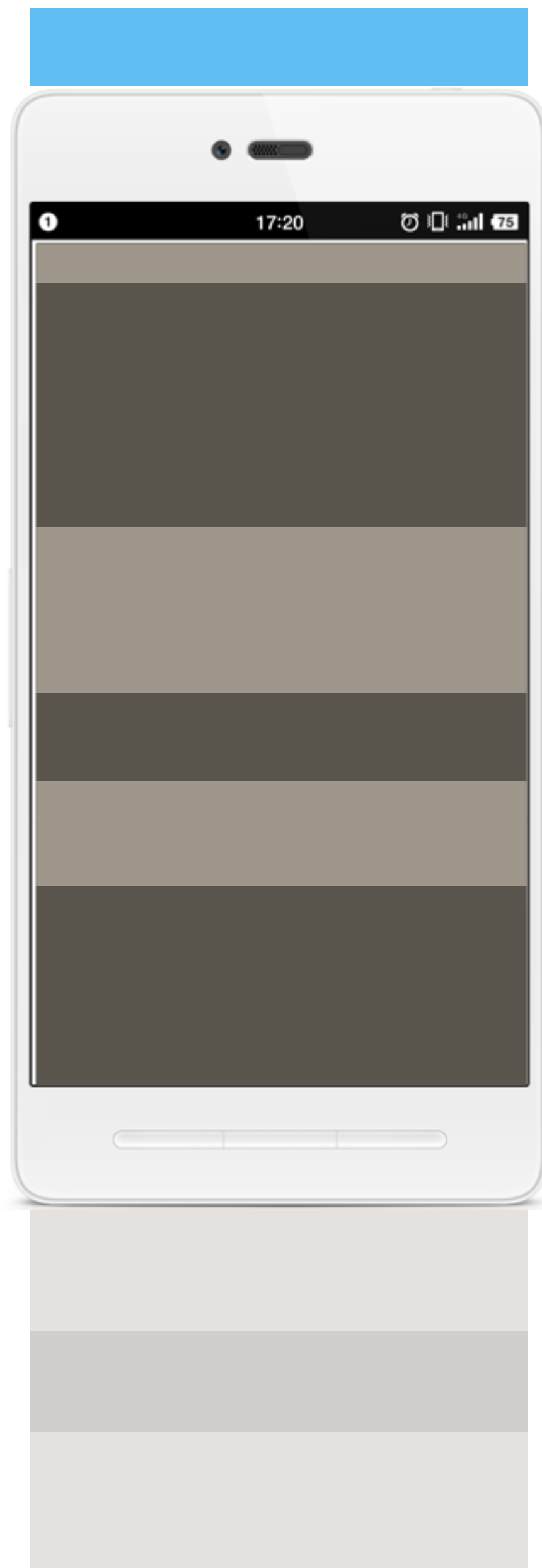
padding-top



CurrentList

WholeList

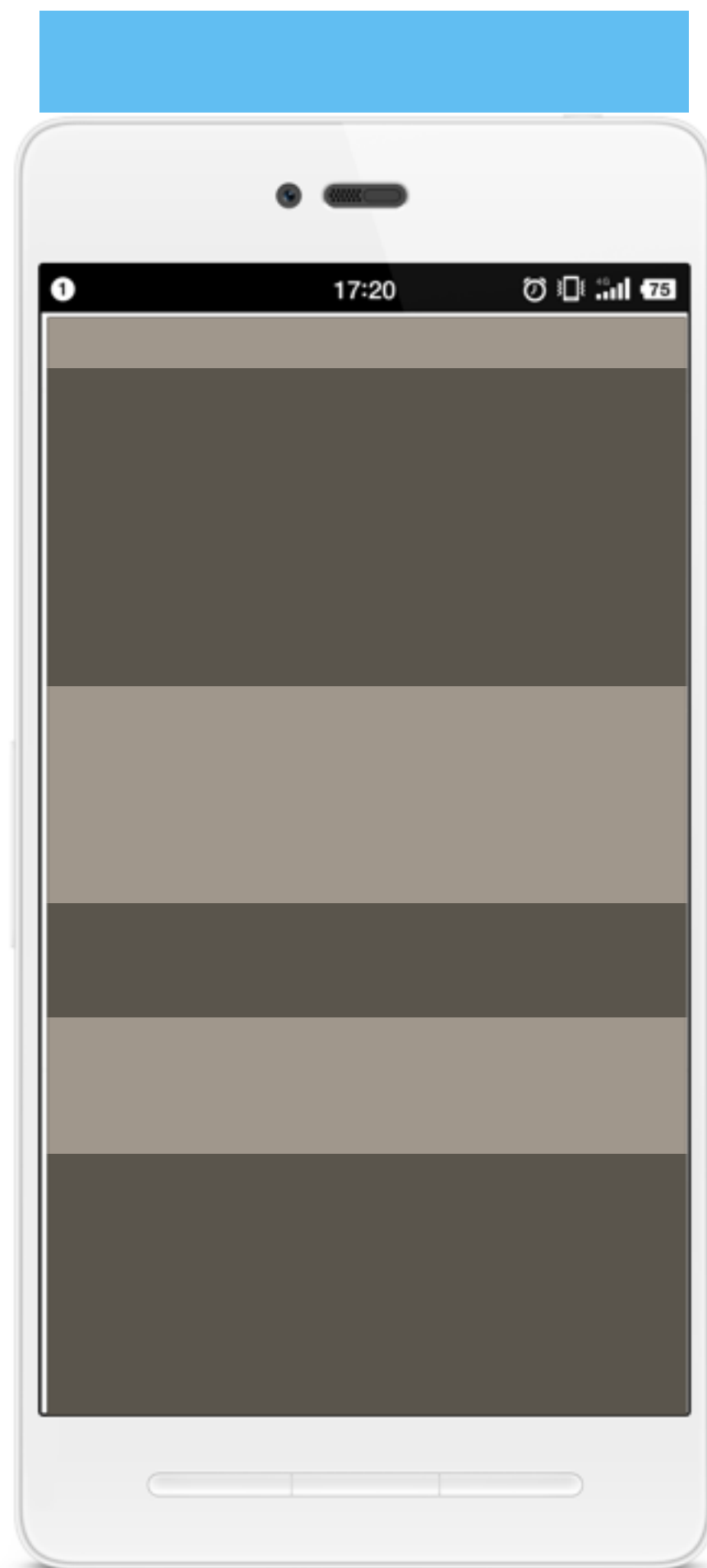
padding-top



CurrentList

WholeList

padding-top



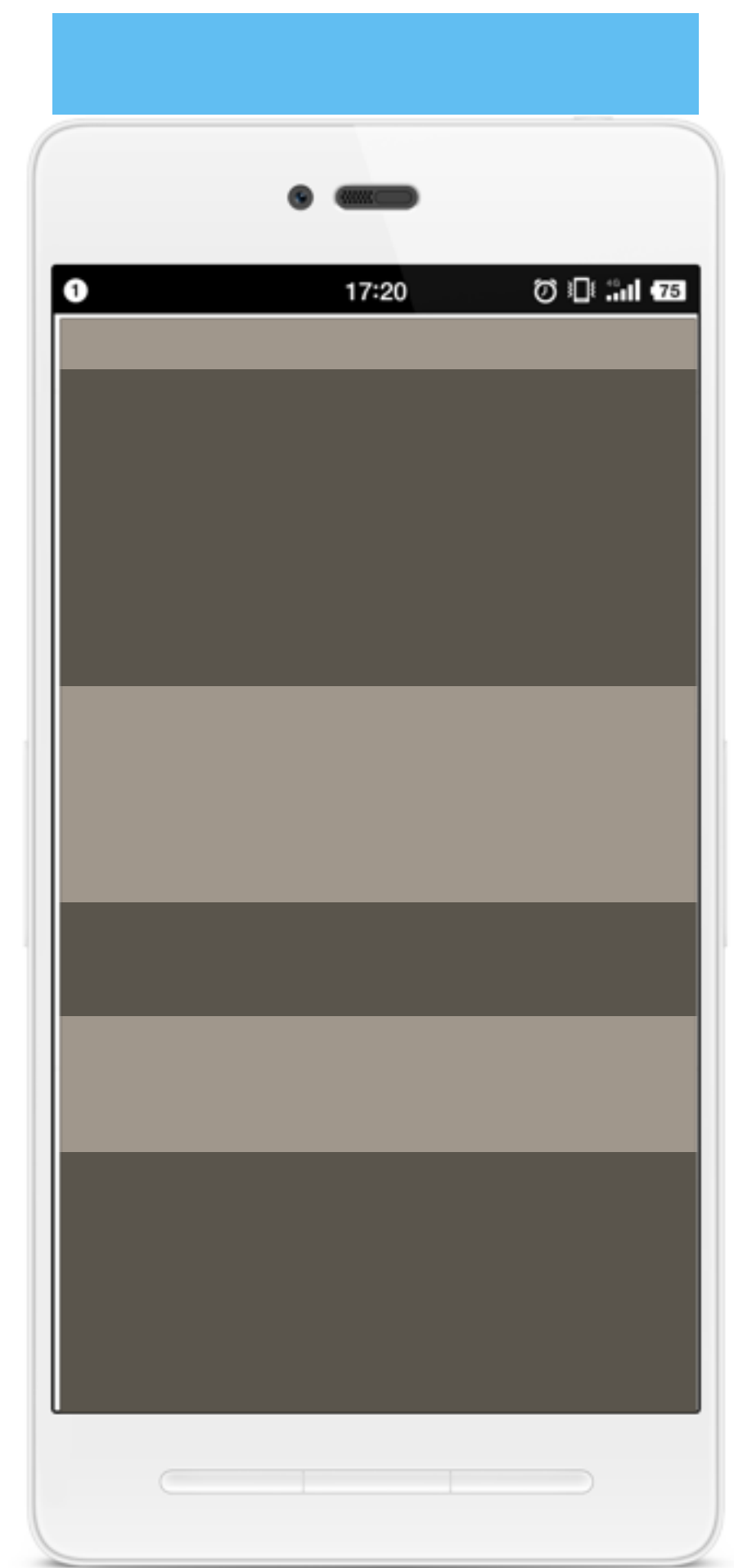
padding-bottom



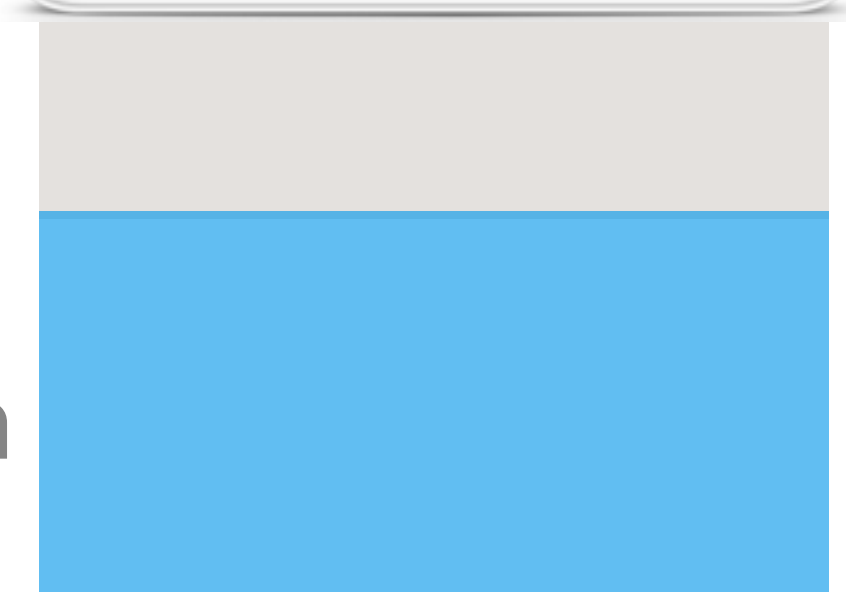
CurrentList

WholeList

padding-top



padding-bottom



旧版:

- 300条左右浏览器闪退



新版:

- 1500条以上无压力

解决方案：

- 手动点击翻页，破坏连贯性
- DOM过多：卡顿、白屏、崩溃
- 跳转页面后不能返回上一次阅读位置
- 无限滚动
- 微博卡片回收
- SPA记录滚动高度

其他解决思路：虚拟滚动行为

模拟弹性、模拟惯性.....

带来的另外一些问题

1. 快速滚动时的定位:

- * 滚动过程中JS阻塞带来的白屏
- * 滚动过程中JS不阻塞时不必要的渲染

2. 图片资源的缓存



Part 2

拖放排序解决方案

传统思路的拖放排序

SortableJS



github.com/RubaXa/Sortable

SortableJS

S O R
T A B
L E

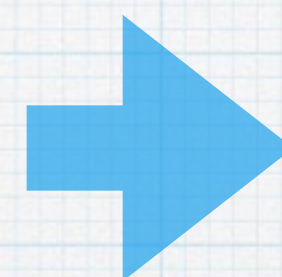
Group A



SortableJS

Vue.Draggable

SOR
TAB
LE



VUE
DRA
GGA
BLE

移动端没有 drag 事件，所以SortableJS需要一个占位元素和一个正在被拖拽的影子元素模拟跟随拖拽运动的元素。

The screenshot displays a web browser with a Vue.js application. The application has a title "Vue Draggable For" and two sections: "Draggable" (green