案例：微投票

这个章节将会讲解如何定制一个微投票的完整过程。

5.2.1 案例功能详解

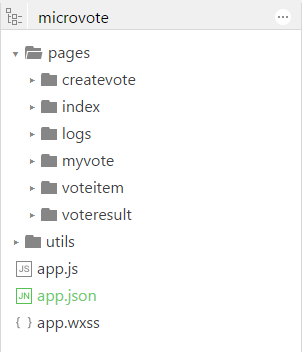
麻雀虽小，五脏俱全，本案例有5大功能：

1. 创建新的微投票主题。
2. 我的微投票主题
3. 投票功能（分享、生成二维码）
4. 扫码参与投票
5. 查询微投票结果

界面如下：



5.2.2 案例程序结构



其中index是首页， createvote是创建微投票主题和选项的页面，myvote是我创建过的微投票主题，voteitem是投票页面（含分享、二维码功能），voteresult 是微投票结果页。

5.2.3 案例程序细化详解

首先修改app.json ，设置标题：



然后定义index界面：

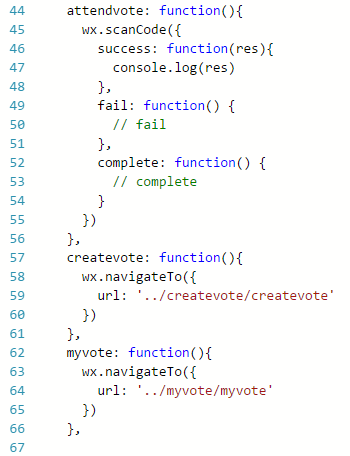


这里我们定义了三个button 组件作为微投票的功能菜单，一个是扫描二维码图片，然后跳转到对应的微投票主题上。第二个用来查看我创建过的微投票主题。第三个button用来创建一个新的微投票主题。分别绑定了三个方法，attendvote、myvote和createvote，这些方法将在index.js里实现。

先看看 index.js 里定义了一些变量：



而三个button绑定的方法代码如下：



扫码button调用wx.scanCode方法然后跳转到path上，其他两个button分别调用对应的页面，简单进入即可。

那让我们来实现这两个功能吧，先看“创建我的投票”的界面：



微投票的功能上我们可以设置了3个属性，一个是投票类型，说明微投票是单选，还是可以多选。一个是标题，最后一个是选项，我们会把它设计成可以变化的多个选项，只需要简单的点击 ”添加选项” 按钮就可以。

我们看看页面代码（createvote.wxml），类型和标题：



选项部分的页面代码：



业务方面我们的微投票至少要有2个选项，最多可以有16个选项，所以我们直接写死了前2个name分别为 voteopt1和voteopt2 的input选项 ，而从第三个开始到第十六个就不一样了，我们用block 把整个选项的view包裹起来，并通过op3condition这个变量开关来控制block 是否显示，实现选项的动态显示和隐藏， 达到自定义选项数量的目的。

添加选项的button我们绑定了一个叫addOption 方法，将会在createvote.js里实现：



而整个页面，我们用一个form的组件包裹，然后通过submit来提交到createvote.js 的formSubmit方法上：







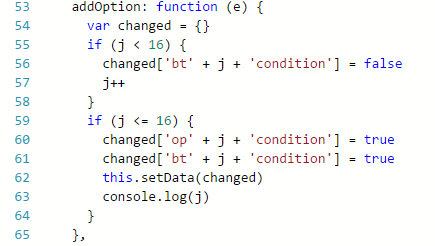
我们看看点击“添加选项”后会发生什么：



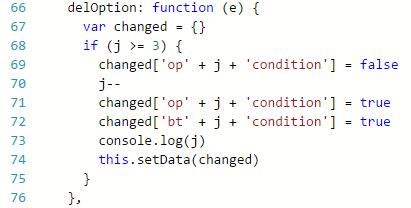
点击后发现多了个选项3，而且后边还多了个“-” 号，这个减号是我们拿来删除对应某个选项的操作。再点一次“添加选项”：



我们又发现，选项4已经成功添加，同时选项3的 “-” 减号已经消失，这时如何实现的呢？ 看看代码：



这里就是通过控制对应选项的block上定义的变量op4condition，变更成true来显示，同时把上个选项的“-”减号的变量bt3condition 变成false来隐藏。同理，点击“-” 减号，也可以把本选项隐藏掉，同时把上个选项的“-”减号显示：



j 是页面变量，用来控制对应的选项。点击“-“ 减号后就变成：

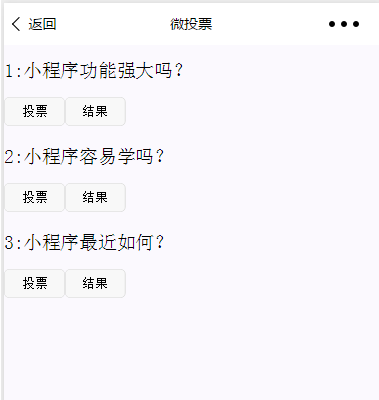


这时候，我们选择好类型，填好标题、选项内容后，点击“创建微投票主题“按钮，就可以点生成我们的微投票主题啦：



这里我们的form会调用formSubmit方法，并通过POST提交数据到远程服务器上，如果成功返回，就会弹出弹窗提示相关信息，之后跳转到”我的投票列表“页面上。关于服务器代码和数据库结构，在后面的部分，将会提到。

第二个功能“我的投票主题“：



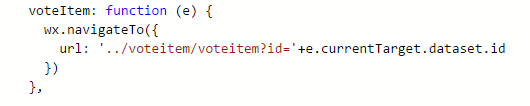
通过这个界面可以查看我曾经创建的微投票，并且可以通过这个界面去到微投票页面和微投票的结果页。这个页面在 myvote 目录下，页面myvote.wxml很简单：



