

浙江大学

硕士研究生读书报告



题目 分析基于 weX5 的开源框架

作者姓名 詹敏

作者学号 21651149

指导教师 李启雷

学科专业 移动互联网与游戏开发

所在学院 软件学院

提交日期 二〇一六年十二月

Analysis of open - source framework based on WeX5

A Dissertation Submitted to

Zhejiang University

in partial fulfillment of the requirements for

the degree of

Master of Engineering

Major Subject: Software Engineering

Advisor: LiQiLei

By

ZhanMin

Zhejiang University, P.R. China

2016

目录

摘要	I
Abstract	II
1. Hybrid 开发模式的发展历程	1
1.1 Hybrid 开发模式的诞生	1
1.2 从“重混”到“轻混”的转变	1
2. 高性能 HTML5 APP 跨端框架	2
2.1 HTML5 App 的框架结构	2
2.2 WeX5 的框架选择	2
3. WeX5 可视化快速开发工具	5
3.1 快速开发	5
3.2 极致体验	5
3.3 多端运行	5
3.4 开源免费	5
4. 对 WeX5 的学习重点	6
4.1 组件编程	6
4.1.1 组件的创建	6
4.1.2 组件的属性	6
4.1.3 组件的方法	6
4.1.4 组件的事件	7
4.2 数据绑定	7
4.2.1 规则表达式	7
4.2.2 绑定表达式	8
4.3 前端路由和页面跳转	9
4.3.1 路由模块 (Router)	9
4.3.2 页面跳转	9
5. 从实践开发中学习 WeX5	10
5.1 开发账目列表	10
5.2 开发记一笔	11
5.3 开发账目分类设置页	12
5.4 开发首页	12
6. 总结与展望	14
参考文献	15

摘要

随着移动互联网的崛起，人们对移动开发越来越重视，从而产生了对移动开发模式的探讨，即 Native APP、Web APP 和 Hybrid APP。此文重点关注了目前十分流行的 HybridAPP 开发模式，而它的开发历程也从“重混”模式过渡到“轻混”模式。根据 2016 年国产开源软件 Top100 数据显示，WeX5 这一高性能高效率的 H5 APP 开发工具位列第二十一，并且凭借其开发的应用均能“一秒打开”，一次开发，多平台运行的特点得到了广大开发者的青睐和支持。

本文从 Hybrid 开发模式的发展历程开始介绍，然后介绍 HTML5 APP 的框架以及 WeX5 的框架，接着叙述 WeX5 作为快速高性能的开发工具的显著特征，再讨论我在学习 WeX5 开发时所关注的重点，最后通过我在学习 WeX5 时所开发的例子中，阐述我对 WeX5 的理解和学习收获。

关键词：WeX5， Hybrid APP，移动开发

Abstract

With the rise of mobile Internet, people pay more and more attention to mobile development. And it leads to the development of the mobile model, such as Native APP、Web APP and Hybrid APP. This article focuses on the popular development model named as Hybrid APP. And its development process has changed from the "remix" model to the "light mix" model. According to the Top100 open source software made in 2016 report, WeX5 which is the high-performance efficient H5 APP development tool ranked twenty-first. With its development of applications can be "one second to open" and the characteristics of multi-platform operation, WeX5 has owned the favor and support from the majority of developers.

This paper begins with the development of hybrid development model. Then I introduce the framework of HTML5 APP and the framework of WeX5. WeX5 is then described as a rapid and high-performance development tools. And then I discuss the the key concerns in study of WeX5, and finally I described my understanding of WeX5 and learning gains through my learning example of the WeX5 development.