background) and "Normal v-for" (grey background). The "Draggable" section contains a list of names: Joao, Jean, and John. The "Normal v-for" section is currently empty. The browser's developer tools are open, showing the JavaScript code and the DOM tree.

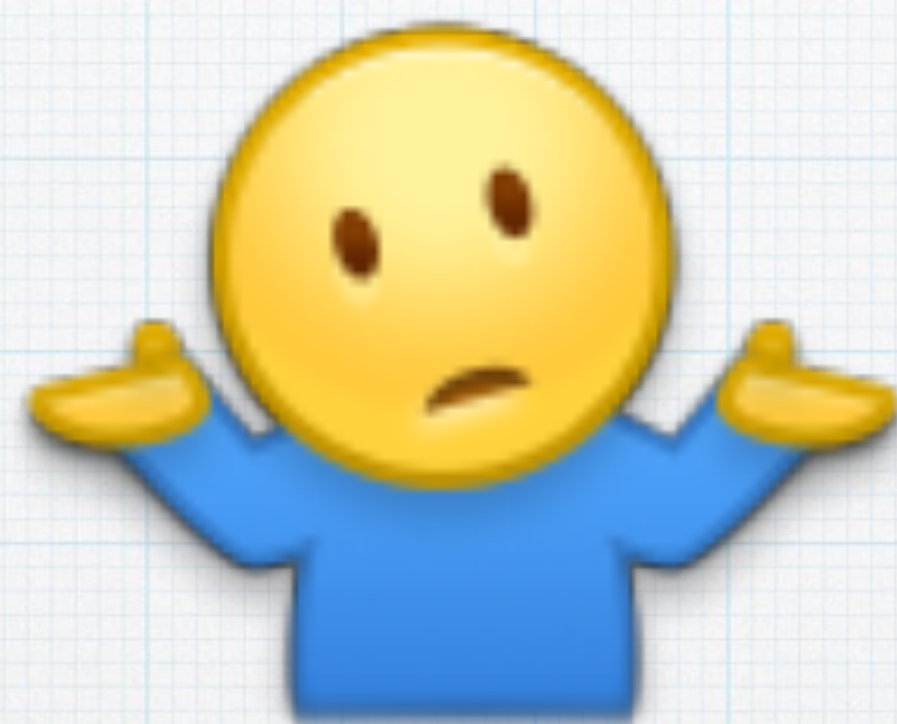
```
var vm = new Vue({
  el: "#main",
  data: {
    list: [{name: "John"},
           {name: "Joao"},
           {name: "Jean"} ]
  },
  methods: {
    add: function() {
      this.list.push({name: 'Juan'});
    },
    replace: function() {
      this.list = [{name: 'Edgard'}];
    }
  }
});
```

The DOM tree shows the following structure:

```
<div id="main">
  <h1>Vue Draggable For</h1>
  <div class="drag">
    <h2>Draggable</h2>
    <div class="dragArea">
      <div draggable="false" class="Joao">Joao</div>
      <div draggable="false" class="Jean">Jean</div>
      <div>John</div>
    </div>
  </div>
  <div class="normal">...</div>
  <button>Add</button>
  <button>Replace</button>
</div>
```

