

基于 NodeJS + Express 框架的轻应用定制平台的设计与实现

王伶俐 张传国

(万达信息股份有限公司研究发展中心 上海 201112)

摘要 在“互联网+”浪潮的推动下,需要通过技术手段来实现企业级的移动应用。如何工作高效化、响应及时化、成本最小化、收益最大化,是目前面临的难题。在分析轻应用基本概念的基础上,总结了轻应用突出于原生应用的特点。同时,利用 NodeJS+Express 框架,搭建了快速定制及发布轻应用的平台。基于此平台,能降低开发者开发应用的难度,提高应用开发效率,减少研发成本,同时缩短应用发布周期。但应用模块库中模块数量及类型不足,页面编辑智能化也需后续进一步完善。

关键词 NodeJS, Express, 轻应用, 应用定制, 应用发布

中图分类号 TP311 **文献标识码** A

Design and Implementation of Light Application Customizations Platform Based on NodeJS and Express Framework

WANG Ling-li ZHANG Chuan-guo

(R&D Center, Wonders Information Co., LTD., Shanghai 201112, China)

Abstract In the “Internet+” wave, we need to achieve enterprise-class mobile applications through technical means. The current challenges are how to work efficiently, response timely, minimize the cost and max the revenue. For those questions, this article analyzed the basic concept of light application and then summarized the characteristics highlighting in native applications. At the same time, we put up the platform for customizing and releasing the light application based on the NodeJS and Express framework. Using the platform, we can reduce the difficulty to develop application, cut the research cost, shorten the development cycle and improve the efficiency of developing the application. In future work, the lack of modules in the module library and the drag-and-drop editing of the page need to be perfect.

Keywords NodeJS, Express, Light application, Application customizations, Application Customization

1 引言

“互联网+”行动计划已经成为中国经济的加速器,从根本上改变着个人生活、企业市场、政府管理的各个方面。特别是移动互联网化,在为我们带来更加广阔的发展空间的同时,也带来了挑战。通过技术手段来实现企业级的移动应用,如何工作高效化、响应及时化、成本最小化、收益最大化,是目前面临的难题。

2013 年,轻应用这一概念被百度提出后,各平台相继加入轻应用的阵营中,轻应用开放平台瞬间进入了繁荣期^[1-2]。移动互联网领域关于轻应用有如下观点:1)轻应用是一种无需下载、即搜即用的全功能 APP,既有媲美甚至超越 Native APP 的用户体验,又具备 Web APP 的可被检索与智能分发的特性,将有效解决优质应用和服务与移动用户需求对接的问题^[3];2)基于超级 APP 之上、围绕手机特性开发、能更方便地曝光和被调用的服务都是轻应用,包括但不限于 Web APP^[4];3)Light APP 是介于 Native App 和 Web APP 的中间态服务产品,其既比 Native App 更轻,又比 Web App 有着更加良好的使用体验^[5]。

轻应用模式更易于开发者开发和用户使用,具体表现在:

1)以超级 APP 为基础,如百度、微信、支付宝等;2)无需下载安装,直接运行于超级 APP 之上,如微信轻应用直接在微信

APP 中添加公众账号后使用;3)轻应用在呈现和展示上具有 Web APP 属性,由超级 APP 结合浏览器技术呈现和展示,绕开浏览器入口,同时受到超级 APP 的限制;4)超级 APP 通过自身的开放平台为轻应用开发提供技术支撑,同时,超级 APP 本身为轻应用提供分发渠道和服务人口。对开发者而言,轻应用的开发和维护更为简单,且没有跨平台的成本。

