

# 融合移动互联网应用的响应式 Web 开发模型设计\*

张烈超

(武汉交通职业学院, 湖北 武汉, 430065)

**摘要:** 文章结合当前流行的移动开发模式, 给出融合传统 Web 开发模型、移动互联网 APP 智能客户端及响应式 Web 技术的混合跨平台开发模型, 并针对不同的 Web 应用给出适合的开发设计思路及建议。此外, 文章还对模型中跨平台 Web 技术的应用进行了测试, 取得较好的效果, 以期从事移动互联网 Web 应用开发学习者提供新思路。

**关键词:** 框架; 响应式 Web; 移动 APP 客户端; 混合跨平台智能终端

DOI: 10.3969/j.issn.1672-9846.2017.01.019

中图分类号: TP311.52

文献标志码: A

文章编号: 1672-9846(2017)01-0078-05

随着互联网技术的不断发展, 融合传统移动通信、互联网及各种智能终端的创新服务新模式的移动互联网技术日渐成熟, 基于移动智能终端的应用朝着智能化及便捷性等方面迅猛发展并得以普及, 而基于传统桌面客户端的互联网办公模式(即传统 Web 办公方式)愈来愈无法满足用户的多样性需求。任何新技术只是对传统技术有效的补充。相对于传统的桌面有线办公模式, 移动互联办公模式有明显的不足之处, 如由于移动终端的限制及资源短缺性, 与其相关的接入网络、运营商之间的强相关性等因素, 智能终端服务要求响应时间极短, 无法适应大批量的数据处理业务。因此, 有效利用传统 PC 端 Web 处理数据的优势(如对大批量文档、图形图像的批处理等), 又充分考虑移动智能终端处理数据的便捷化等特性, 进而将应用扩展至当前业界热点——跨平台的混合移动 Web 开发与应用, 是该领域应用发展的主流方向。

本文结合传统 Web 服务器端应用开发及移动 APP 客户端开发技术、混合跨平台移动互联网

开发的特点, 融合三者的优势, 结合当前互联网专业的发展热点, 同时针对当前移动互联网界基于多系统智能应用系统的不兼容问题, 从专业学习者的基础出发, 设计了适合大多数移动终端系统的通用开发框架, 为从事移动互联网学习、Web 开发学习者提供参考。

## 1 融合 APP 客户端的响应式 Web 架构开发模型

结合传统 Web 服务器端应用开发及移动 APP 客户端开发技术、混合跨平台移动互联网等技术的特点, 为充分发挥三者的优势, 作者给出融合 APP 客户端的响应式 Web 三层架构应用开发模型。

如图 1 所示, 基于传统 PC 端的 MVC(Model, View, Controller)三层应用开发模型与基于移动智能终端合开发模型合二为一, 这也是目前既成事实的 Web 应用开发主要模型。

基于传统 PC 端的 MVC 三层应用开发模型中(如图 1 右侧), 为提高应用系统开发效率及充分发挥各成员的技术专长, 同时兼顾后期维护的便捷性, Web 应用系统一般由团队协作开发完

\*收稿日期: 2017-02-25

作者简介: 张烈超(1973-), 男, 湖北武穴人, 武汉交通职业学院教师, 主要从事数据挖掘、移动互联网研究。

成,通常采用框架技术以节减开发成本及缩短开发周期,典型架构思路是基于 MVC 设计的。基于 Web 的应用主要由后台(服务器端)及用户前端两部分组成,随着移动互联网的流行,用户前端又可进一步细分为桌面应用(浏览器)和移动客户端(智能移动终端)。在 B/S(浏览器/服务器)模型的应用中,通过浏览器(客户端)对服务器的访问所需数据及事务处理均由服务端进行处理,将结果返回至客户端;同理,移动客户端与服务器交换数据的方式与上述方法类似。由此可知,在移动互联网应用中的后台(Web 服务器端)依然是 Web 应用的核心及关键。