Keywords: WeX5, Hybrid APP, Mobile Developme

1. Hybrid 开发模式的发展历程

1.1 Hybrid 开发模式的诞生

自苹果和谷歌分别发布了 iPhone 和 Android 手机开始，我们就进入了一个移动应用的新时代。随之而来便产生了一个新的问题——“一个应用两套实现”，这对开发成本和维护成本都是非常大的挑战。基于这样一个现实，其解决方案就是想办法实现一套代码跨端域运行，因此 Hybrid APP 混合应用开发模式应运而生。在 Hybrid APP 整个开发框架上，有各种各样的框架和技术类型，而现在主要的技术就是 Web 技术，不仅简单，而且容易实现，开发者人数也比较多。

1.2 从“重混”到“轻混”的转变

在 Hybrid APP 框架中大部分使用的都是 Web+Native 组合模式。在它发展早期，Web 的运行性能是最主要的瓶颈，因此开发者使用 Native 来弥补 Web 性能上的不足，实际就是多 WebView 和 Native UI 组件的混合开发。然而混合的单 WebView 在页面切换时会有严重的闪白和延时，用户体验并不好。所以大多数开发者都选择采用多 WebView，从而实现平滑转场。

因为早期硬件比较差，浏览器性能也一般，所以有些比较复杂的组件在实现功能的时候速度比较慢，所以使用 Native UI 组件来弥补，配合 Web 实现功能，这种模式称作“重混”。它的优缺点也很明显，优点是提升了运行性能，增强了交互。缺点是 Web 和 Native 技术交叉混杂，增加了开发人员的工作难度。

而且随着技术的发展，手机硬件配置有了很大的改善，浏览器技术也有了长足的进步。“轻混”模式便成了 Hybrid APP 的必然选择，它的 UI 部分必须用纯 Web 技术，而底层的设备接口上，需要 Native 技术弥补。并且“轻混”框架的 Native 的技术是不应该去侵入 UI 的，这也它是和“重混”之间的最大区别。

2. 高性能 HTML5 APP 跨端框架

2.1 HTML5 App 的框架结构

HTML5 APP 简单理解就是以网页技术为主来实现的移动应用。轻混模式下的 HTML5 APP 由网页和外壳两部分组成。网页主要负责界面的显示和交互，而外壳会内置一个浏览器来提供网页的运行环境，并会通过插件为网页提供扩展的原生调用能力。



图 2.1 HTML5 App 框架结构

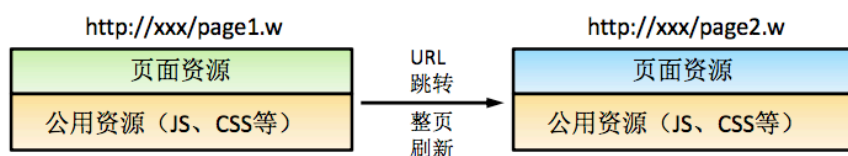
2.2 WeX5 的框架选择

WeX5 作为一款典型的开源“轻混”H5 移动开发框架，需要解决两个框架的问题，一个是 HTML5 的框架，一个是 Native 的框架。

2.2.1 H5 页面框架的选择

在不使用重混架构的前提下，为了解决 HTML5 的一些性能问题，例如页面切换的障碍问题等，WeX5 采用 SPA 单页应用模式，它是基于传统的页面加载模式 MPA，页面之间相互独立。但是 SPA 的不同之处在于，其框架里整个页面是由外壳页面框架组成的，使用 AJAX 技术完成，通过局部刷新来提升用户体验，并且整个页面框架都是用 AJAX 来实现。

多页应用模式MPA (Multi Page Application)



单页应用模式SPA (Single Page Application)

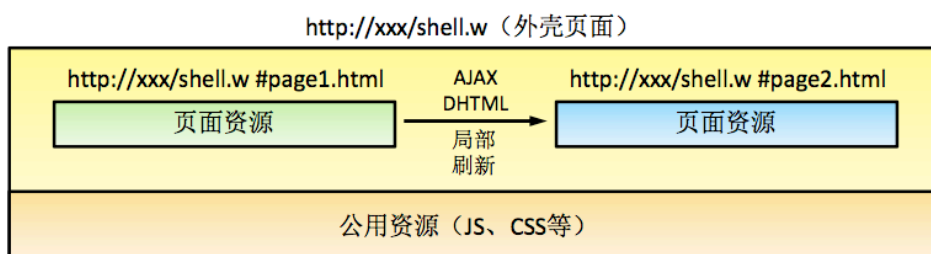


图 2.2 多页应用模式与单页应用模式对比

虽然 SPA 在性能方面能提升很多，但是在多人协作过程或开发复杂应用是，会出现 ID 冲突，样式冲突，JS 冲突等问题，因此 WeX5 通过页面隔离来解决这样的问题。