2 相关技术介绍

2.1 NodeJS 介绍

NodeJS^[6-7]是 Ryan Dahl 于 2009 年发起的开源项目,是一个基于 Chrome V8 引擎,能够快速构建网络服务与应用的 JavaScript 执行平台。NodeJS 对 V8 引擎做了二次封装^[8],针对服务环境重新编写了后端的 API,并优化了一部分代码,构成了一个高效的 JavaScript 运行环境。它同时采用了模块化设计,开发者可以将程序分解成不同的模块,然后通过导入模块的方式实现程序功能的复用,从而大大减少了开发者的工作量。NPM 是 Node 的包管理器,管理着数万基于 Node 平台的第三方开发库,NPM 允许开发者自由上传自己编写的程序库。

NodeJS 作为服务器端 JavaScript 的运行平台,弱类型、基于作用域和原型链是其本身的特征,重点在于将 Web 前端

中一些思想(如事件机制等)迁移到了服务端的环境中。

2.2 Express 框架

Express 是一个简洁而灵活的 NodeJS Web 应用框架,提供了一系列强大的特性来帮助开发者创建各种 Web 应用^[9]。其强大的特性表现为:快速进行开发,拥有灵活的扩展机制,使用简单方便,路由、多模块支持等^[10]。

1)快速开发:不需要手写很多的代码,只需要一行命令,就可以生成 Express 框架的基础模板。

2)灵活的扩展机制:Express 框架可以通过其扩展机制,方便地加入其他功能。

3)使用简单:Express 的 API 都非常的直观、简单,还有详细的 API 文档以供查看。

3 轻应用平台的需求分析

轻应用定制平台主要包含首页定制、应用定制、应用发布,其具体的功能需求如下所述。

3.1 首页定制

首页作为应用内容的汇总和索引,其主要作用体现在让软件内容更加明晰,有自己的个性;指引使用者操作软件,易上手;提高软件的认可度和市场竞争力。

软件首页布局与否合理、是否符合大众的使用习惯也直接影响着使用者对此软件的喜爱程度。如果软件的首页布局不合理,不符合大众习惯,即使有非常强大的功能,那也发挥不出应用本身的作用。

首页定制提供了一系列模版供用户选择,用户可以依据软件的内容、用户的定位选择合适的软件首页。首页模板包括常见的九宫格、侧滑栏、底部 Tabbar 以及不同的主题风格。

3.2 应用定制

轻应用能提供立体高效的分发体系,并大幅降低开发成本,对于开发者而言,可以一次开发多终端适配解决开发过程的重复性问题,不用修改后端程序代码就可以同时用于 Web 界面、手机、平板等多种客户端;对于最终用户而言,能继承原生模式的 UI 表现力让最终用户得到了良好的用户体验,对于企事业单位而言,省事省力地快速完成应用的分发,安全、易维护。

应用定制作为轻应用定制平台的核心功能,需完成从独立模块到应用的组合和个性化配置,包含模块的增加(模块最大不超过 10 个)、删除、修改(修改模块图标和模块名称)、应用模块基本风格配置、模块源码编辑、模块中自定义服务对接。同时,提供应用源码下载,用户依据项目的需求按照开发规范个性化地开发轻应用。

3.3 应用发布

应用发布的核心能力是渠道能力。作为一个渠道,其必须将应用从开发者手中实时传递至用户手中,以高质量的应用来聚集用户,同时以成规模的用户吸引开发者。微信公众号、支付宝生活号作为当今渠道能力较强的聚合平台,成为了应用推广的快捷途径。其具体功能体现在可以给关注者推送文字、语音、视频、图片和图文消息;向公众平台提问,提问的内容可以通过文字发送或者语音发送,平台会自动识别内容,依据设置好的关键字推送给相应用户。

轻应用借助微信公众号、支付宝生活号等社交渠道,方便在企业之间、用户之间进行分享,无需下载,所点即所得。发布的功能需包含绑定公众号(生活号)、绑定菜单、绑定关键字;绑定成功后,用户可以关注公众号(生活号),通过菜单、关