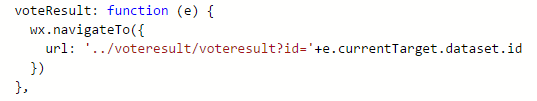
主要是通过onLoad 方法加载数据：



通过voteItem 去到真正的微投票页面：

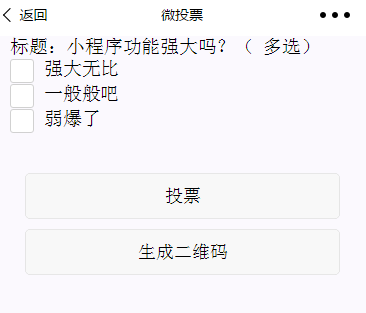


通过voteResult 去到微投票的结果页面：



第三个功能“投票“：

在我的投票主题里，点击”投票“，去到微投票真正的投票页（voteitem.wxml）：



页面代码如下：



数据通过onLoad 加载（voteitem.js）：



选择选项后，点击”投票“，通过form提交：



执行成功返回后，根据返回的状态码，成功的话跳到为投票的结果页，失败的话返回微投票主题列表：



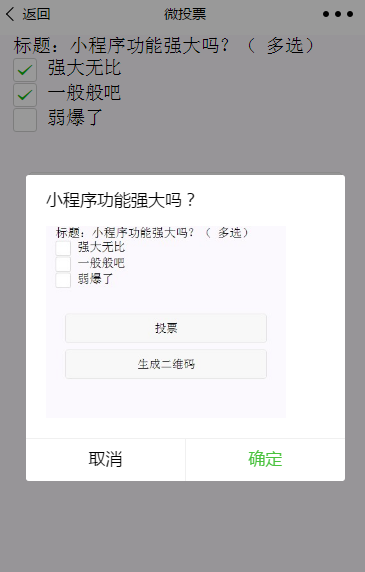
这个页面上，我们还看到一个生成二维码的按钮，点击后将会在页面的下部生成一个二维码的图片，具体实现代码如下，必须先要根据你应用的appid和secret 先获取access\_token，然后才能调用api去生成：



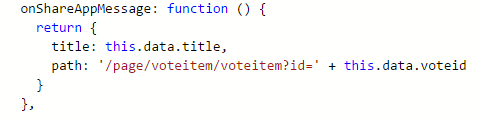
在这个页面，我们还实现了分享功能，点击右上角的”...“ 后会弹出：



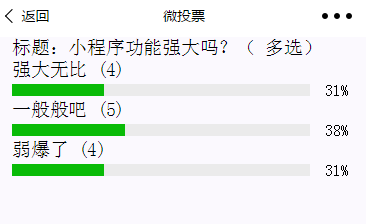
再点击分享，就会弹出：



其实很简单，只要在需要分享的页面里添加一个方法，把对应页的路径填入path就可以了，但是要注意的是path要用 / 开头：



最后一大功能是：微投票结果页



页面代码也很简单（voteresult.wxml），用个progress来显示比例：

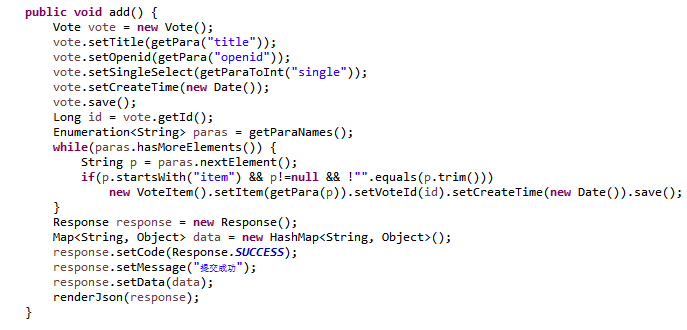


voteresult.js 的加载代码也简单，把微投票的id提交到对应的服务器上，查询返回结果：

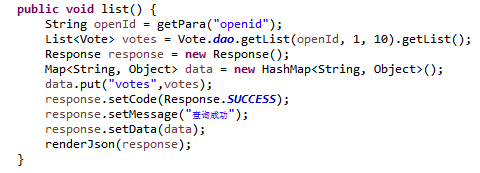


服务端控制层部分方法代码（采用了jFinal框架），只是为了示例而用，比较简单，很多逻辑和错误控制都没加上去，请读者见谅：

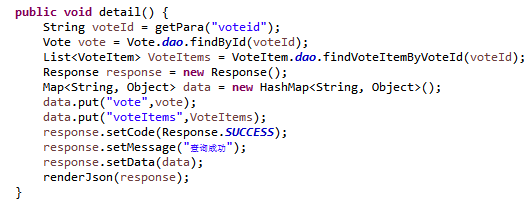
新增微投票主题：



我的微投票主题列表：



某个微投票详情：

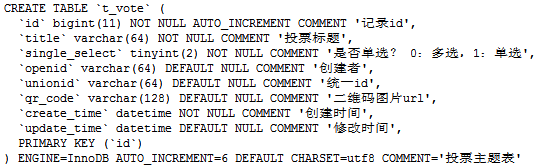


某个微投票结果页：

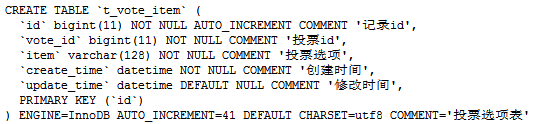


部分数据结构，也很简单：

微投票主题表：



微投票选项表：



微投票结果表