- (1) 元素 ID 隔离：因为 WeX5 是可视化工具，所以用 XID 来替换，不直接设定 ID 属性。这样到内存里，会动态生成真实的 ID，会在 XID 后面骗一个页面标志，这样就可以保证多人写的页面在加载内存里的 ID 是不相同的。
- (2) CSS 样式隔离：每个页面都有一个同名的 CSS 文件，只作用于当前页面。通过对工具的编译，把私有 CSS 文件里的所有样式加一个页面标志，在页面节点的属性上加一个标志，这样就使得 class 只能作用域当前页面的元素。
- (3) JavaScript 隔离：借用 JavaScript 模块化技术，如 RequireJS，通过 define 定义模块，require 来加载模块。无论是方法还是变量，都封装在闭包里，每个代码都是写在 define 的模块里，这样就把代码自然隔离了。

2.2.2 Native 框架的选择

Cordova 框架作为受到业内主流厂商的支持并推崇的开源移动跨端框架，它在技术实现上也 Web+Native 的框架，它提供了义序设备相关的 API，通过这些 API，移动应用能够以 JavaScript 访问原生的设备功能，如摄像头、麦克风等。

此外，WeX5 选择使用 Cordova 作为自身的原生框架还可以很方便地定义自己的插件或者扩展其自带的插件，从而实现想要的功能。同时，Cordova 还是一个非常“干净”的轻混合的跨端框架，它是不侵入 UI 的，只要负责原生设备的调用。

2.2.3 WeX5 整体技术架构

基于上面的分析和理解，WeX5 的整体框架可用下图来表示：

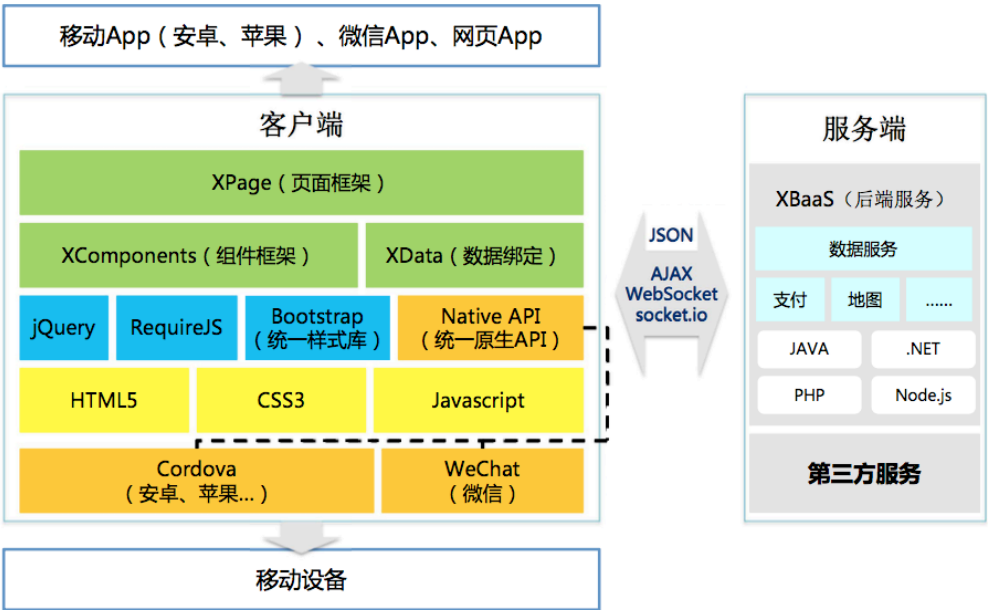


图 2.3 WeX 整体框架图

最底层部分用的是 Cordova，解决安卓和苹果的原生调用问题。同时要兼容微信，所以上面把 Cordova 和微信又做了封装，抽象成统一的 HTML5 API。通过 Cordova 接口调用，可以更方便的实现一次开发，多端运行，代码不需要改，既可以运行在原生 App 里，也可以运行在微信里。