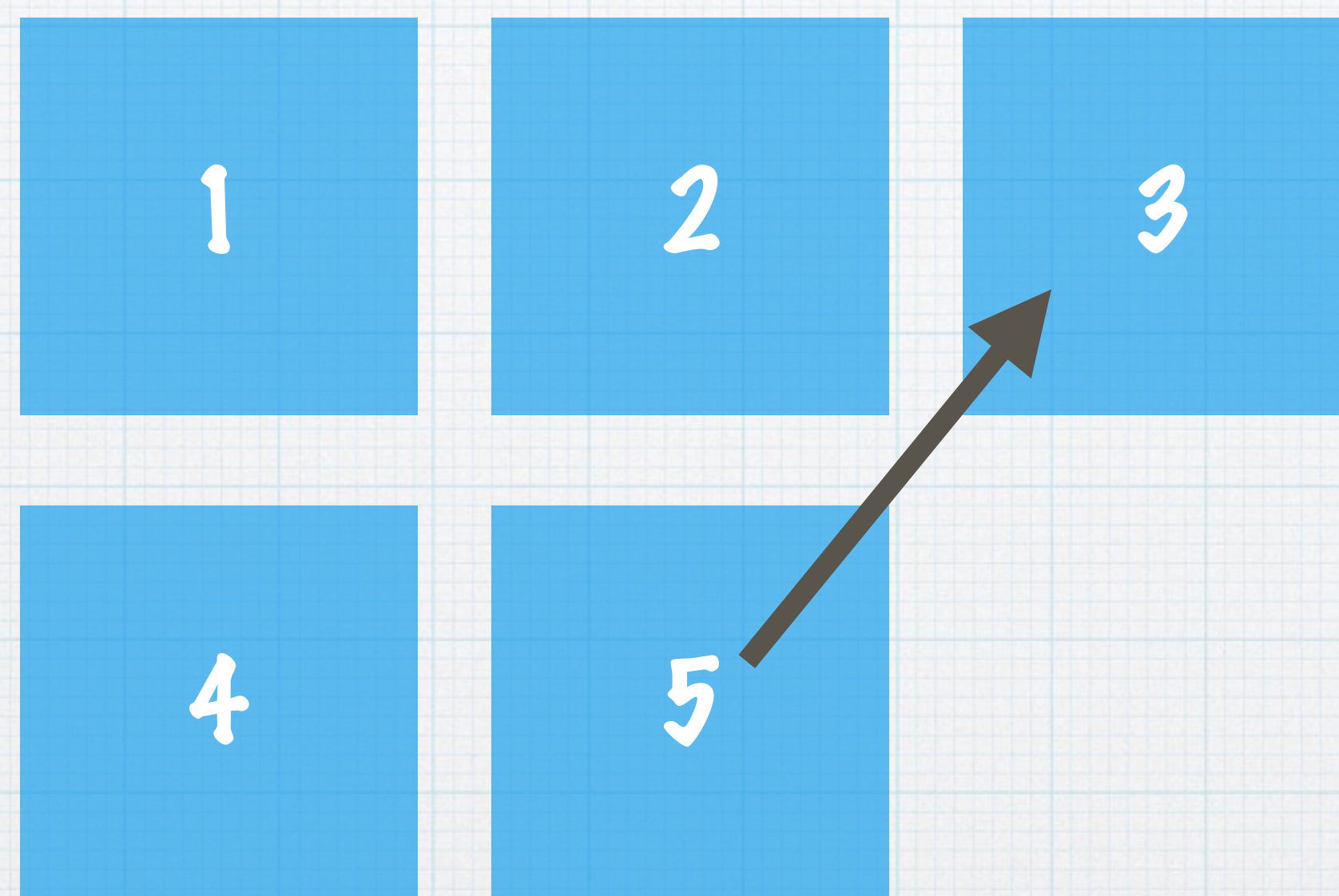
The Styles panel shows the following styles for the selected element:

```
div {
  display: inline-block;
}
```

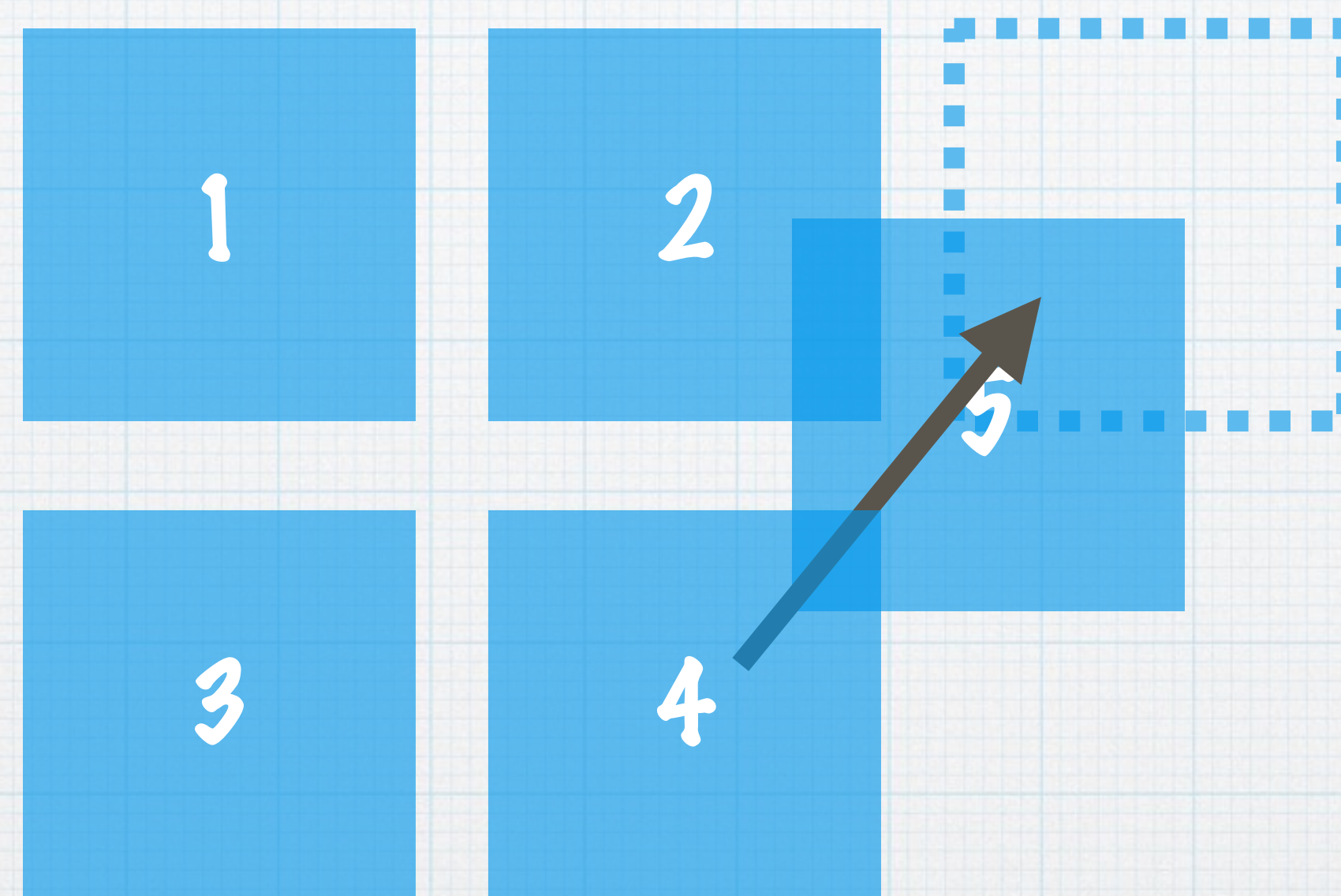



这就很不 MVVM 啊~
不是我想要的!