键字回复访问绑定的轻应用。

4 平台系统架构

轻应用定制平台架构如图 1 所示,其中前台逻辑和页面效果主要采用 EJS+CSS+JavaScript 来实现。EJS 完成动态页面,CSS 完成页面的排版,JavaScript +Jquery 主要用于对前端数据的验证,采用 Ajax+AngularJS 技术来保证用户交互的及时响应。平台在 UI 设计时,参考现在较为流行的扁平化风格,使用了一种简单直观的布局方式。定制平台中的首页定制、应用定制、应用绑定界面均以 EJS 编写,用 Ajax+AngularJS 技术进行用户的交互响应。

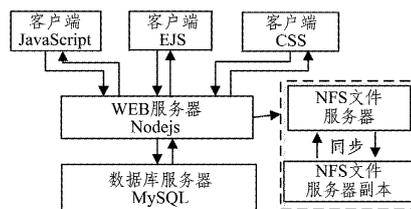


图 1 系统架构设计图

轻应用平台的后台实现采用 NodeJS 框架。NodeJS 是一个新的 Web Server 解决方案,它已经成为支持 JavaScript 在服务器端运行的有效平台。NodeJS 在不新增额外线程的情况下仍然能够对任务进行并行处理,NodeJS 使用 Module 划分不同的功能,并采用事件驱动机制、异步编程风格,使得 NodeJS 具有相当高的性能。应用模块的增删改、首页定制、应用绑定发布接口均以 NodeJS 实现。

数据库采用 MySQL 来实现。MySQL 数据库具有可扩展性和可移植性,同时优越的稳定性、灵活性和强大的数据保护功能用以保护应用的数据安全。应用信息、应用模块信息、模块库信息、应用绑定信息等都存在 MySQL 数据库中。

以应用模块添加为例,前台通过 HTTP 请求,发送添加模块的请求,后台通过 NodeJS 处理前台发送过来的 HTTP 请求,将模块的 id 和副本路径、相关应用信息写入应用信息表中,并将模块副本解压至 NFS 服务器中对应的应用副本中,写入成功后将处理结果返回给前端,新增加的模块以列表形式展示。

对于备份数据库问题的解决,使用传统的数据库集群方式进行及时备份。

网络文件系统(Network File System,NFS)是 FreeBSD 支持的文件系统中的一种,它允许网络中的计算机之间通过 TCP/IP 网络共享资源。NFS 服务器用于存放应用的副本文件及其备份。

5 轻应用平台的实现

5.1 首页定制

首页定制前需要开发相应样式的首页模板,作为首页模板库数据源,包含但不限于九宫格、侧滑栏、底部 Tabbar、不规则矩形,目前开发了 6 个首页模板,效果如图 2 所示。

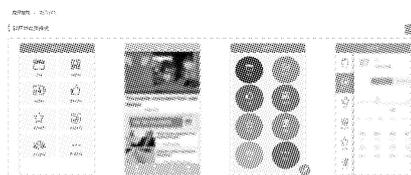


图 2 首页定制

系统的首页模板库使用 MySQL 进行存储,其数据结构定义为:

```

{
  "home_library_id": "int",
  "image": "varchar",
  "name": "varchar",
  "description": "longtext",
  "zip_uri": "varchar",
  "created_at": "int",
  "updated_at": "int"
}

```

其中 home_library_id 为首页模板的唯一标识。image 和 name 分别为模板的示例图片和模板名称。zip_uri 为模板压缩包在工程目录中的存放路径,created_at 和 updated_at 为模板的创建时间和更新时间。当用户在首页定制界面选择某首页后(见图 2),点击保存,采用 post 的方式将首页 id 传输到服务器上,并与 app 表(应用基本信息表)中的 home_type 进行对比,若记录一致,则不做修改;若不一致,则更新 app 表中的 home_type 字段,并将对应的首页压缩包解压至文件服务器的应用副本中,同时返回更新成功信息。当用户预览应用时,从 app 表中查找应用副本的路径(即 manage_uri 字段),然后读取应用首页副本,进入首页界面。