Bootstrap 在 WeX5 中也发挥了其前端优化的能力。一个是样式美化，扁平化风格，另外响应式布局。基于 Bootstrap 设计的页面，运行在不同的设备上不需要考虑分辨率，会自动处理设备分辨率。再上面实现了 WeX5 的组件框架和数据框架，页面上不仅有交互的 UI 组件，页面里面还有数据。接下来是业务框架层，即 SPA 单元页面框架。

在服务端 WeX5 还提供了 XBaaS 服务，负责后端数据存取、逻辑，还有第三方地图、支付等功能实现。WeX5 提供多语言实现，提供了不同语言的版本，开发者可以针对自己的版本来集成到自己的框架里。

3. WeX5 可视化快速开发工具

3.1 快速开发

WeX5 是基于 HTML5 的开源跨端可视化开发工具，它拥有上百个页面组件，可视化页面设计以及模板化快速开发能够极大提高开发效率。编码、调试、发布一体化的集成开发环境，无需原生开发经验就能通过 WeX5 开发出自己想要的 APP，大大降低了开发的成本。

3.2 极致体验

WeX5 拥有时尚的扁平化样式风格，能够满足大部分使用者的视觉体验，它的响应式布局可以自适应分辨率，从而解决了多设备多手机屏幕不适配的问题。它内置的触摸滑动等手势操作，页面的急速加载足以媲美原生的交互体验。此外，它还拥有完整的原生设备能力，支持地图、分享、消息、支付等功能。

3.3 多端运行

WeX5 对跨平台多前端支持极好，一次开发多平台运行。目前，WeX5 支持的前端有：移动 APP（苹果 APP 和安卓 APP）、微信应用（包括公众号、服务号和企业应用号）、PC 浏览器和其他轻应用（如百度直达号）。WeX5 的后端完全开放，可通过 http 和 websocket 等协议连接各种后端中间件或云服务（Java、PHP、.NET 等）。

3.4 开源免费

WeX5 采用 Apache 许可证开源模式，商业友好，完全免费。基于 WeX5 开发出来的应用，每一行代码都在你手里，发布部署无任何限制，可以部署在任何公有云或私有云服务器上。

4. 对 WeX5 的学习重点

WeX5 开发主要分为三个部分：前端 UI、后端服务、前端设备 API 开发。在此，我主要是研究了前端开发的一些要点，下面将会举例我个人对 WeX5 前端学习过程中的一些总结归纳。

4.1 组件编程

WeX5 的界面是由组件构成的，组件是一个可以以图形化方式显示在屏幕上，并能与用户进行交互的对象。组件是对数据和方法的封装，有自己的属性、事件和方法等。

4.1.1 组件的创建

WeX5 创建组件共有两种方式：一是直接在集成环境中摆放组件模型，二是动态创建组件。第一种比较简单，在此就不做过多介绍。第二种主要是应用在运行时需要创建组件的场景中使用。它也有两种方法：第一种方法，在参数中给出父节点 `parentNode`，在 `new` 时会在父节点下直接创建出新的组件；第二种方法，不给出 `parentNode` 参数，在 `new` 时创建组件，调用 `addComponent` 方法后将新的组件加入到页面中。

4.1.2 组件的属性

WeX5 开发工具提供了非常友好的可视化前端开发工具，不但有各类模板组件，而且可以很方便地修改和配置其属性。共有三种方式：一是在设计器中设置；二是使用表达式配置；三是使用 `js` 代码修改。

在设计器中可以分别通过属性编辑器或通过右键菜单中设置。组件都有 `xid` 属性，在同一个 `w` 文件中不能重复，通过 `xid` 可以获取组件的 `js` 对象和 `jQuery` 对象。注意：WeX5 使用单页模式，系统会根据 `xid` 给组件生成不重复的 `id`，因此不要自己增加 `id` 属性，否则容易造成 `id` 重复。

4.1.3 组件的方法

WeX5 的组件方法分为公共方法和私有方法。其中公共方法中最常用的是 `湖区` 和设置组件的属性。这里需要注意的一点是，`set` 方法可以一次修改多个属性，所以它的参数是 `json` 值，而 `get` 方法一次只能获取。此外，组件还提供了 `on`

方法用于添加事件处理器，提供了 off 方法用于取消事件处理器。在私有方法中，data 组件的方法尤为重要，因为 WeX5 的组件是数据感知组件，只需针对数据今夕编程即可。

