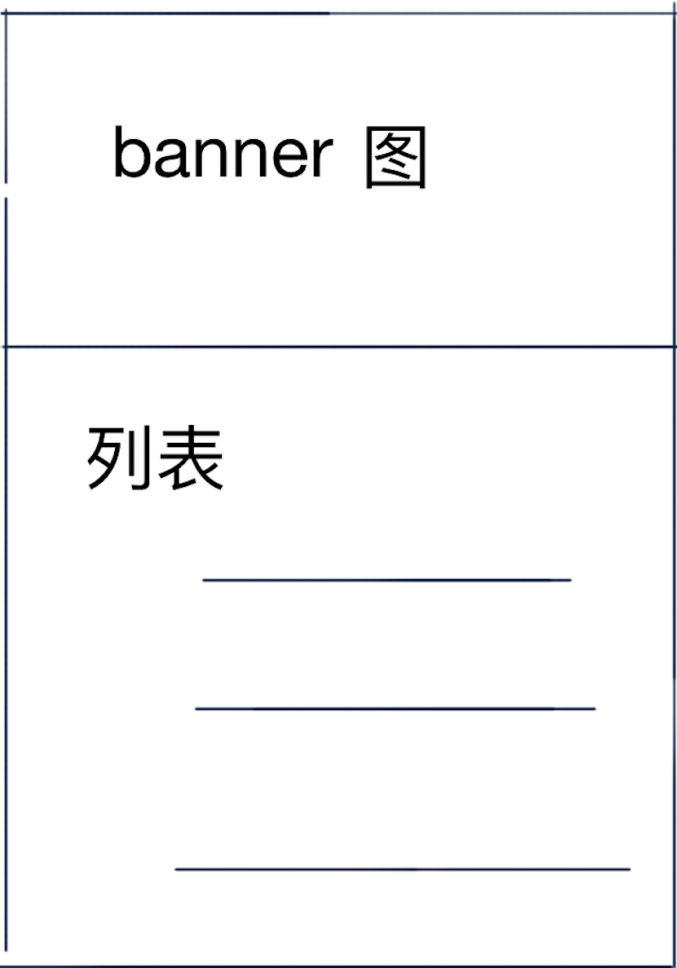


豆豆尖的入门之旅：可爱的小程序《二》

如果我们想写一个下面这种列表页：



我们先看 index.wxml 部分：

```
<view class="index-wrap">
  <view class="top-banner">
    <image src="https://s1.xxx.com/app/images/index/indexBanner.jpg?v=1" class="b
    <view class="banner-title">banner图</view>
  </view>
  <view class="items border-top-hack">
    <block wx:for="{{list}}" wx:for-index="i">
      <view bindtap="toItem" class="plan-list-content border-bottom-hack" data-id
      <view class="plan-content-top">
        <view class="plan-content-font">
          <view class="title">{{item.title}}</view>
          <view class="content-font-desc">{{item.advertisement}}</view>
          <view class="p">{{item.guaranteeScope}}</view>
          <view class="p">{{item.guaranteePeople}}</view>
          <view class="button">加入我们</view>
        </view>
      </view>
    </block>
  </view>
</view>
```

整页都是 view 标签看起来好蓝瘦。。。

下面的列表部分，我们用到了 wx:for 这个指令来循环这个列表，这里的用法跟 vue 真是神似，下面依旧是模板数据填充，bindtap 用于绑定一个方法，用法可以在后面的 index.js 里看到。

接下来 wxss 就直接贴个图好了：

```
.items{  
  margin-top: 32rpx;  
  margin-bottom: 100rpx;  
  background: #fff;  
}
```

```
.plan-list-content{  
  display: block;  
  margin-top: 20rpx;  
  margin-left: 30rpx;  
  padding-top: 30rpx;  
  height: 200rpx;  
  background: #fff;  
}
```

```
.plan-content-top{  
  position: relative;  
  height: 170rpx;  
  background: #fff;  
  display: -webkit-box;  
  display: box;  
}
```

index.wxss

这里提一句，小程序的单位统一为 rpx,同样是根据屏幕宽度进行自适应的，但存在一个像素的问题，在不同手机里边框的 1px 是不同的视觉感受，不过搜到一个比较靠谱的解决办法，就是在公共 wxss 里进行如下设置：

(例) .border-top-hack {position: relative}

```
.border-top-hack:after {position: absolute;content: ";width: 100%;left: 0;height: 1px;background-color: #dadada;-webkit-transform: scale(1,.5);transform: scale(1,.5);-webkit-transform-origin: center top;transform-origin: center top;}
```

接下来,看看 index.js 部分：

我们通过 require 把需要用的模块引入进来。

若需要对一个值进行重新设置，一定要用上 this.setData()

```
var api = require('.././config/api.js'),  
    router = require('.././config/urlRouter.js');
```

```
var app = getApp();
```

```
Page({  
  data: {  
    list: []  
  },  
  
  onLoad() {  
    this.getPlanList();  
    app.getLoginInfo();  
  },  
  
  getPlanList() {  
    var that = this;  
  
    wx.request({  
      url: api.planList.url,  
      method: 'POST',  
      success: function(res) {
```

index.js

wx.request() 发送一个获取列表的请求，将返回值 push 出去，然后如前所说 wx:for 将列表渲染出来

```
toItem(e) {  
  var id = e.currentTarget.dataset.id;  
  
  wx.navigateTo({  
    url: router.planItem.url + '?id=' + id  
  });  
}
```

index.js-跳转事件

之前我们在 wxml 里用 bindtap 绑定了一个 toItem() 的方法，点击时就会触发这个事件，使用小程序封装得路由，wx.navigateTo 跳转到 下一页，并带上本条列表的 id 值。

这是一个简单的列表页，用到的 api 也就那么几个，但小程序提供了丰富的 api，有兴趣可以直接看小程序官方文档，感觉已经写得非常全面了~

小程序一直在更新组件，前天刚写了 `wx.navigateTo` 这玩意儿只能跳转五层→_→，今天发现又更新了！！！新出炉的 API，
`wx.navigateBack(OBJECT)`，可关闭当前页面，返回上一页面或多级页面。通过 [getCurrentPages\(\)](#) 可获取当前的页面栈，决定需要返回几层。
这就解决了上次说的只能返回 5 层的问题，里面的参数 `delta` 即返回的页面数，如果 `delta` 大于现有页面数，则返回到首页。
哈哈，第二篇小例子就说到这儿啦，感觉还是没办法把一个东西的精髓讲出来，看来自己还需要在这条路上慢慢修炼呐??~