

Vue+Webpack框架在银行App前端开发的应用

■ 兴业银行广州分行信息科技部副总经理 江 庆
兴业银行广州分行 叶浩荣



江庆, 软件设计师, 现任兴业银行广州分行信息科技部副总经理(主持工作), 参与过基金代销系统、集合资产销售与管理系统、理财产品销售管理系统等多个项目的建设, 并主持建设IC借记卡系统二期、手机近场支付系统(云闪付、ApplePay)等项目。

摘要: 兴油宝App是兴业银行与中石化联合推出的联名卡配套App, 是基于Cordova进行开发的。在开发初始阶段, 采用Html+JavaScript+CSS来实现每一个界面, 但随着需求增加, 代码变得凌乱, 管理也愈发困难。为此, 内部提出用Vue+Webpack对页面资源进行重新整理, 以达到维持项目迭代更新的目的。

关键词: 兴油宝; Vue; Webpack; 构建工具; 组件

随着移动互联网日益盛行, 浏览器功能日益完善, 越来越多的项目倾向于采用BS架构。为了提高开发效率和代码复用率, 越来越多的网页开发框架开始流行, 例如Vue, React, Angular等。其中由于Vue具有轻量、高性能的特点, 越来越多的爱好者将它应用于国内各种项目中, 于是就有了越来越多的共享组件。现在以兴油宝项目中使用Vue为例, 简单介绍一下Vue及其所用到的构建工具Webpack。

一、背景介绍

兴油宝App是基于混合式移动开发框架Cordova进行开发的App, 即用Html+JavaScript+CSS实现用户

界面,通信、加解密、消息推送等原生功能以插件形式为WebView提供支持。这样不但能满足安全监管的需要,还具备快速迭代的更新能力。

项目开发初期,内部决定采用iframe嵌套的形式,将固定组件提取到上层页面,可变部分由iframe加载。这种形式在一定程度上减少了重复编写固定组件的工作量,而且在切换时,能添加部分动画。如图1所示,固定部分是底部的选项卡,其他部分是可切换的,于是这两个界面就被设计为3个页面:

第一,底部选项卡+覆盖其余高度的iframe;

第二,首页其他内容;

第三,特惠商户其他内容。

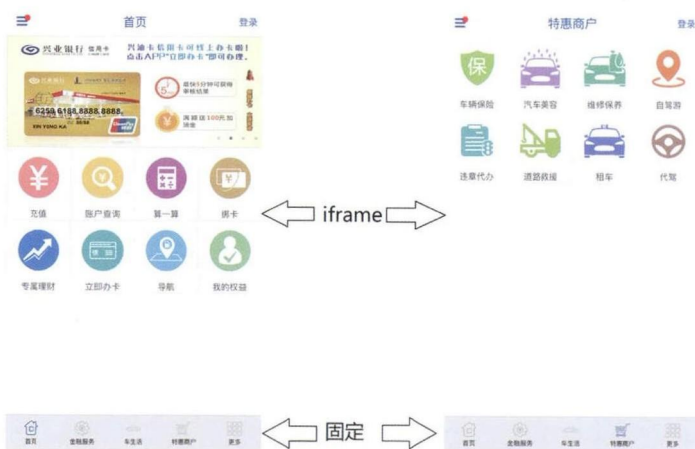


图1 iframe嵌套页面

我们采用jQuery简化JavaScript使用,并利用Bootstrap简化CSS布置,这两个都属于易学易用的资源库,可以减轻开发难度。在分工方式上,每人负责若干个页面,在前期需要不多且每个页面的复杂度不高的情况下,开发体验良好,开发进度和页面效果也不错。但随着项目需求不断增多,页面要求越来越复杂,iframe嵌套的形式显得越来越笨重,jQuery和Bootstrap也暴露其局限性。

(一)JavaScript和CSS的引用越来越乱,维护难度大

每个页面除了要引入jQuery的JavaScript文件

和Bootstrap的CSS文件外,还要引入自身的公共JavaScript文件和CSS文件,如果需要用滚动图片、上拉更新等第三方插件,也要引入相应的JavaScript的文件和CSS文件。最后,如果当前页面有特殊的样式和方法,相应的JavaScript文件和CSS文件也要被引入。这样一来,每个页面平均的引用语句有10条以上,导致大部分页面都会有多余的样式和JavaScript函数,甚至会出现污染样式,需要为页面添加样式将其抵消。浏览器的CSS引擎和JavaScript引擎都是后来的样式和函数覆盖之前的样式和函数,修改相应文件或调整引用顺序可能会导致难以预料的后果,给项目维护更新带来不便。

(二)代码风格不统一,代码管理效率低

jQuery提供了很多语法糖,每个人都可以根据偏好选择使用,例如笔者倾向于以下语法:

```
$(...).click(function(){...})
```