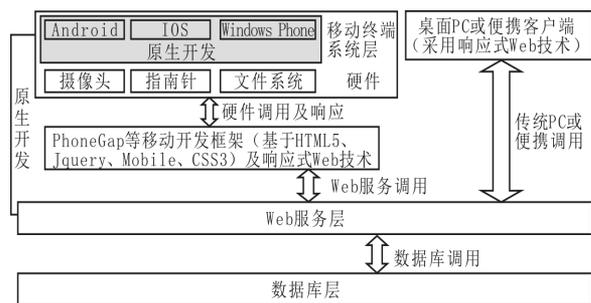


图 1 融合 APP 客户端的响应式 Web 架构开发模型

基于移动端应用开发典型模型如图 1 左侧所示,智能终端的应用主要有以下几种方式:原生(Native)APP 应用、基于 HTML5 的跨平台移动 Web 应用及混合应用(Hybrid APP)开发<sup>[1-2]</sup>。其中:(1)原生 APP 是基于不同操作系统的应用开发,与操作系统联系紧密,但移植性差;(2)基于 HTML5 的移动 Web 应用则采用目前较为成熟的 Web 技术(HTML5、jQuery(Javascript 框架)、CSS3 等)及开发框架,能以较少的成本及开发周期,使开发人员迅速从基于桌面电脑的应用开发转向基于智能移动终端的应用开发,同时该方式支持跨平台的移动智能终端(在 Windows Mobile、Android、IOS 为基础的系统正常运行),缺点是对底层硬件支持较少;(3)基于混合应用(Hybrid APP)开发则充分利用原生应用开发、HTML5 跨平台 Web 开发二者的技术优势。开发者无需掌握原生 APP 开发语言,通过编写一次 Web 代码,借助于相应的框架,便可将其编译成基于不同系统(Android、IOS 等)的智能终端上应用,可大大提高跨平台应用开发的效率。使用该开发模式,Web 开发人员无需额外的学习,以几乎零成本融

入到移动互联网开发学习中来,迅速转型为移动应用开发者。此外,相对于 HTML5 跨平台移动 Web 应用,混合应用开发者可以自行进行系统 API 封装,调用任何期望的系统能力,在很大程度上弥补了目前 HTML5 跨平台调用 API 匮乏的缺陷。

在移动互联网应用开发阶段,可实现传统 Web 开发模式与移动开发模式优势互补:经过授权的 PC 端使用浏览器即可访问应用系统,适用于处理大批量的业务数据;灵活便携的移动智能终端延伸了传统 PC 端的触角,利用移动智能终端便捷的联网能力及可有效利用的散碎时间进行通信等优势,方便用户随时随地联网应用,有效地扩展了网络应用的功能。达到随时随地以协同办公、学习、娱乐等方式应用的目的,该方式采用智能终端移动则更为合理。

开发模型中的应用前端(客户端)之一:桌面 PC 或便携客户端适用于固定办公,对于 Web 应用中的大批量数据处理问题、网络稳定性要求比较高的应用或者耗时较长的交互式应用及系统后台管理,采用桌面 PC 或便携客户端比选用移动智能终端处理起来更为便捷。以桌面 PC 或便携客户端办公为主的 MVC 三层应用架构分工明确,其优点在于把视图(表示层 V)设计与数据持久化(模型层简称 M)进行分离,从而降低耦合性,易于扩展,提高团队开发效率。其中,View 层对应图 1 中的桌面 PC 或便携客户端,用浏览器浏览的网页页面显示数据和提交数据;Controller(控制器)对应图 1 中的 Web 服务层,主要是用作捕获请求并控制请求转发,对于模块事务的具体处理即由此层来完成,需要访问数据时则向与数据层进行交互,将用户需要显示的数据进行处理后以 HTML 网页语言进行包装,并经浏览器访问获取相应视图。Model 层对应图 1 的数据库层,该层主要实现业务逻辑以及数据库的交互;