4.1.4 组件的事件

WeX5 中 on 开头的事件是组件特别提供的事件，在触发和执行效率方面都优于标准的 dom 事件。例如：onclick 方法是组件提供的回调型事件，效率优于 bindclick 方法，因为 bindclick 方法是标准的 dom 事件，相当于在 HTML 中写 onclick 方法。组件事件中还提供了 event 参数，它是一个 json 对象，其中包含这个事件中需要用到的参数信息。

4.2 数据绑定

WeX5 数据绑定(Data Binding)是基于开源 knockout.js 优化改进的前端 MVVM 库。WeX5 在数据和视图两个层面的组件上，都提供了大量可通过表达式来定义的动态属性。当表达式依赖的数据项改变时，表达式会自动重新计算，并通过属性动态影响组件的显示和行为。我们把页面上的这种由数据驱动界面动态交互的能力统称为“数据绑定”。

它是典型的双向数据绑定模式，即如果可监控对象的值变化了，会自动通知到所绑定节点属性；如果节点属性变化了，也会自动改变可监控对象的值。

由于在数据和视图两个层面，表达式属性的作用有所不同，因此把表达式属性分为两类：

- 规则表达式：用于定义数据组件上的只读、必填、计算和约束等数据规则；
- 绑定表达式：用于定义视图组件上的数据、状态、样式等各种动态交互属性。

4.2.1 规则表达式

规则表达式主要是用于定义数据组件的，其中包括对整个数据集生效的只读规则以及对数据列生效的只读、计算、必填和约束规则。然而并不是任何一个组件都能感知数据规则，因为只有通过 ref 属性绑定到数据的视图组件才能感知数据规则。也不是任何一个数据变量的改变都会自动触发表达式的运算：这里的数据特指“可观察(observable)”数据。“可观察”数据不是一个简单变量，而是一个

特殊的数据对象，当它的数据域改变时会自动刺激所有依赖它的表达式重新计算。

我们在表达式对话框里，经常会看到每个列都有 `ref`、`val` 和 `label` 三个函数，它们的区别如下：

- **val**: 返回列的数据值，用于表达式计算（由于这些数据本身是“可观察”数据，所以当数据值改变时，会自动触发所依赖的表达式规则重新计算）。
- **ref**: 返回列的“可观察”数据对象，用于视图组件的 `ref` 绑定。
- **label**: 返回列的名称，一般用于页面显示列名标签。

4.2.2 绑定表达式

绑定表达式主要是视图组件上定义的动态表达式。虽然我们从设计器上来看，绑定表达式是视图组件的属性，但其实绑定表达式可以作用于任何一个 HTML 标签。当然，我们也可以认为每个 HTML 标签本身就是最简单的组件，而其他复杂组件都可以看做是基本 HTML 标签的组合。

绑定表达式的能力极其强大，可以动态定义 HTML 标签的所有属性，甚至可以通过表达式定义动态创建 HTML 片段，可以实现各种复杂的界面动态交互。当绑定表达式依赖的数据项改变时会重新计算，并动态更新对应 HTML 标签的 DOM 结构，从而影响页面的显示，而这一切仅需要定义一个表达式就搞定。

在前面的数据规则表达式中我们提到过，视图组件必须通过 `ref` 绑定到数据才能感知数据规则。`ref` 绑定是一种高级的复合数据绑定，它可以为组件同时带来多种数据感知能力：

- **双向数据感知**: 数据改变会自动更新组件显示，组件编辑后也会自动更新数据。从界面交互效果上来看，就是多个关联到同一个数据的组件，一个数据编辑后其它组件都会自动更新。
- **数据规则感知**: 组件可以通过 HTML 状态和样式的改变将只读、必填、约束等规则提示给用户。
- **数据类型感知**: 这里特指 `input` 组件通过 `ref` 绑定数据列后，可以感知列的数据类型。例如：当列是日期类型时，`input` 自动变成日期编辑模式；当列是整型数值时，`input` 只能输入数值。

4.3 前端路由和页面跳转

4.3.1 路由模块 (Router)