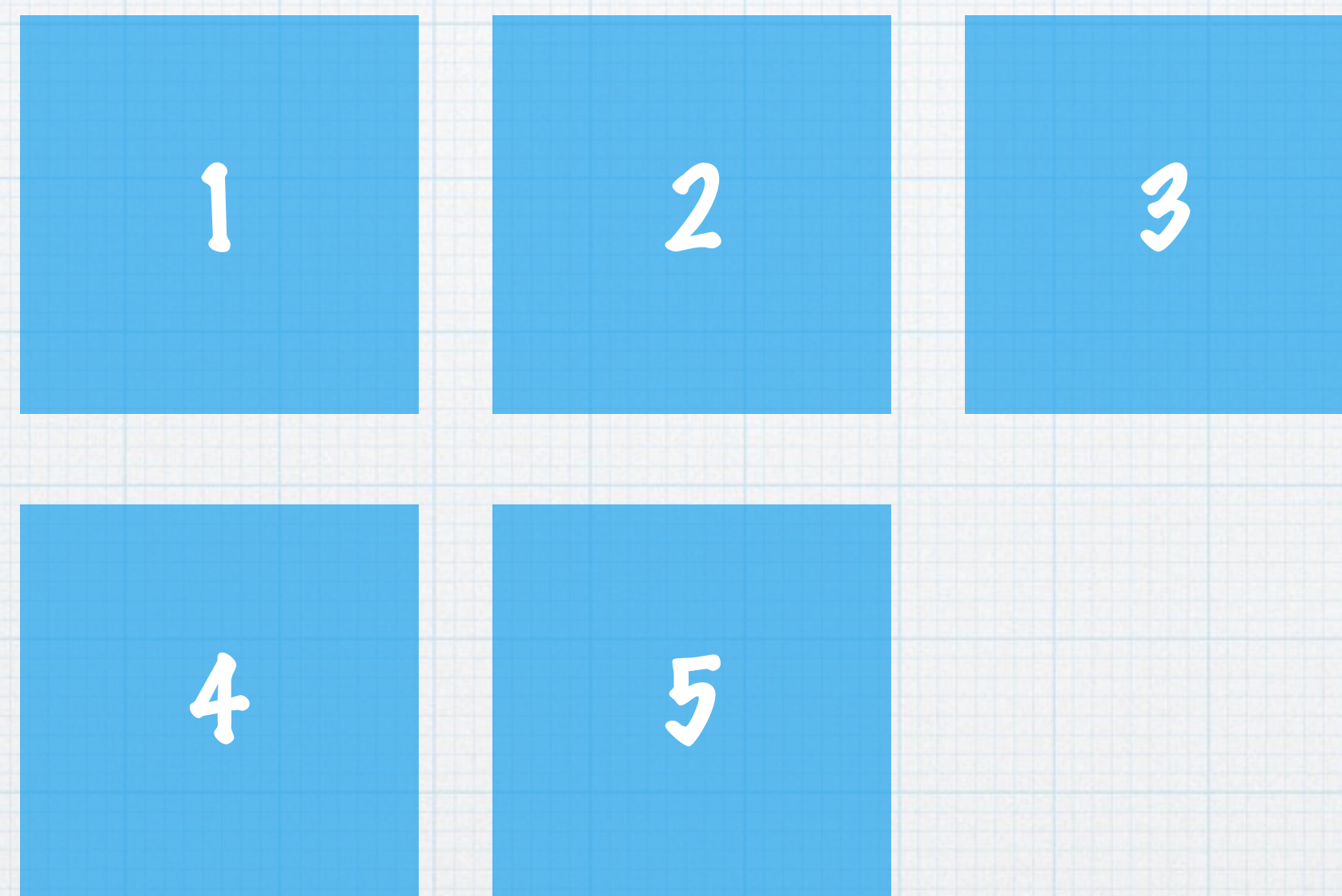
MVVM



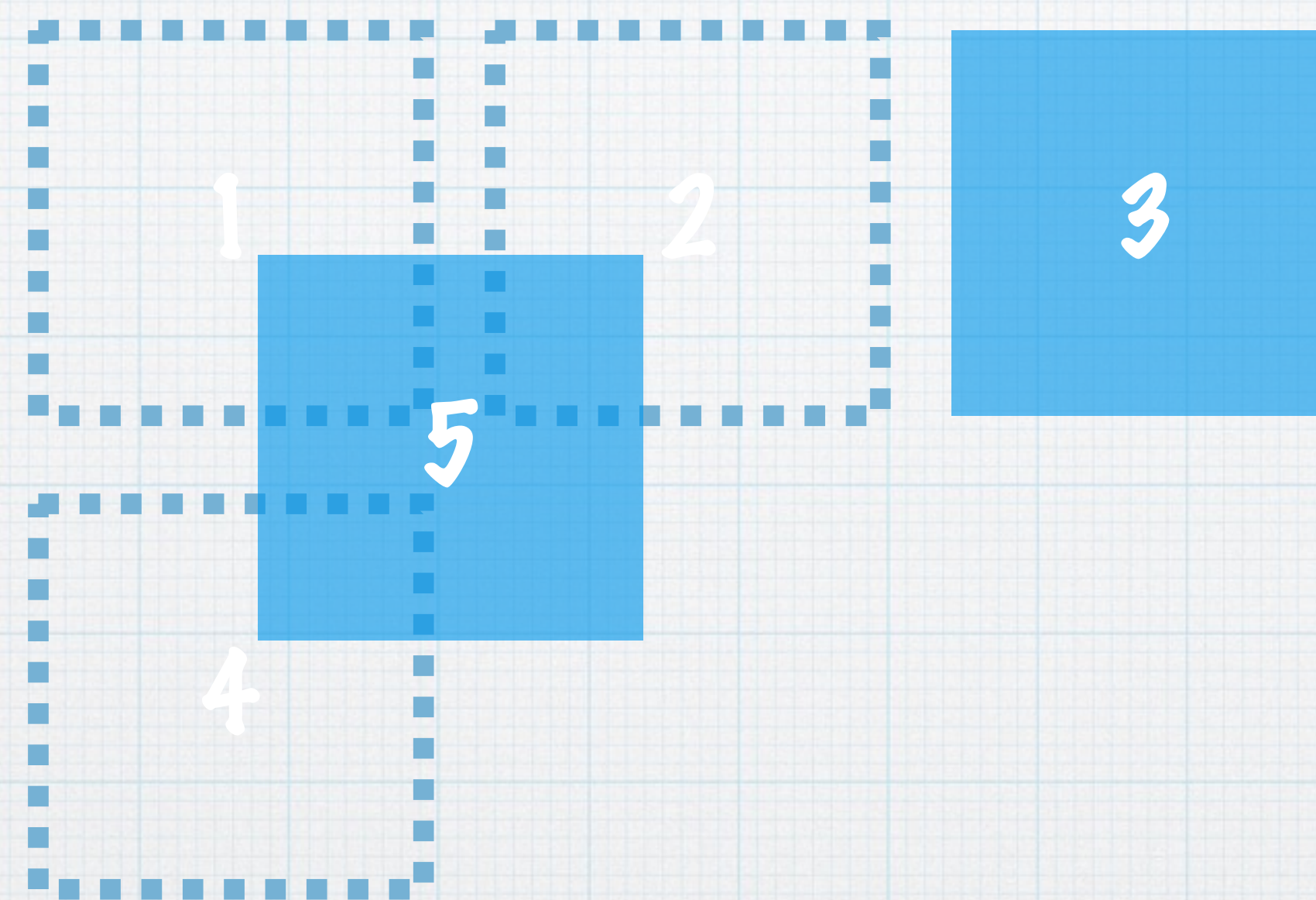
MVVM



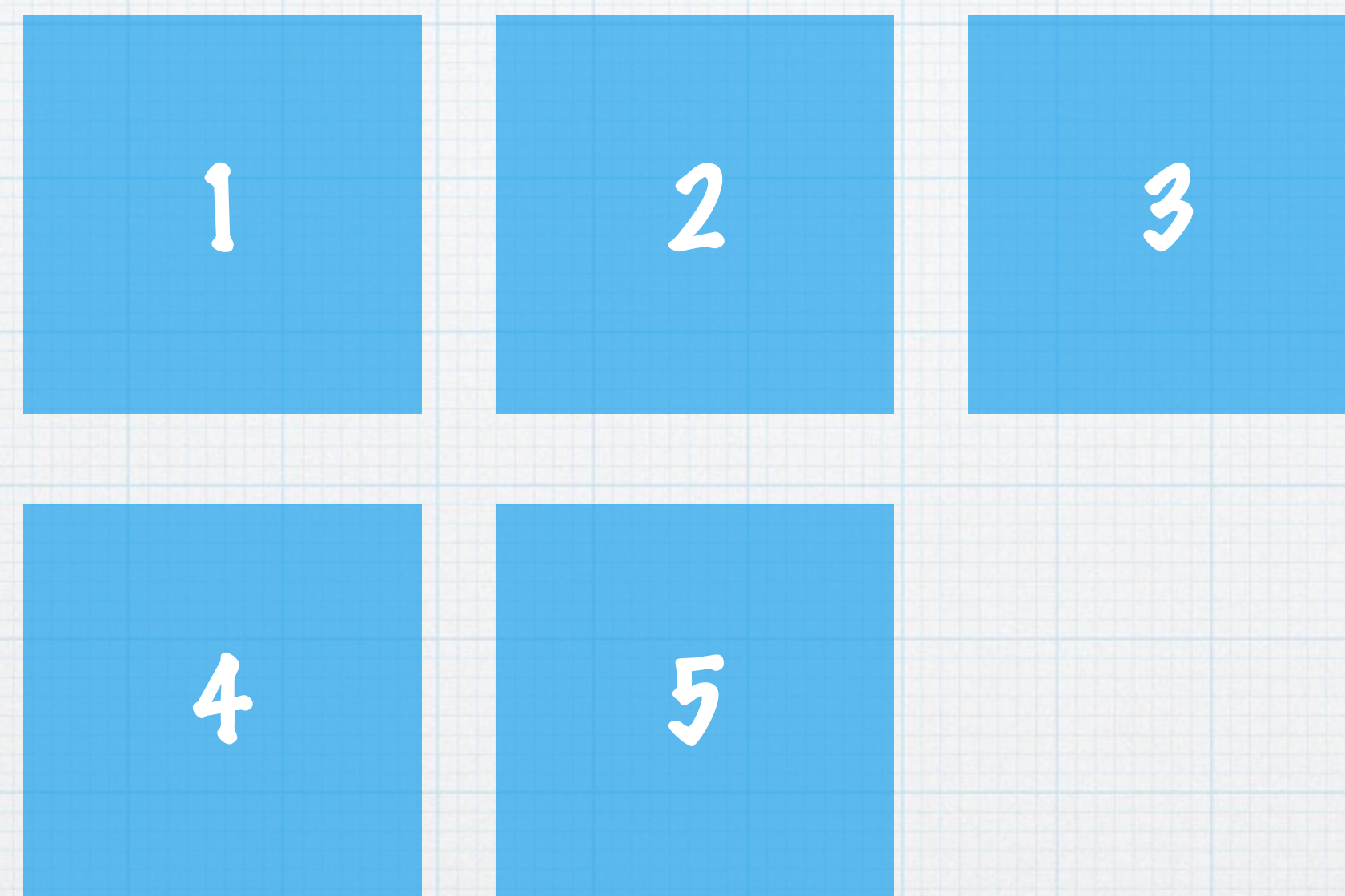
MVVM



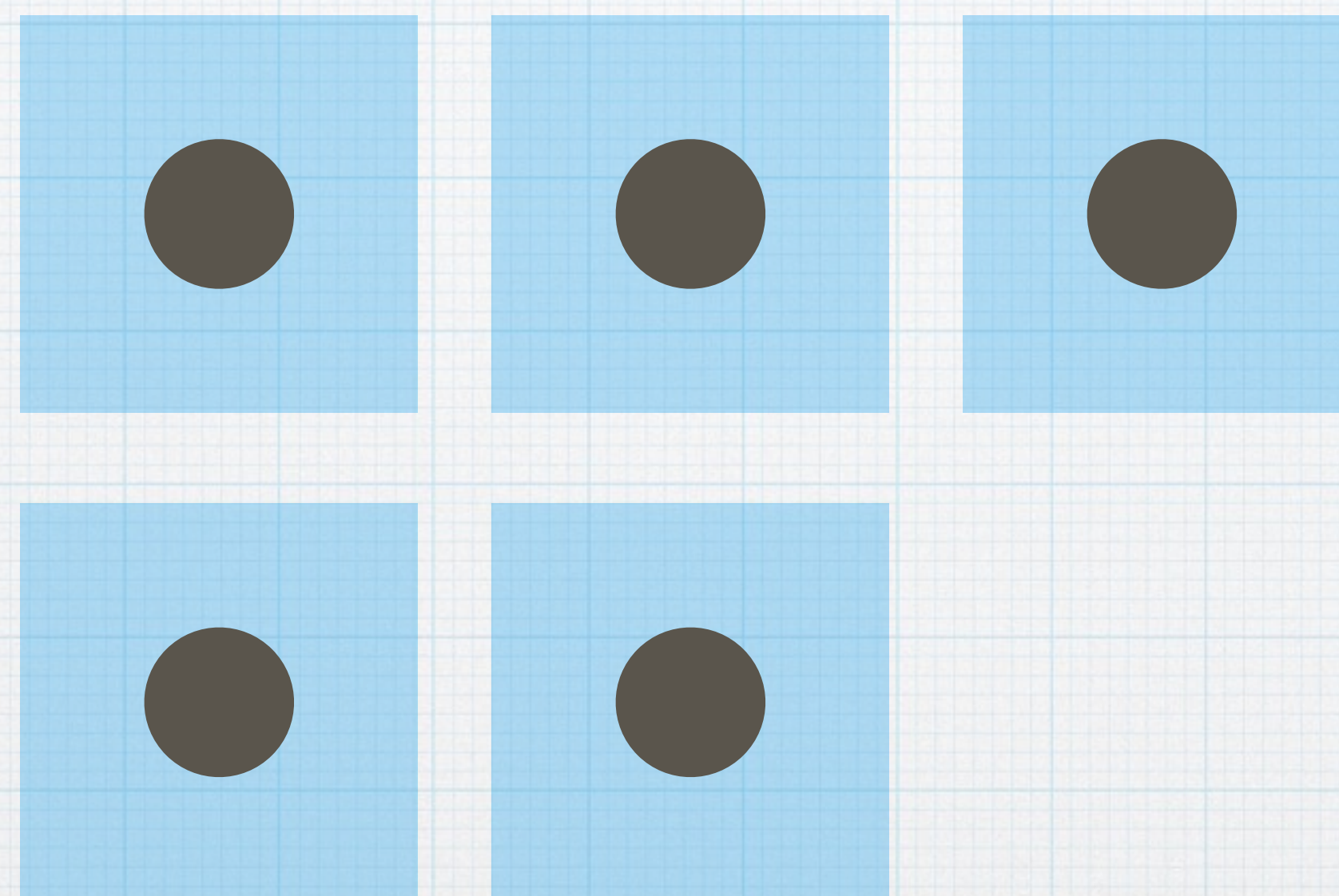
