第七章：微信小程序编辑名片页面开发

编辑名片有两条路径，分为新增名片流程与修改名片流程。

用户手填新增名片流程：

首先跳转到我们的新增名片页面 1 需要传递用户的当前 userId，wx.navigateTo 带值跳转。Manual 为 true 设置用户走的是新增路线。



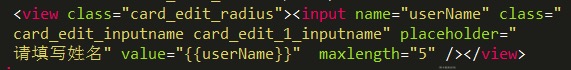
新增名片页面 1 基本布局如下：



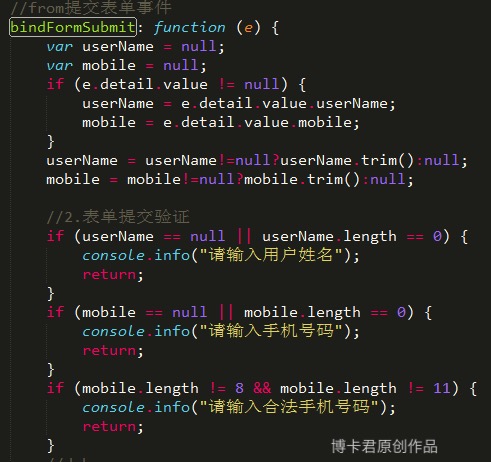
取到 userId。



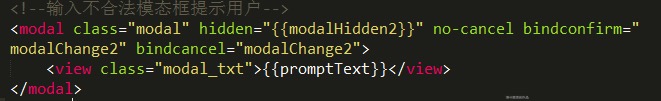
使用微信自带的 input 组件验证也非常好用，如 maxLength 属性，可以限制用户输入长度，如我这的姓名长度是最大 5 位，直接数字 5 即可。



也可以自定义一些验证效果，具体可以根据需求进行一些验证配置，取到用户输入的值，进行操作。



这里绑定了自带的模态框提示组件。



其中 modalHidden2 是模态框开关。

另外 proptText 是需要提示的内容。

即使很多输入框也支持数据动态改变，非常方便。



实际效果，非常快捷，比以前省去很多事情，编写小程序，发现最大的好处可能就是我们不必去考虑一系列兼容性问题。



最后还有个头像上传图片，测试了下目前上传到后台服务器还有点问题，应该是内测版本不太完善导致吧。



设置的直接是背景图片。



提交表单与跳转。

提交表单使用的是自带的 bindsubmit 事件组件，在 button 组件上添加 formType=”submit” 即可，还有点需注意的是使用表单提交功能时 input 需加上 name 属性，这个传递方式是以键值对的形式传递的。



这时候跳转到编辑页 2 页面，这个页面是根据用户填写的手机号码识别到匹配的公司，页面非常简单，一个数据循环而已，单选框日后可能还需要美化一下。



同样也是一些数据绑定以及验证效果。



实际渲染效果可以看到。



这个和第一个编辑页面逻辑基本不相上下，一些基本验证与提交，这里就讲到前面两步骤即可，编辑页面3也是同理，这里不再过多啰嗦。

修改名片流程效果图与需求，修改名片是一次性把以前填写的个人信息全部渲染出来，供用户来改动：



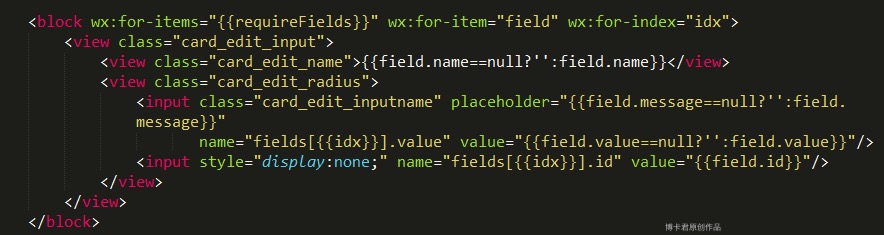
名片图片模块，上传图片暂时还有点问题，这里就是模仿了个跳转组件，比较建议需要跳转的页面还是使用 wx.navigateTo 控制好一点，wx.navigateTo 提供了给我们不同的 3 个跳转路由，封装的都很好，而且跳转页面很多牵连到传值之类的，可以达到统一管理也可以避免一些看不到的 bug 吧，总之还是根据业务需求来定：



姓名手机必填模块：



个人信息模块，直接循环（block）出来：



Onload 时我们请求必填与选填数据：

* requiredGroup 必填中文信息
* notRequiredGroup 选题中文信息
* requiredGroupEn 必填英文信息
* notRequiredGroupEn 选题英文信息