Router 是一种根据不同 URL 来表现的页面状态的能力，基于 HTML5 history API 实现。WeX5 中的 Router 模块提供了根据不同组件 state、model 的 states、以及自定义 states 来实现的对浏览器的前进后退导航、有状态的书签、可分享的 URL、URL 自定义处理器等支持。

如果说组件就是一个状态机，根据不同的状态响应不同的逻辑、展现，那么每个组件可以把自己的状态反映到 URL 中，但是 WeX5 的页面是有很多 model (可以认为一个 .w 页面就是一个 model) 组合出来的，所以每个 model 需要维护当前 model 下的状态，那么每个 model 下就有一个对象 \$routeState。

model 的状态变化了，它有可能是根 model (比如在 portal 中的 portal 页面)，它也有可能是子 model (比如 portal 中打开的页面，windowDialog 打开的页面等)，那么这些状态的合并、向上传递以及父 model 收到变化通知要向子 model 分发等问题就需要调用 router.js

4.3.2 页面跳转

页面跳转可以是打开新的一页，也可以是当前页重定向，还可以是跳转到当前页的某个状态，下面会从打开 w 文件，打开其它后缀文件，使用路由等几个方面分别介绍一下在 X5 中的页面跳转。

(1) 打开 w 文件

打开 w 文件有两种最常见的两种方式，一是使用 shell 提供的方法；二是使用弹出窗口打开。前一种方法打开子页面不需要返回，可以直接调用 showPage 和 closePage 方法。后一种方法打开子页面并需要返回，需要在父页面中添加 windowDialog 组件，并调用 windowDialog 组件的 open 方法打开子页面，在子页面中，需要使用 window 组件的 close 操作或者 this.close() 方法关闭子页面。

(2) 打开 html 等其他类型的文件

为了打开其他类型的文件，有以下几种方法：1. 可以使用类似打开 w 文件的方法，即放一个 iframe 标签，设置 src 属性为要打开文件的路径；2. 使用 a 链接跳转，设置 href 属性，通过点击 a 标签跳转页面；3. 页面重定向，通过设置 window.location.href 实现页面跳转；4. 使用 window.open 打开页面。

（3）使用路由

Router 是一种根据不同 URL 表现不同页面状态的能力，提供对浏览器的前进、后退导航、有状态的书签、可分享的 URL、URL 自定义处理器等。能够实现把这个有状态的 URL 分享出去，打开这个 URL 后这个页面能定位到相应状态。这就是前端路由，虽然 URL 变化了，但是并没有发送后端请求。

5. 从实践开发中学习 WeX5

为了更加有效地掌握 WeX5 的开发方法，深入理解 WeX5 框架的概念原理，我通过实现一个记账本 APP，掌握了数据的增删改查、列表和表单的制作、shell 及对话框的使用。这个 APP 主要实现了账目列表、记一笔、分类设置三个功能。运行效果如下图：