MVVM



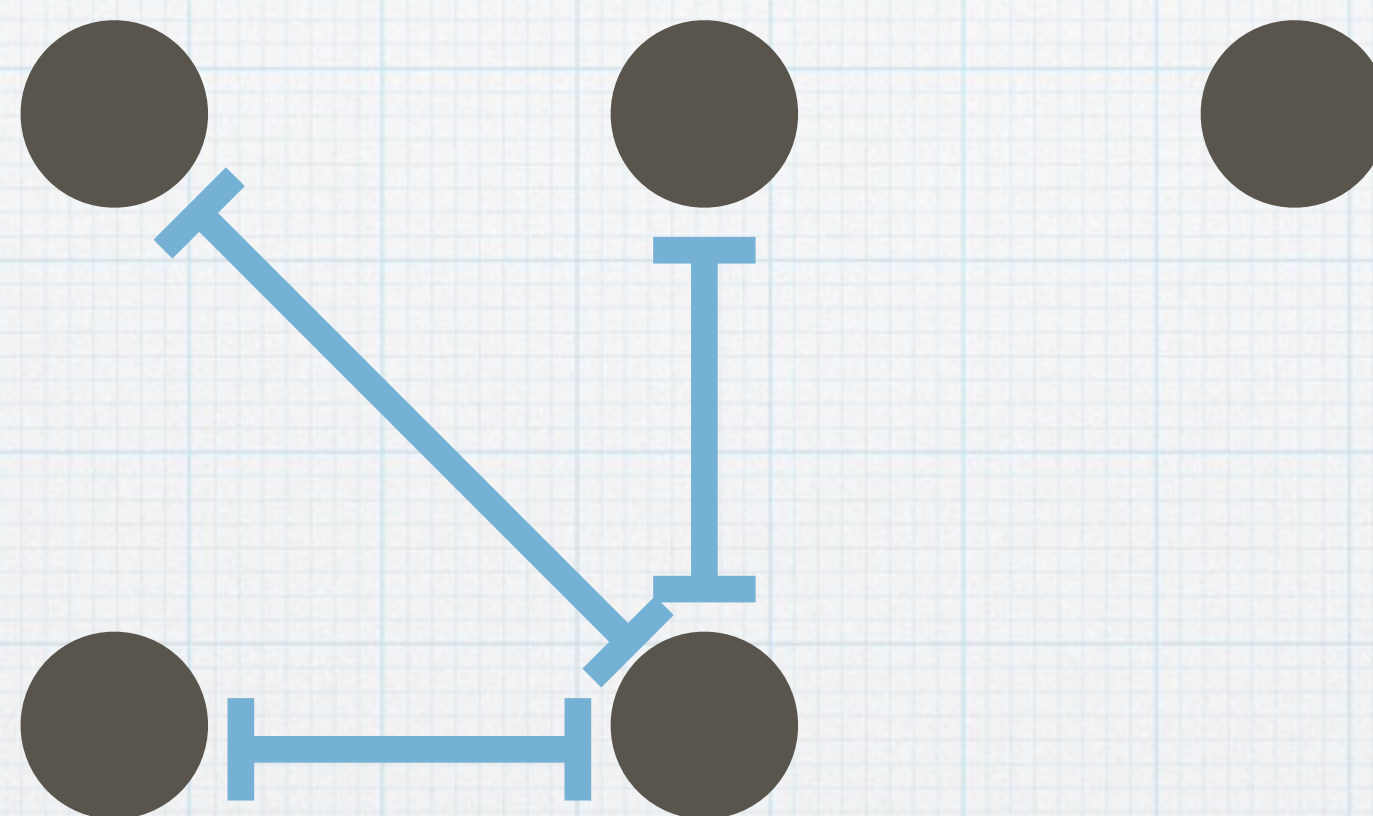
MVVM



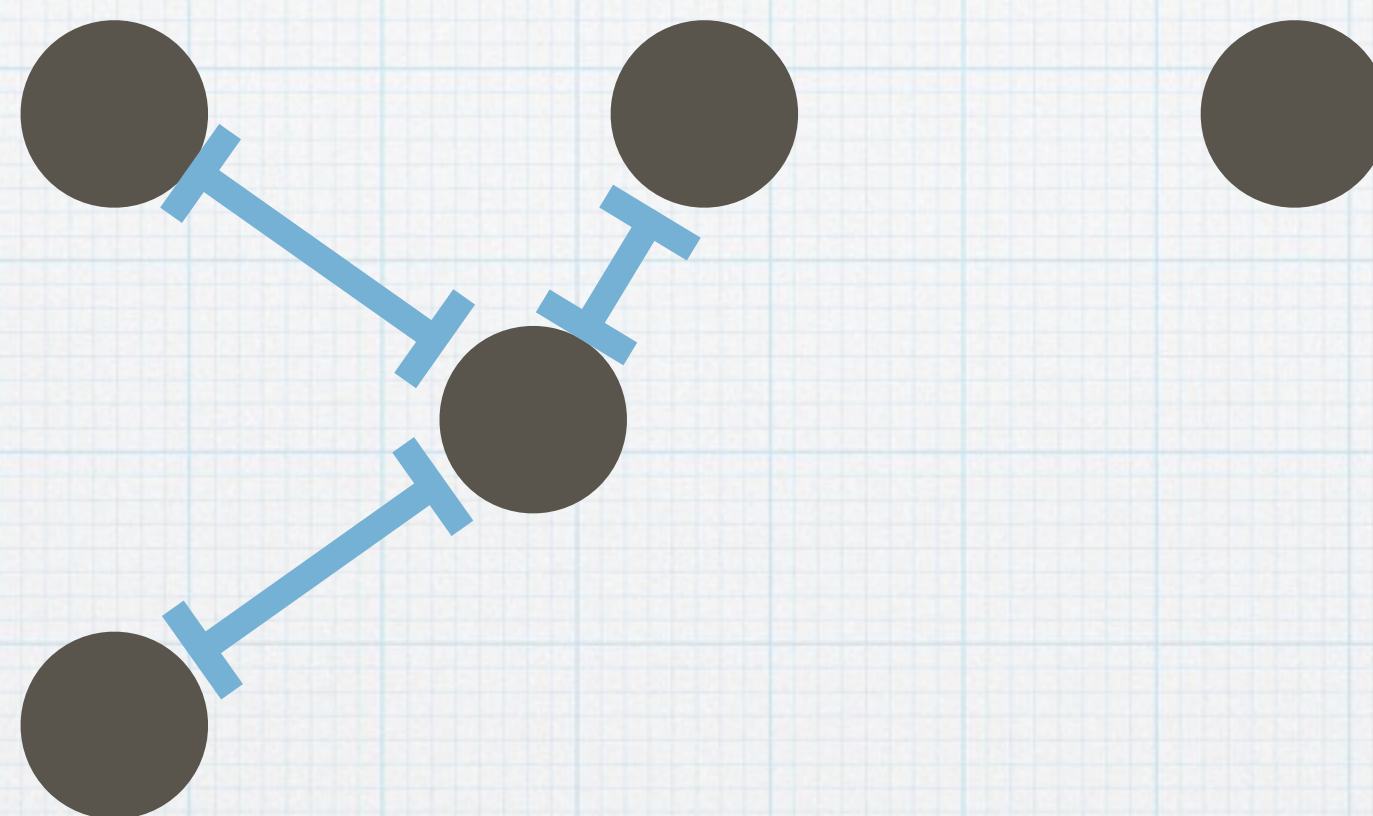
MVVM



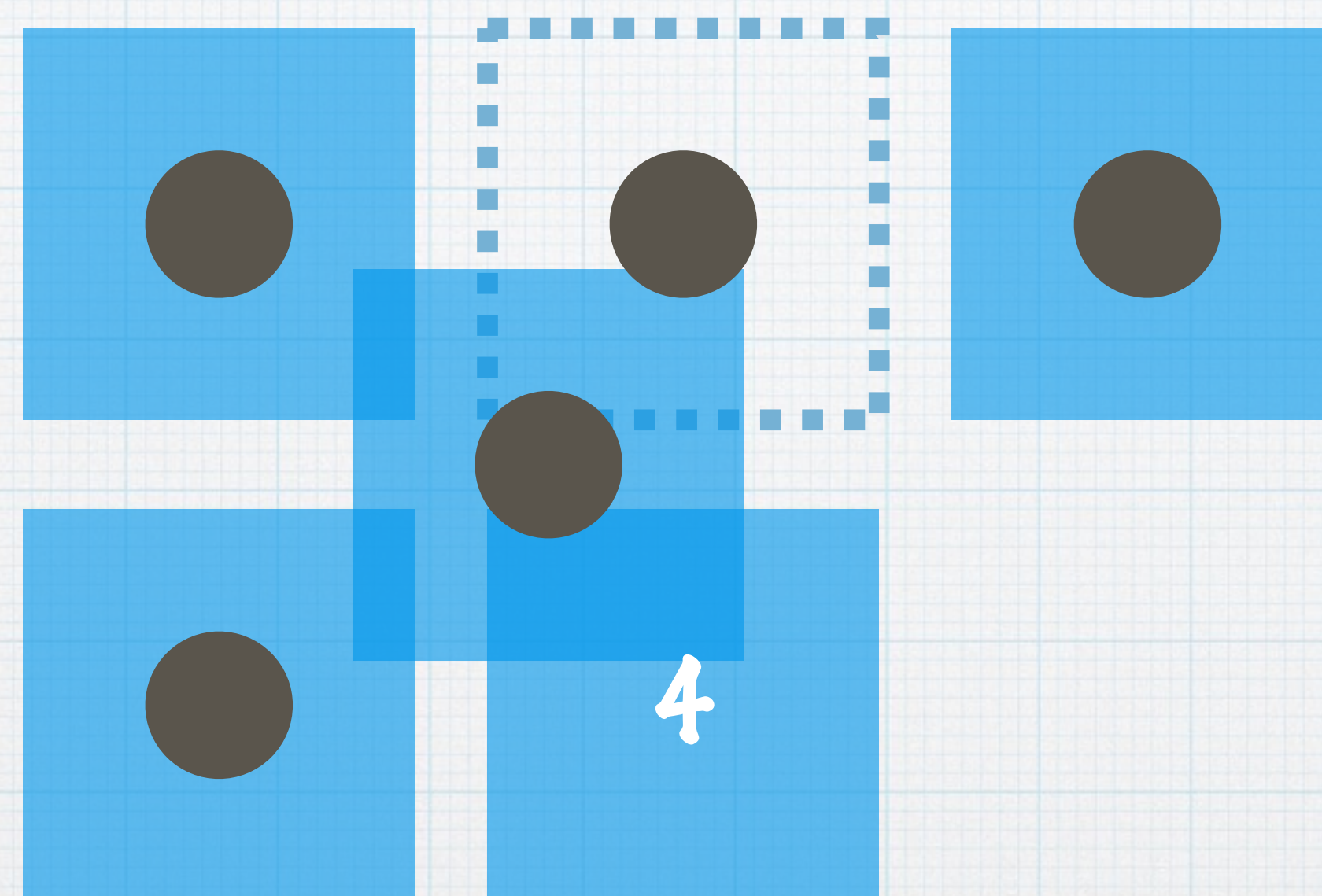
MVVM