开发模型中的应用前端(客户端)之二:移动智能终端则更适用于移动应用,在实际选型时,还应根据应用问题的类型进一步细化:

(1)对于交互性频繁、实时性要求较高、开发周期长的移动端应用,动画、游戏等需求较高的应用,选用图 1 的原生(Native)开发模型可望取得较好的效果;具体实现时应注意相关的技术细节,如移动终端实时访问 Web 应用系统时,受时限影

响较明显,表现为:移动智能终端启用 APP 应用程序或者进行远程数据库访问服务时,若响应时间超过 3—5 秒时,系统会提示不可达等相关信息,处理该问题必须引入线程等技术。

(2)对于交互性一般、需要跨平台的移动互联网应用,并尽量减少针对不同平台的二次开发工作。选用图 1 中的混合应用(Hybrid APP)开发模型可望取得更佳的效果。使用该模型,Web 开发人员无需额外的学习,迅速从基于桌面电脑的应用开发转向基于智能移动终端的应用开发。此外,混合应用开发者可以自行进行系统 API 封装,调用任何期望的系统能力,对移动智能终端的摄像头、指南针、文件系统等访问。除此之外,该模型利用如图 1 的 PhoneGap<sup>[3]</sup>(基于 HTML5、CSS3、Jquery)等移动开发技术框架技术,只需编写一次代码,便可将其编译成基于不同系统(Android、IOS 等)的智能终端上的应用,可大大提高跨平台应用开发的效率。混合应用开发模式为开设移动互联网专业的高校及培训机构在人才培养方案的规划及实施提供新的思路。混合跨平台的智能终端开发模式兼容性,既可缩短开发周期,又可节约成本。随着技术的进一步发展,该模式应能大量的应用。

开发应用软件最终目标是面向用户及提高工作效率,融合 APP 客户端的响应式 Web 架构开发模式也应该从用户使用便捷性角度出发。响应式 Web 技术的流行,进一步助推了移动互联网技术的发展。本模型中所有的 Web 应用,均采用响应式 Web 开发技术(如 HTML5)。但响应式 Web 技术并非颠覆原有的 Web 开发技术,只是顺应移动互联网的发展,对先前技术进行有效的扩展,即兼容传统的 Web 技术如 Ps、HTML、CSS、Jquery(Javascript 类库)等。通过响应式 Web 前端开发技术,只需一次性编码,便可自适应于各种终端平台(PC 端或智能终端)。根据智能终端自动调整适配屏幕尺寸,大大减少开发者的工作。

本模型中,除 PC 端之外的移动智能终端的应用,不论采用何种开发模型(混合应用(Hybrid APP)开发模式或者基于 HTML5 的移动 Web 开发模式),与后台数据进行交互处理模式均采用与传统 PC 端应用类似的典型 Web MVC 三层架构开发方式,此举同样以减少模块之间的耦合性,提高开发效率为目的。

需要注意的是,无论选用何种开发模式,在开发过程应尽量优化结构,使系统性能达到最优。如在原生(Native)开发模式下可将移动 Web 服务器端 APP 客户端所需的数据库数据转换成目前流行的 Jason 格式,并在数据传输的过程中进行压缩和加密,以确保应用系统的安全性<sup>[4-5]</sup>。

## 2 融合 APP 客户端的响应式 Web 架构的实现

根据图 1 模型架构思路,作者对跨平台的 Web 应用 APP 进行了测试。相关步骤如下:

(1)采用响应式 Web 技术(基于 Kendo 网页框架),融合 HTML5、CSS3、Jquery Mobile 技术实现了客户前端界面,并使相应 Web 页面适用于各终端页面。关键代码如下所示:

```
<meta name="renderer" content="webkit"
>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="description" content="">
<meta name="keywords" content="">
<meta name="viewport"
content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
<!-- No Baidu Siteapp-->
<meta http-equiv="Cache-Control" content="no-siteapp"/>
<link rel="icon" type="image/png" href="assets/i/favicon.png">
<!-- Add to homescreen for Chrome on Android -->
<meta name="mobile-web-app-capable" content="yes">
<link rel="icon" sizes="192x192" href="assets/i/app-icon72x72@2x.png">
<!-- Add to homescreen for Safari on iOS -->
<meta name="apple-mobile-web-app-capable" content="yes">
<meta name="apple-mobile-web-app-status-bar-style" content="black">
<meta name="apple-mobile-web-app-title" content="Amaze UI"/>
<link rel="apple-touch-icon-precomposed" href="assets/i/app-icon72x72@2x.png">
```

```

<! --Tile icon for Win8 (144x144 + tile
color) -->
< meta name = " msapplication-TileImage "
content = " assets/i/ app-icon72x72@2x.png ">
< meta name = " msapplication-TileColor " con-
tent = " # 0e90d2 ">
< link href = " styles/ kendo. common. min. css "
rel = " stylesheet ">
< link href = " styles/ kendo. rtl. min. css " rel
= " stylesheet ">
< link href = " styles/ kendo. default. min. css "
rel = " stylesheet ">
< link href = " styles/ kendo. dataviz. min. css "
rel = " stylesheet ">
< link href = " styles/ kendo. dataviz. default.
min. css " rel = " stylesheet ">
< link href = " styles/ kendo. mobile. all. min.
css " rel = " stylesheet " />
< link rel = " stylesheet " href = " assets/ css/
amazeui. min. css ">
< link rel = " stylesheet " href = " assets/ css/
app. css ">
< link href = " css/ font- awesome. min. css " rel
= " stylesheet " />
<! --[if IE]>-->
< link href = " css/ font- awesome- ie7. min. css "
rel = " stylesheet " />

```

(2) 在 Eclipse for android 创建相关项目, 按照跨平台移动项目的部署方法, 导入相关文件(包括前面已完成的 Web 页面)至对应的目录, 并编写 MainActivity.java 项目文件: 关键代码如下:

```

private void initView() {
    WebView = (WebView) findViewById(R.id.
WebView1);
    String url = "file:///android_asset/index.ht-
ml";
    WebSettings webSettings = WebView.getSet-
tings();
    // 启用 javascript
    WebSettings.setJavaScriptEnabled(true);
    // 从 assets 目录下面加载 html
    WebView.loadUrl(url);
    // Web 调用 Android 接口名称

```

```

WebView.addJavascriptInterface(this, "wst");
// 覆盖 WebView 默认使用第三方或系统默认
浏览器打开网页的行为, 使网页用 WebView 打开
WebView.setWebViewClient(new WebView-
Client() {
    @Override
    public boolean shouldOverrideUrlLoading
(WebView view, String url) {
        view.loadUrl(url);
        return true;
    }
});

```

(3) 将项目文件发布至 Android 平台。项目运行正常。相关页面在不同平台下显示效果如图 2 所示。

本周课程表	
星期一	
1-2节	大学英语(二), 1-16周A3-320
3-4节	
5-6节	
7-8节	
9-10节	
星期二	
1-2节	数据结构, 1-8周, A7-320
3-4节	
5-6节	体育, 12-16周, 体育场
7-8节	大学英语(二), 1-16周, A3-320
9-10节	
星期三	

图 2(a) PC 系统中页面显示效果

本周课程表	
星期一	
1-2节	大学英语(二), 1-16周A3-320
3-4节	
5-6节	
7-8节	
9-10节	
星期二	
1-2节	数据结构, 1-8周, A7-320
3-4节	
5-6节	体育, 12-16周, 体育场
7-8节	大学英语(二), 1-16周, A3-320
9-10节	