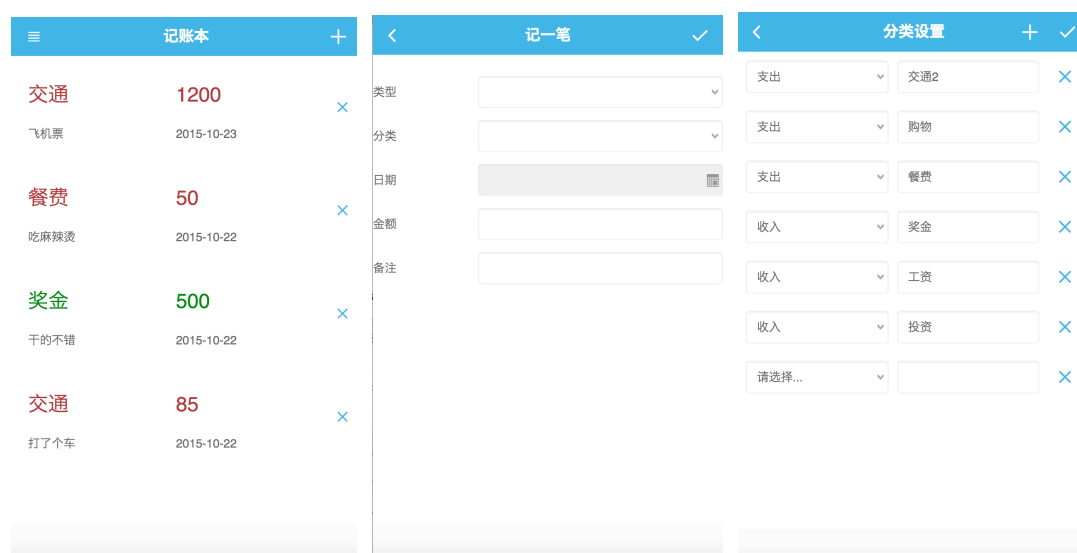


图 5.1 记账本 APP 效果图

5.1 开发账目列表

我在开发账目列表这一模块过程中，认为关键之处就在于将数据库中的数据与 list 组件进行绑定，为了访问数据库数据，我使用了 baasData 组件，从而调用后端服务，获取账目表中的数据。同时，为了达到下拉重新加载数据的功能，我还使用了 scrollview 这一视图组件实现上滑加载的功能。下图展示了嵌套层次摆放的组件。

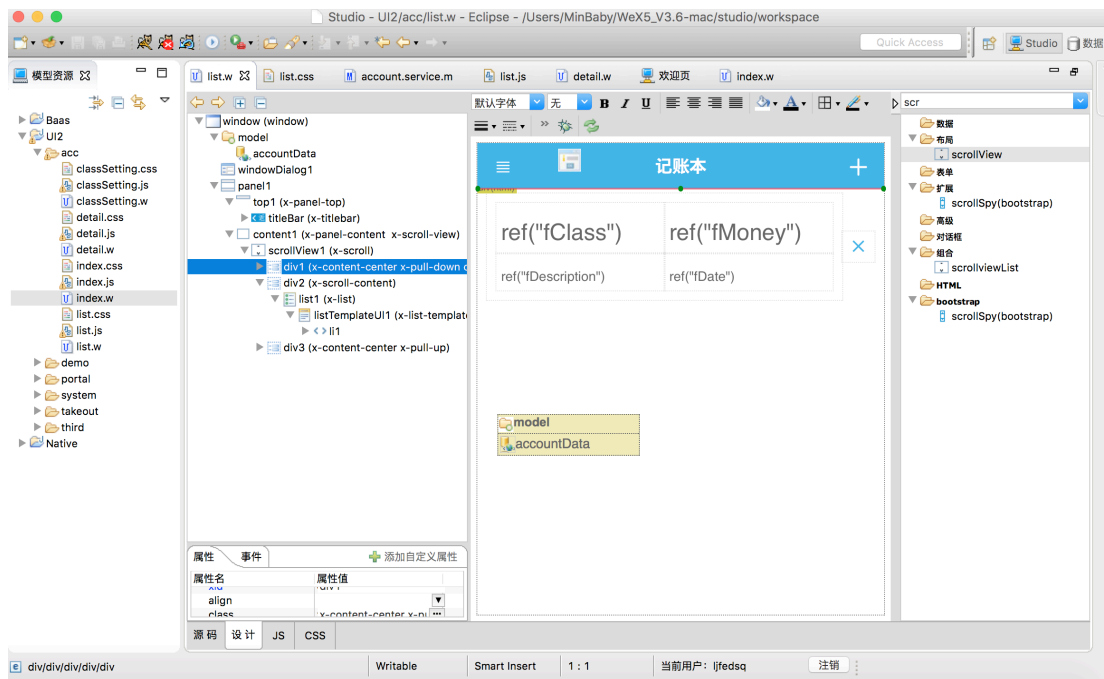


图 5.2 账目列表页设计界面

5.2 开发记一笔

记一笔这个功能较为复杂，包含新增、修改账目数据。新增时，需要一条空数据；编辑时，数据来自账目列表页，所以不需要调用后端服务获取数据。同时，这里还涉及了页面之间的跳转，并且是带数据返回的页面跳转，因此，我选择使用组件 `windowDialog` 来实现。