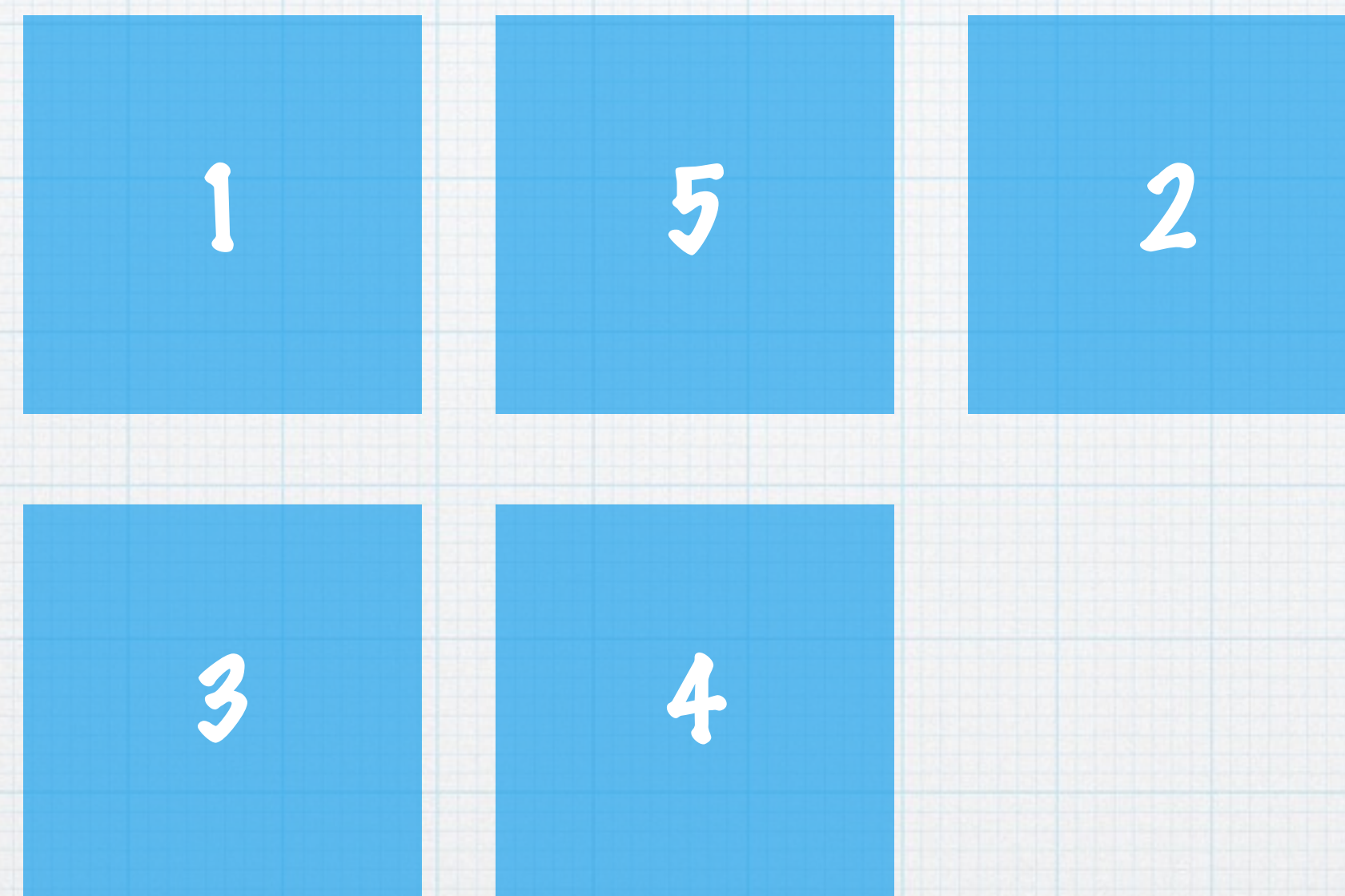
MVVM



MVVM



MVVM



基于MVVM的拖放排序的特点：

- 不需要另外设置影子元素和占位元素，所有元素都是真实元素的位置改变
- 需要知道每个元素的初始位置
- 除被拖动的元素之外的元素位置可预测

分享新鲜事...