图 2(b) Android 平台下页面效果

使用其他跨平台移动 Web 框架技术也能取得同样的效果。此处不再赘述。后期将在此基础上对基于跨平台的 Jason 格式数据交换相关技术作进一步的探索。

### 3 结语

随着移动互联网的发展及智能终端的普及,基于移动互联网的应用需求日益强烈。若能借助传统 Web 应用开发的优势,并融合基于 HTML5 的移动 Web 应用开发框架及混合跨平台开发框架技术,并在移动开发过程中选择合适的响应式 Web 技术,相信这种方法将会获得事半功倍的效果。

(上接第 31 页)

尽管《慈善法》已于 2016 年 9 月 1 日开始实施,但是《慈善法》仍存在疏漏,它显然没有经得住罗尔事件的考验。微公益依托于互联网,而互联网瞬息万变,很多新的情况层出不穷,制定规章总是存在一定的滞后性,这就导致很多问题无法在其初次出现的时候得到有效解决。但是问题一旦出现,相应的法律法规就应进一步明确和细化,如对“骗捐”界限进一步明晰,成立专门的组织对被捐助人进行一个基本情况筛查等。同时,相应的法律法规也需要对责任与义务进一步的明确:一方面,强调慈善管理者、慈善捐赠者、慈善受益者、慈善流程的权利与义务;另一方面,设立专门的法律监管机构对施与者的捐赠给予合法保护,避免网络虚假慈善信息的欺骗,避免部分微公益机构进行欺诈活动,避免部分人挪用捐赠款等不法行为,保障公民正当利益,提高微公益的公信力和透明度,促进微公益在良好的氛围下健康发展<sup>[1]</sup>。

(三)加强网络建设,提升网民媒介素养,确保网络金融账户安全

确保网络金融账户的安全应从两个方面入手,一方面要加强网络建设,提升网络金融账户的安全,网络安全建设是微公益发展的基石,要从网络技术上确保个人网络金融账户安全、个人隐私信息不会被泄露;另一方面,应加强网民媒介

### 参考文献:

- [1]孟凡宁,丛中昌,黄志兴.移动互联网应用跨平台开发研究[J].移动通信,2013(13):60-63.
- [2]杨勇,邝宇峰,魏骞.移动互联网终端应用开发技术[J].中兴通信技术,2013(19):19-23.
- [3]司徒有波.基于 PhoneGap 的跨平台移动电子书店的研究与实现[D].江苏:南京邮电大学,2014.
- [4]李张永,陈和平,顾进广.跨平台移动 Web 开发框架与数据交互方法[J].计算机工程与计,2014(35):1828-1831.
- [5]权重明,彭昕昉.利用 JSON 实现 Android 高效、安全访问远程数据库的一种方式[J].韶关学院学报(自然科学版),2011,32(12):16-20.

素养,培养网民正确辨识虚假信息的能力,让网民不被有心之人利用,让善良不再遇上谎言,通过合法合规的网络募捐平台发起募捐或进行募捐。

由于微公益还处于起步阶段,目前仍存在许多的不完善之处,面临着公信力缺失、监管机制不完善、网络金融安全存隐患等诸多困境。相关部门以及组织、平台应通力合作,及时、透明反馈微公益的捐助情况;完善细化相应的法律法规,加强监管力度;加强网络建设,提升网民媒介素养,保障网络金融账户安全。此外,应正确引导网民对微公益的认识,而非一味夸赞其优点,更不应弱化甚至掩盖微公益存在的不安全因素。唯有如此,才能使得微公益真正发挥其独特价值,为社会传播正能量。

### 参考文献:

- [1]高燕,王永川,焦健.微公益凸显的伦理问题及对策探析[J].重庆邮电大学学报(社会科学版),2015,3(2):95-99.
- [2]张银锋,侯佳伟.中国微公益发展现状及其趋势分析[J].中国青年研究,2014,10(1):41-47.
- [3]刘绩宏.利他网络与社交网络的拟合——关于微公益信息传播效果的改进[J].新闻界,2011,8(1):85-91.
- [4]宋辰婷.微公益 2.0 中的信息权力建构——以天津爆炸事件为例[J].福建论坛(人文社会科学版),2015,12(1):81-87.