在添加新增账目时，点击新增按钮时，触发按钮的 `click` 事件，调用 `windowDialog` 组件的 `open` 方法，并传递一个“new”的参数，在记一笔页面中，接收这一参数，给账目数据组件增加一条新记录。在新记录中输入相关数据后，关闭页面返回账目列表，可看到新增的数据。如果是修改数据，则直接点击列表中的数据，触发列表中组件的点击事件，跳转到编辑页面，并把数据传递过去。

`list.js` 中添加记录的关键代码如下：

```
Model.prototype.addButtonClick = function(event) {
  this.comp("windowDialog1").open({ //获取账目列表中的windowDialog组件
    params : {
      operator : 'new'
    }
  });
};
```

`list.js` 中修改记录的关键代码如下：


```

Model.prototype.editacc = function(event) { // event参数有一个绑定的上下文, 其中的$object能获取所在行的数据
    var row = event.bindingContext.$object;
    this.comp("windowDialog1").open({
        params : {
            operator : 'edit',
            rowData : row.toJson()
        }
    });
};

```

detail.js 中添加、修改记录的关键代码如下:

```

Model.prototype.modelParamsReceive = function(event){
    var data=this.comp("accountData");
    data.clear();
    if(this.params.operator=='new'){//接收新增参数, 添加一条数据
        data.newData({
            "defaultValues" : [ {
                "fID" : justep.UUID.createUUID(),
                "fCreateTime" : new Date()
            } ]
        });
    }else{
        data.loadData([this.params.rowData]); //接收的参数不是新增, 则为修改, 将list中的数据加载过来
        data.first(); //将第一行设置为当前行
    }
};

Model.prototype.OKBtnClick = function(event){
    this.owner.send(this.comp("accountData")); //将修改或新增的数据发送到数据组件中
    this.close();
};

return Model;

```

最后, 修改和添加的数据只是存在了前端的数据组件中, 并没有存入后端的数据库中, 因此还需要调用 saveData 方法。

list.js 中需要添加以下代码:

```

Model.prototype.windowDialog1Received = function(event) {
    this.comp("accountData").saveData();
};

```

5.3 开发账目分类设置页

账目分类设置页主要也是实现新增、修改、删除账目分类。和上面的账目列表功能类似。账目类型是下拉选择的数据, 所以需要提供账目类型数据, 用于下拉选择。账目类型分为收入和支出, 是静态数据, 因此使用 data 组件; 账目分类则需要数据组件 baasData, 通过调用后端服务, 获取账目分类表中的数据。

5.4 开发首页

除了用 `windowDialog` 组件进行页面跳转外，还可以用 `Shell` 方式打开一个 `w` 文件。WeX5 提供的 `Shell` 可以给 App 增加左（右）边栏；也可以相应安卓手机的物理返回键，实现按两次返回键退出 APP 的能力。

在使用首页向导建出的页面文件中，用 `wing` 组件可以实现 APP 的左（右）边栏；用多内容页组件 `contents` 实现 `Shell` 的页面跳转。

6. 总结与展望

通过学习 WeX5 开源框架，我掌握了其框架结构以及它的实现方式，并通过学习它的双向数据绑定和组件化开发理念，发现目前虽然有很多类似的移动互联网开发框架，但是都有一个共通的特点，就是简洁化和高效化，这极大地节省了开发者的学习成本，以及降低了维护更新软件的成本。因此，虽然现在的框架既是多样化的，但它们又是逐渐走向统一的，无论是在追求性能方面，还是用户体验上，都在不断地进步和发展，所以我们要不断创新不断努力，通过不断学习前沿领域技术，打牢基础，并找出自己的创新点，为了我国的软件事业做出自己的贡献。

参考文献

- [1]. 王洁. Hybrid App 走向“轻混”:剖析 WeX5 开源高性能 H5 App 开发框架. <http://www.infoq.com/cn/presentations/the-analysis-of-wex5-open-source-high-performance-app-h5-development-framework>, 2016-06-17.
- [2]. 姜玉龙. 浅析 HTML5 的技术与应用模式. 无线互联科技 第 22 期, 2016-11
- [3]. 熊亚蒙. HTML5 在移动互联网中的应用研究. 无线互联科技 第 22 期, 2016-11
- [4]. 飞扬. 2016 国产开源软件 Top100. 互联网周刊, 2016-04-20