而有的人则更倾向于以下语法:

```
$(...).on('click',function(){...})
```

在创建DOM(Document Object Model)时,有的人倾向于利用字符串连接生成DOM字符串,而有的人则倾向于利用JavaScript数组来生成DOM字符串。还有其他很多的差异。当需求更改需要修改逻辑时,代码片段特征不统一将不利于定位需要修改的代码片段,容易出现疏漏,从而影响项目的升级迭代,影响用户体验。

(三)容易引发兼容性问题,调试复杂度大

尽管用户界面是用Html+JavaScript+CSS实现的,但是Android和iOS的渲染引擎与JavaScript引擎存在差异,不同的Android版本之间和不同的iOS版本之间的渲染引擎和JavaScript引擎也有差异,导致App在手机上运行的时候得不到预想的效果。最明显的例子是Click事件,Android对Click事件响应很快,而iOS对Click事件的响应则有延时,要用Touchend事件才能满足需要。然而,滥用Touchend事件的后果是拖动页面

时, Touchend事件容易被误触发, 严重影响用户体验。最严重的是不兼容的CSS甚至会导致App进入假死状态, 且调试CSS十分困难, 因为遇到错误时没有友好的提示, 而且对于浏览器来说这样的CSS没有错误。以上问题在开发过程中给我们造成了很大的困难。

Web App的飞速发展催生了很多优秀的页面开发框架、资源库和构建工具, 例如Facebook的React, Google的Angular、国人尤雨溪的Vue, Webpack等, 这些框架和资源库在资源引用、代码管理和兼容等问题上都有很好的对策。为了开发出更好的产品, 我们用Vue+Webpack对页面资源进行重新整理, 以达到维持项目迭代更新的目的。

二、Vue简介

Vue是一个构建Web界面的JavaScript库, 它的理念是数据驱动, 能通过尽可能简单的API实现响应式的数据绑定和组件式的视图组合。

响应式的数据绑定是指通过特殊语法将DOM“绑定”到底层数据, 使DOM与数据保持同步, 一旦数据被修改, DOM会相应地被更新。绑定的语法非常简单, 如图2所示, 将一个Cell组件的Title属性和Value属性分别绑定到变量CardType和CardBalance, 再将一个Spinner是否显示的状态绑定到变量CardBalance。页面初始化时, CardType和CardBalance的值都为“”, 页面渲染结果如图3所示。当App从服务器获取数据后, CardType的值被置为“理财卡”, CardBalance的值被置为“41.14”, 页面渲染的结果就变为图4。

Vue实现响应式的数据绑定是利用ES5的Object.

```
<body-container>
  <group>
    <cell :title="cardType+ ' 余额 ' " :value="cardBalance">
      <spinner type="ios-small" v-show="cardBalance==''"/>
    </cell>
    <cell :title="cardType+ ' 加油卡余额 ' " :value="oilBalance">
      <spinner type="ios-small" v-show="oilBalance==''"/>
    </cell>
  </group>
</body-container>
```

图2 响应式的数据绑定语法

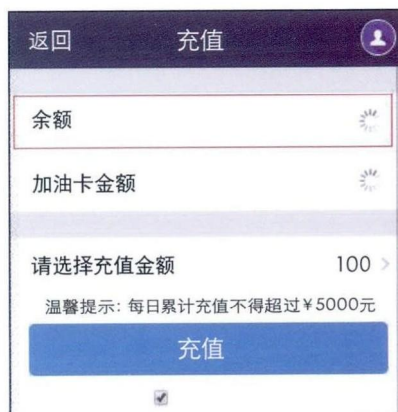


图3 页面初始化时的渲染结果

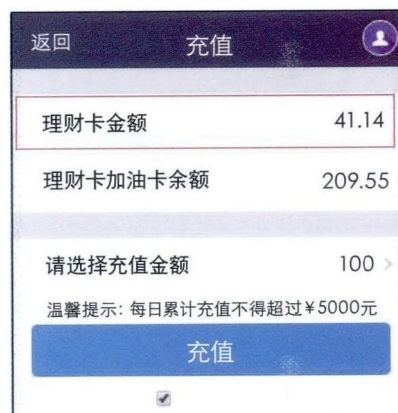


图4 App获取数据后的渲染结果

DefineProperty特性。Vue实例化时会用Object. DefineProperty为data对象的属性设置getter和setter函数, 读取属性的值时, Getter函数会被触发, 实际读取到的是Getter函数的返回值; 设置属性的值时, Setter函数会被触发, Setter函数里更新绑定DOM的逻辑就会被执行。

Vue将组件绑定到数据的特性能使JavaScript代码量减少约50%。一是因为重新对对象赋值就可以更新多个DOM, 不再需要为每一个需要更新的DOM

的相应属性和节点值编写相应的设置语句;二是因为需要动态创建DOM时,用Vue的V-for命令可以很方便地进行列表渲染,不需要编写生成复杂DOM的JavaScript代码,降低了调试的工作量和阅读的难度。