5.2 应用定制

应用定制功能的实现依赖于系统已经存在的模块库。定制前需要独立开发各种通用的模块,如个人健康档案中的就诊、住院、医学影像、用药等,市政服务中的办事机构公告、失物招领、网上办事等,目前开发了 14 个独立模块,各模块可以独立运行成一个应用,也可以组合成一个应用。

应用定制是整个平台的核心部分,主要包含应用模块添加、应用模块删除、应用模块配置与源码编辑,定制流程如图 3 所示。

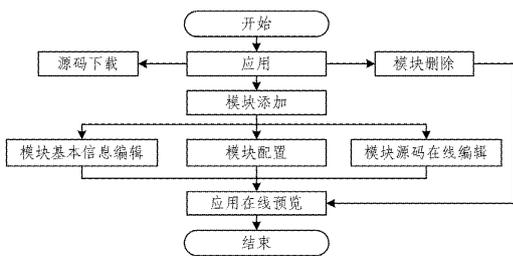


图 3 应用模块的定制及编辑流程

应用信息存储在 MySQL 数据库中,其中 app 表(应用基本信息表)存储应用的基本信息,其数据结构定义为:

```

{
  "app_id": "int",
  "icon": "varchar",
  "name": "varchar",
  "description": "longtext",
  "home_type": "int",
  "manage_uri": "int",
  "updated_at": "int",
  "is_active": "tinyint"
}

```

其中 app_id 为应用的唯一标识,icon 和 name 分别为应用的图标和名称,home_type 为应用首页的类型,manage_uri 为应用副本在文件服务器的存放路径,created_at 和 updated_at

分别为应用的创建时间和更新时间。

app_module 表(应用模块表)存储应用的模块信息,其数据接口定义为:

```

{
  "app_id": "int",
  "icon": "varchar",
  "name": "varchar",
  "description": "longtext",
  "module_library_id": "int",
  "module_id": "int",
  "manage_uri": "varchar",
  "created_at": "int",
  "updated_at": "int"
}

```

其中 app_id 为应用的唯一标识,module_id 为应用模块的唯一标识,两者用来查找应用模块,icon 和 name 分别为应用模块的图标和名称,module_library_id 为应用模块在模块库中的 id,manage_uri 为应用模块副本在文件服务器的存放路径,created_at 和 updated_at 分别为应用模块的创建时间和更新时间。

应用定制依赖这两张表进行,其界面如图 4 所示。



图 4 应用模块定制界面

(1) 模块添加

平台限制了每个应用添加的模块数不得大于 10 个。开发者从系统模块库中选择需要添加的模块,如图 5 所示,平台采用 post 的方式将该模块 id 以及应用 id 传输到服务器上,并写入 app_module 表中,同时将模块解压至对应的应用副本中。



图 5 模块库

(2) 模块删除

开发者删除某一模块时,平台采用 post 的方式将该应用的模块 id 以及应用 id 传输到服务器上与 app_module 表(应用模块表)中的 module_id 和 app_id 进行对比,若记录一致,则删除此条记录,同时依据 manage_uri 的地址将对应的应用模块副本从文件服务器中删除,最后返回删除成功的信息提示。

(3) 模块配置

模块配置包含页面默认字体大小、默认字体颜色、默认背

景颜色、导航条字体大小、导航条字体颜色、导航条背景颜色。模块配置的修改不对数据库进行读写操作,直接对文本服务器中应用模块副本的配置文件进行修改,模块配置的界面如图 6 所示。



图 6 模块配置界面

(4) 模块编辑

模块编辑分为模块基本信息编辑和模块源码编辑。

模块基本信息编辑包含模块名称、模块图标(界面如图 4 所示)。平台将修改后的模块名称、模块图标以及模块 id、应用 id 传输到服务器上,在 app_module 表中依据模块 id 及应用 id 查找相应的应用模块记录,并进行更新操作。