//请求名片对应的公司的中文信息的属性组数据，分为必填和选填

//选题项变量以no开头

requester.getOfflineCardInfoGroupFields(userId, cardId,

function (res) {

//debugger

var userName = res.card.userName;

var mobile = res.card.mobile;

var requiredGroup = res.requiredGroupCh;

var notRequiredGroup = res.notRequiredGroupCh;

var requiredGroupEn = res.requiredGroupEn;

var notRequiredGroupEn = res.notRequiredGroupEn;

var reqLen = requiredGroup.fields.length;

var nreqLen = notRequiredGroup.fields.length;

var reqLenEn = requiredGroupEn.fields.length;

var nreqLenEn = notRequiredGroupEn.fields.length;

self.setData({

userName: userName,

mobile: mobile,

requireFields: requiredGroup.fields,

notRequireFields: notRequiredGroup.fields,

requireFieldsEn: requiredGroupEn.fields,

notRequireFieldsEn: notRequiredGroupEn.fields,

l1: reqLen,

l2: nreqLen + reqLen,

l3: reqLenEn + nreqLen + reqLen

});

self.forceUpdate();

}, function (code, msg) {

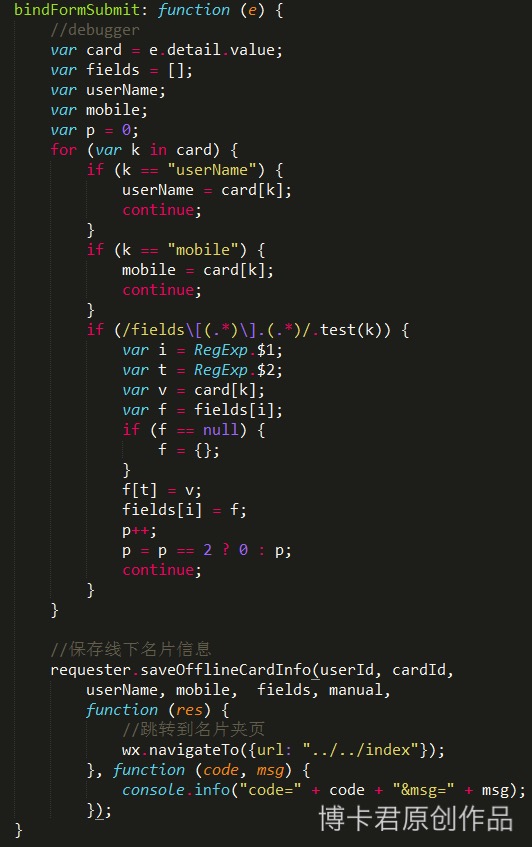
console.info("code=" + code + "&msg=" + msg);

});

中英文信息必填与选填渲染：



这里表单提交数据转换有点复杂（大家根据业务需求来做，不必花时间研究这里的方法），获取的是数组，按照后台需要的数据格式进行转换传递过去。



今天再回去理下首页 A、B、C 定点跳转功能实现方法。



首先是右边小索引布局以及数据绑定，数据绑定和名片夹列表上的字母一样，该字母下面有名片则渲染出来，没有则不需要渲染，id同样是当前字母与右边显示的内容一样：



数据 sort，和 group.name 数据一样：



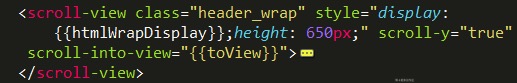
这里是因为 # 不支持设为 id（就是 id=”#”），故而进行了一个转化。

http://oe3s2czrh.bkt.clouddn.com/image27.jpg

点击事件：获取到当前 ID，以及绑定数据 toView 为当前 ID。



首先名片列表，名片上的字母索引都在 scroll-view 里面，这个 scroll-view 必须设置好固定的高度，设置成 100% 与 100vh 是无效的，y 轴的滚动开关打开，scroll-into-view 需要跳转到它子元素的 id 上。



可以看下：



这个 group.name==sortmsg，等于就是 A==A,B==B 同理。

http://oe3s2czrh.bkt.clouddn.com/image31.jpg 

在这里如顶部有些菜单栏的话，你就要注意好布局了，不然会出现向下偏移这个菜单栏的高度，其实你只要和字母索引同级下即可避免此问题（这里的顶部菜单以模板分离出去了，分离模板的时候需注意下，需要在这里绑定模板的一些数据会出现失效，具体没有继续深入研究下去）。



跳转功能基本实现（ohter 就是 # 底部）。