公开

```
▶ <div class="m-box-model m-pos-r">...</div>
▼ <div data-v-6a23f01c>
  ▼ <div data-v-6a23f01c class="image-list">
    ▶ <div data-v-6a23f01c draggable="true" class="m-box-center m-box-center-a image-wrap m-trans img3 none" style="transform: translate(0px, 0px); touch-action: none; user-select: none; -webkit-user-drag: none; -webkit-tap-highlight-color: rgba(0, 0, 0, 0);">...</div>
    ▶ <div data-v-6a23f01c draggable="true" class="m-box-center m-box-center-a image-wrap m-trans img3 none" style="transform: translate(0px, 0px); touch-action: none; user-select: none; -webkit-user-drag: none; -webkit-tap-highlight-color: rgba(0, 0, 0, 0);">...</div>
    ▶ <div data-v-6a23f01c draggable="true" class="m-box-center m-box-center-a image-wrap m-trans img3 none" style="transform: translate(0px, 0px); touch-action: none; user-select: none; -webkit-user-drag: none; -webkit-tap-highlight-color: rgba(0, 0, 0, 0);">...</div>
    ▶ <div data-v-6a23f01c draggable="true" class="m-box-center m-box-center-a image-wrap m-trans img3 none" style="transform: translate(0px, 0px); touch-action: none; user-select: none; -webkit-user-drag: none; -webkit-tap-highlight-color: rgba(0, 0, 0, 0);">...</div>
    ▶ <div data-v-6a23f01c draggable="true" class="m-box-center m-box-center-a image-wrap m-trans img3 none" style="transform: translate(0px, 0px); touch-action: none; user-select: none; -webkit-user-drag: none; -webkit-tap-highlight-color: rgba(0, 0, 0, 0);">...</div>
    ...
    ▼ <div data-v-6a23f01c draggable="true" class="m-box-center m-box-center-a image-wrap m-trans img3 none" style="transform: translate(0px, 0px); touch-action: none; user-select: none; -webkit-user-drag: none; -webkit-tap-highlight-color: rgba(0, 0, 0, 0);"> == $0
      <div data-v-6a23f01c class="image-placeholder"></div>
      
      ▶ <button data-v-6a23f01c class="m-rpic-close m-trans m-box">...</button>
      </div>
    ▶ <div data-v-6a23f01c draggable="true" class="m-box-center m-box-center-a image-wrap m-trans img3 none" style="transform: translate(0px, 0px); touch-action: none; user-select: none; -webkit-user-drag: none; -webkit-tap-highlight-color: rgba(0, 0, 0, 0);">...</div>
```


Part 3

多级页面通用模板组件优化方案

推荐 榜单 社会 笑话 房产家居 星座 健 十



#汶川地震9周年祭# 热

2008年，那场8.0级的地震，夺去了近7万同胞的生命...

3.3万讨论 5126.4万阅读



#优酷员工伤人事件#

腾讯发表声明，称10日晚，腾讯员工应邀参加某影视公...

1403讨论 666.5万阅读



#吴敏霞结婚# 热

12日有网友偶遇吴敏霞录制某节目，男友张效诚向其求...

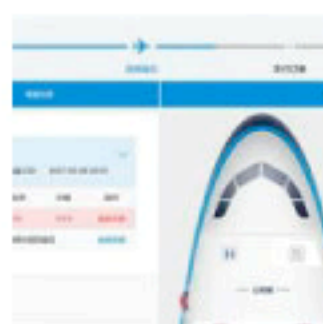
1.9万讨论 1.3亿阅读



#共享手机#

继共享单车、共享充电宝后，又出共享手机，风口的这...

362讨论 7万阅读



#80元买明星航班行程# 热

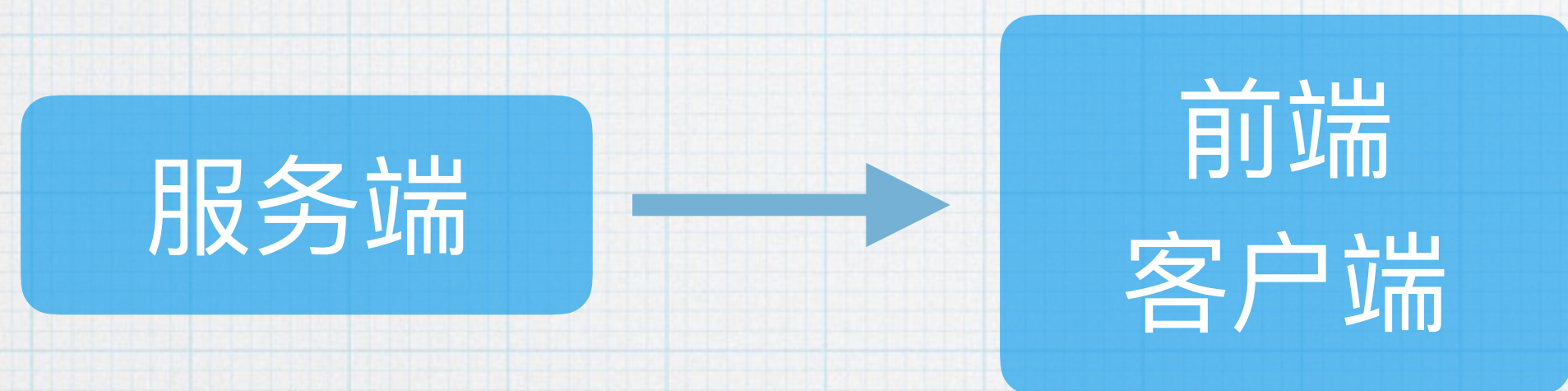
新京报记者调查发现，一些第三方航空软件，也在航空...