模块源码编辑是指当系统模块库提供的模块不能满足开发者的个性需求时,开发者可以对模块进行编辑,编辑时需要依据开发文档、模块配置文档的规范来以保证编辑后的模块能正常在应用中运行,界面如图 7 所示。



图 7 源码编辑

新增页面时,引入了 Bootstrap-Form 表单插件^[11],该插件实现了可视化组件的拖拽,属性设置、简单可靠、轻松高效。界面如图 8 所示,左边为定制化的表单,右边为表单的各类功能块,如输入框、单复选框、按键等,编辑时,只需要将右边的功能块拖拽至左边的表单框中即可,同时可在 render 项中查看渲染的源码。模块源码编辑不对数据库进行读写操作,直接对文本服务器中应用模块的副本文件进行修改。



图 8 表单在线拖拽

5.3 应用发布

应用发布是指将定制好的应用通过微信公众号、支付宝生活号进行推广,其方式为绑定菜单、绑定关键字以及以图文方式嵌入文本信息中。发布的前提条件是必须要先绑定公众号或者生活号,发布流程如图 9 所示。绑定菜单、绑定关键字主要依据微信公众平台文档中心^[12]和支付宝蚂蚁金服开放平台文档中心^[13]约定的规则进行开发。

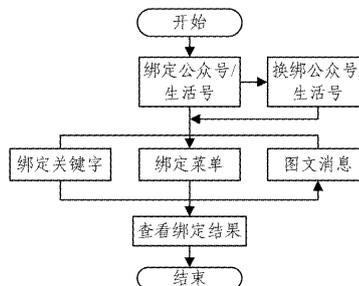


图 9 轻应用通过微信/支付宝发布流程

结束语 本文基于 NodeJS+Express 框架搭建了快速定制轻应用的平台。基于此平台,开发者可以实现应用的快速定制、修改及发布。此平台的技术优势主要表现为降低了原生应用的开发难度,提高了应用开发效率,减少了研发成本,缩短了应用程序的发布周期。其功能特点表现为一次开发多终端适配的跨平台引擎;通过模块集成应用,易扩展,复用性高;省事省力快速完成应用的分发,安全且易维护。但该平台仍有功能待完善,主要表现为应用模块库的模块待丰富,涉及的领域也待补充;源码编辑时,新建页面主要靠代码编辑,对于所见所得的组件拖拽示例较少,页面组件内容待扩展。

参考文献

[1] 白明凤,匡惠华. 高校图书馆移动信息服务中轻应用模式的应用及其借鉴——基于高校图书馆微信公众号的分析[J]. 情报资料工作, 2014(4):78-81.

[2] Keyla. 突然冒出来的轻应用开放平台,都长什么样[Z]. 2013.

[3] 百度百科. 轻应用[Z]. 2014.

[4] 何小鹏. 轻应用,不只是 Web App[Z]. 2013.

[5] Native APP 的长尾困境与 Light APP 的新机遇[Z]. 2013.

[6] 黄扬子. 基于 NodeJS 平台搭建 REST 风格 Web 服务[J]. 无线互联科技, 2015(16):57-59.

[7] 王越. 基于 nodejs 的微博系统的设计与实现[D]. 成都:电子科技大学, 2014.

[8] 曹钟. 结合 Hadoop 大数据处理框架的智能快递信息系统设计与实现[D]. 北京:北京建筑大学, 2016.

[9] 程桂花,沈炜,何松林,等. Node.js 中 Express 框架路由机制的研究[J]. 工业控制计算机, 2016, 29(8):101-102.

[10] 朱建兵. 基于 Node.js 高并发网络应用架构的研究与实现[D]. 北京:北京邮电大学, 2013.

[11] MINIKOMI. Bootstrap-Form-Builder[Z]. 2016.

[12] 微信平台. 微信公众平台开发者文档[Z]. 2016.

[13] 蚂蚁金服开发平台. 蚂蚁金服开放平台文档中心[Z]. 2017.