组件式的视图组合即将应用界面抽象为一个组件树,用独立可复用的小组件来构建大型应用。我们将界面分为5个组件,其中Title组件和Tabbar组件在其他界面都有复用,Swiper组件在这个界面里也实现了复用,在Swiper组件和Tabbar组件里能再抽象出子组件,增加复用深度,减少大量重复代码,代码量大幅下降,如图5所示。



图5 组件式视图组合

利用Vue视图组件的概念和Webpack将分散资源打包的能力,代码组织变得十分清晰。我们坚持一个文件包含一个组件的原则,明确该组件依赖于什么组件、JavaScript文件、CSS文件或其他资源,管理效率有所提高。此外,样式可以做到按需注入,及设置样式的作用域仅限于当前组件,不再需要担心其他组件的样式会对当前组件的样式造成干扰,可以专心于当前组件的开发。如此一来,团队内代码冲突减少,团队开发效率大大增加。

接下来笔者以兴油宝项目为例介绍Vue使用过程的要点。

三、Vue-cli开发介绍

首先,用Vue-cli搭建Vue+Webpack的开发环境。Vue-cli是官方出品的脚手架工具,在使用Vue构建大型项目时,使用Vue-cli是最快的构建方式。利用Vue-cli搭建开发环境时,Vue-cli会从Github下载开发环境的依赖包列表和相关配置文件。如果需要通过代理下载,可以修改% NODE_PATH%/node_modules/vue-cli/bin/vue-init的第121行,在调用Request函数的第一个参数里新加一个属性:

```
proxy: 'http://proxy-server:proxy-port',
```

其次,宏观整个项目,抽象出最基本的组件。如图6所示,将Tabbar、返回按钮、用户头像放在第一层,每一个界面的主要元素分别放在一个组件里,用Router-view进行切换。接着每一个界面都分为Title和Body两个组件,抽象出Title组件是因为方便统一控制样式,抽象出Body组件是为了统一控制页面高度。之后对具体界面进行具体分析图5所示,复用Swiper组件构建了广告、功能按钮和推荐3个模块。而且在各个模块中再抽象出子组件,例如功能按钮模块中抽象出由图片和文件组成的按钮组件。为了快速构建,选用Vux组件库构建界面,Vux是一个对WeUI用Vue封装的JavaScript资源库,它针对移动浏览器做了很多兼容性处理,笔者推荐使用。

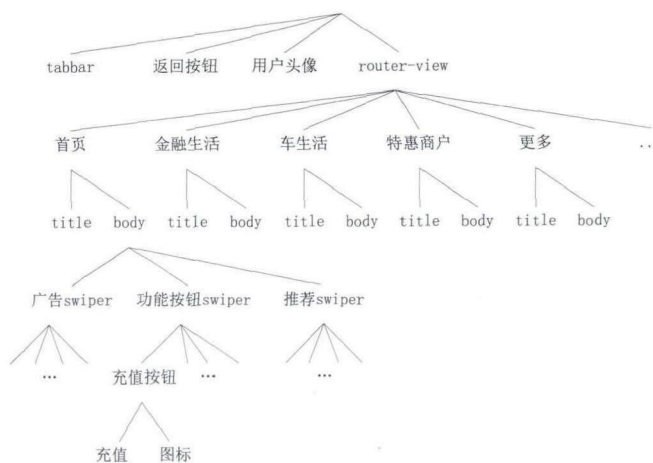


图6 抽象出最基本组件

最后, 打包生成出浏览器能识别的前端资源。利用Vue-cli搭建的Vue+Webpack开发环境默认配置导出的资源默认引用路径是根目录, 由于兴油宝采用Cordova框架, 前端资源只能放在应用的专用目录下, 所以默认配置不能应用于该项目。通过研究Webpack的配置选项, 修改build/webpack.prod.conf.js的第21行, 在Output对象里加一个属性:

```
publicPath: '/',
```

默认配置下打包时, 会默认生成Map文件, Map文件里写着对应的JavaScript文件是由哪些资源构成, 对应用部署没有作用。如果不想生成Map文件, 可以将config/index.js里ProductionSourceMap的值改为False。

至此便能得到所需的页面资源, 打包进Cordova便可使用。

四、总结

Vue+Webpack对前端页面资源重新整理后, 代码

量由原来的2.73 M减至0.59 M, 约为原来的22%, 大大减少了工作量。将包括图片在内的页面资源打包后, 资源大小由原来的5 M减至3.85 M, 约为原来的77%。通过Chrome调试发现相同内容的界面占用内存节约1 M左右。重新整理的结果见表1所列。PTT

表1 前端页面资源整理前后对比

时段 内容	原来	现在
代码量	2.73M	0.59M
资源包大小	5M	3.85M
平均内存	8.2M	7.2M

参考文献:

[1] 尤雨溪. 前端——开源前端框架纵横谈[J]. 程序员, 2013(3): 24-29.

[2] 龚明军, 余军. 移动金融安全框架初探[J]. 金融科技时代, 2016(3): 23-26.