133讨论 6.3万阅读

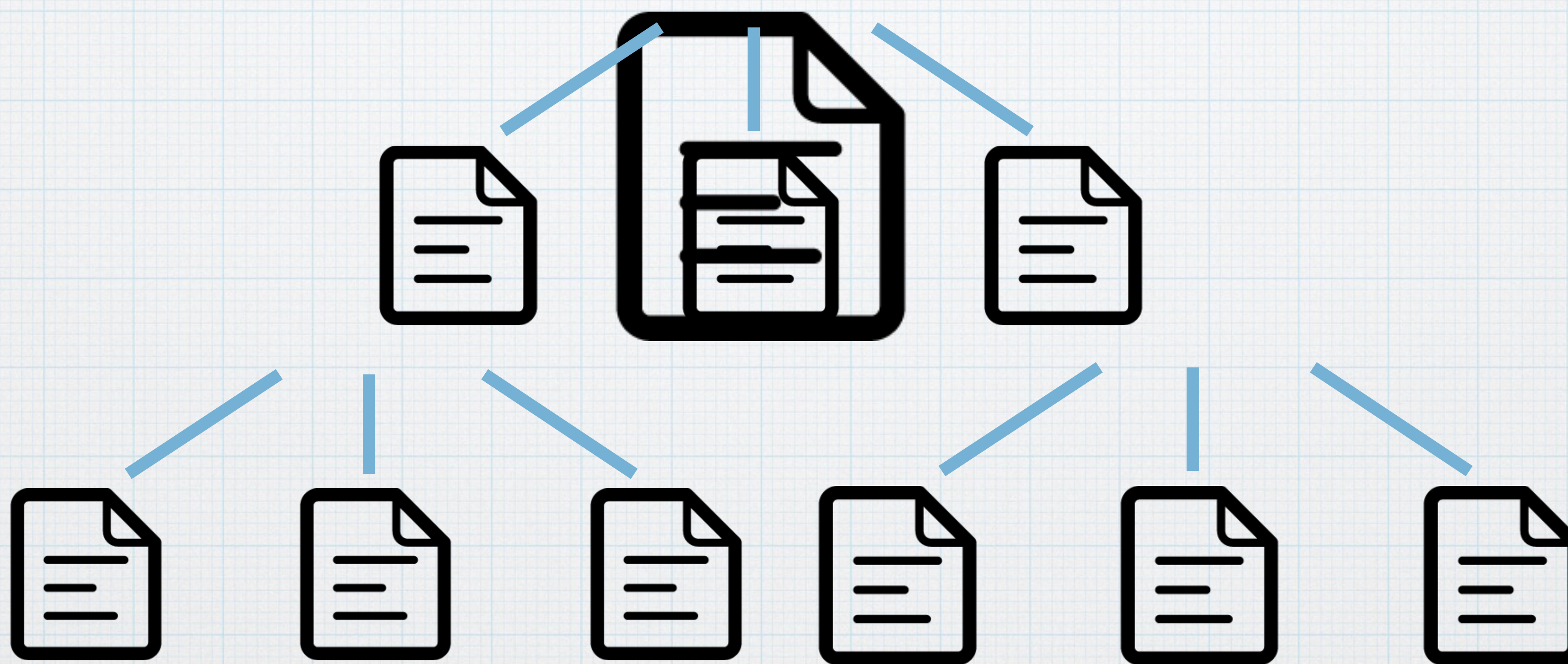
服务端

前端



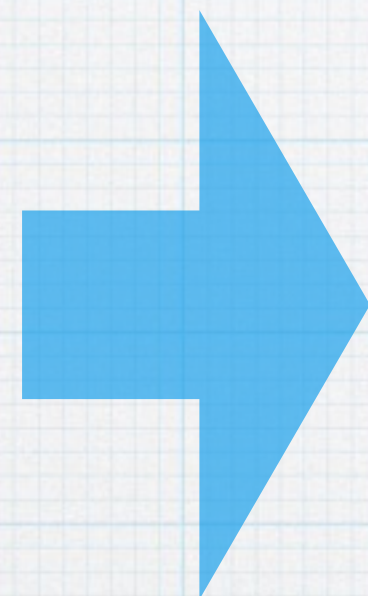


数据与客户端基本一致
相似场景: 文章页跳文章页

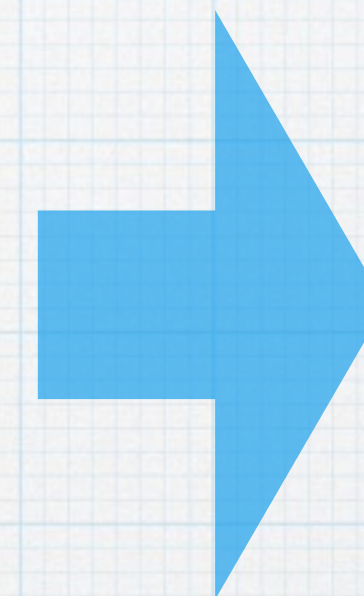


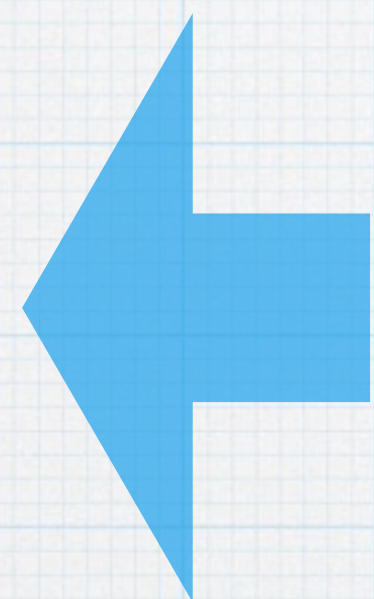
整站用是同一个页面模板组件，组件能根据接口数据自行渲染出想要的页面。那么，页面间的每一次跳转都要请求接口并重置所有数据吗？

都要请求接口并重置所有数据吗？

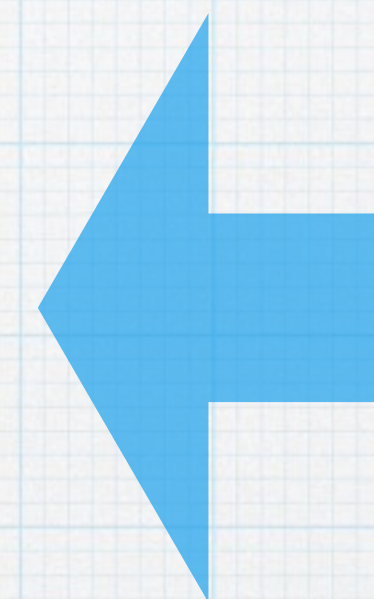


```
watch: {  
  pageId() {  
    this.init()  
    this.getData()  
  }  
}
```





```
watch: {  
  pageId() {  
    this.init()  
    this.getData()  
  }  
}
```

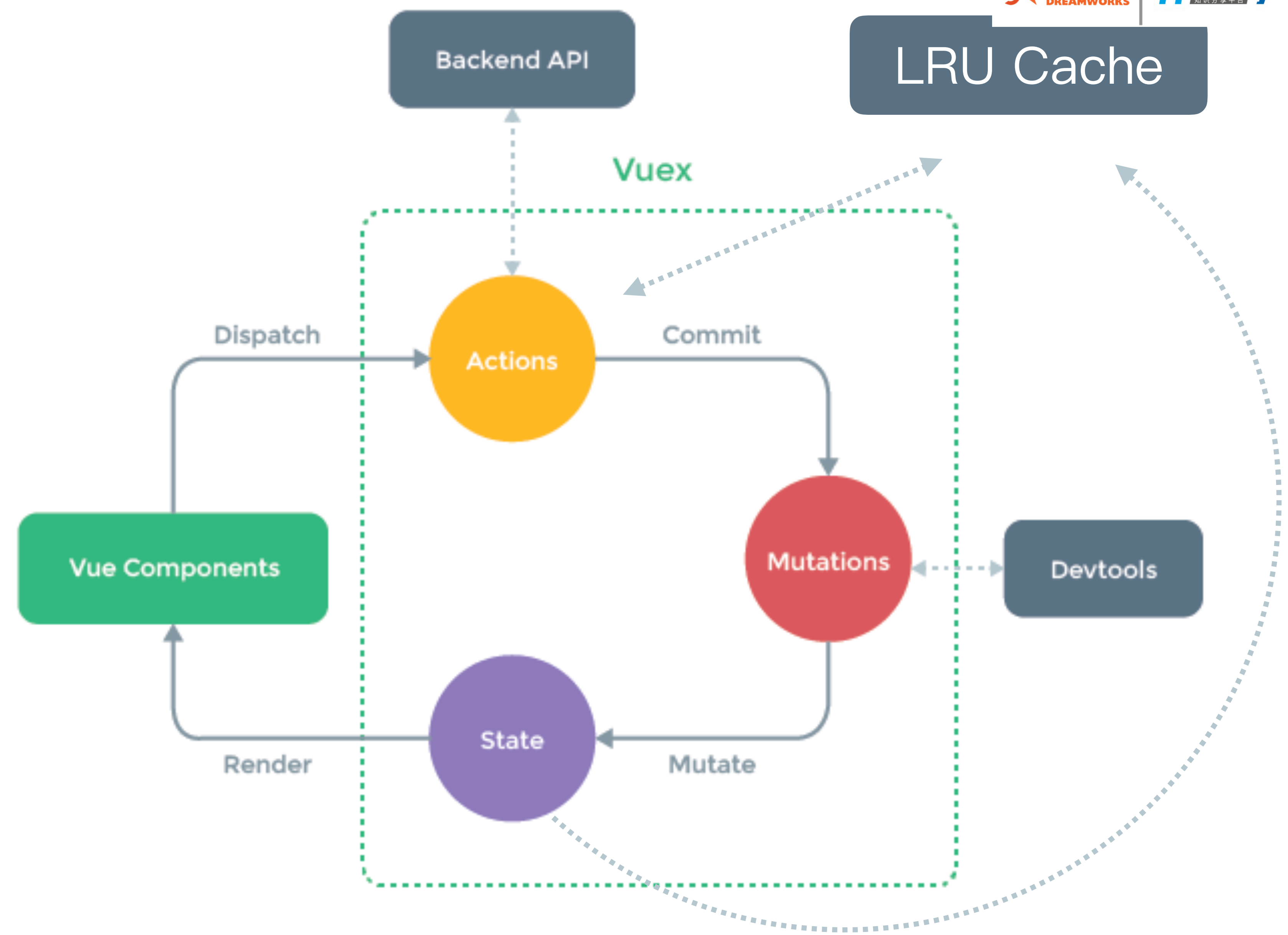




不如撸一个“前端路由”

- 无限滚动
- 拖放排序
- 通用模板组件

LRU Cache + Vuex



推荐 榜单 社会 笑话 房产家居 星座 健十

#汶川地震9周年祭# 热
 2008年，那场8.0级的地震，夺去了近7万同胞的生命...
 3.3万讨论 5126.4万阅读

#优酷员工伤人事件#
 腾讯发表声明，称10日晚，腾讯员工应邀参加某影视公...
 1403讨论 666.5万阅读

#吴敏霞结婚# 热
 12日有网友偶遇吴敏霞录制某节目，男友张效诚向其求...
 1.9万讨论 1.3亿阅读

#共享手机#
 继共享单车、共享充电宝后，又出共享手机，风口的这...
 362讨论 7万阅读

#80元买明星航班行程# 热
 新京报记者调查发现，一些第三方航空软件，也在航空...
 133讨论 6.3万阅读

VS

0:57

m.weibo.cn/p/100803

推荐 榜单 社会 笑话 政务 科十

#315晚会# 热
 中央电视台第27届3·15晚会将于2017年3月15...
 31.2万讨论 7.4亿阅读

#设计秀#
 从今日起，带话题#设计秀#发微博+作品图或短...
 120.7万讨论 12.4亿阅读

#微博315# 热
 微博315，共建消费新生态！
 37.8万讨论 6.7亿阅读

#皮皮虾,我们走#
 皮皮虾，我们走
 1.6万讨论 1323.7万阅读

#安以轩与富商男友结婚# 热
 安以轩在微博公开喜讯，与密恋两年多的男友...
 #安以轩#超级话题

#北京事儿# 本地
 咱是北京人，一起来说说身边的#北京事儿#。...

133讨论 6.3万阅读
 新京报记者调查发现，一些第三方航空软件，也在航空...

咱是北京人，一起来说说身边的#北京事儿#。...
 #北京事儿# 本地

- 无限滚动
- 拖放排序
- 通用模板组件

微博 HTML5 版



m.weibo.cn/beta