

题 目：基于html的移动社交app的设计与实现

姓 名：叶长青

学 号：201508090062

学 院：网络工程学院

专 业：网络工程

年级班级：2015级网络工程2班

指导教师：高光

2019年05月01日

**毕业论文作者声明**

本人郑重声明：所呈交的毕业论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。

本人完全了解有关保障、使用毕业论文的规定，同意学校保留并向有关毕业论文管理机构送交论文的复印件和电子版。同意省级优秀毕业论文评选机构将本毕业论文通过影印、缩印、扫描等方式进行保存、摘编或汇编；同意本论文被编入有关数据库进行检索和查阅。

本毕业论文内容不涉及国家机密。

论文题目：基于HTML的移动社交App的设计与实现

作者单位：网络工程学院

作者签名： （学号：201508090062）

2019 年 05月01 日

**目 录**

[摘要： 1](#_Toc8306538)

[1.绪论 1](#_Toc8306539)

[1.1研究意义 1](#_Toc8306540)

[1.2研究目标 2](#_Toc8306541)

[2.前端的发展与现状 2](#_Toc8306542)

[2.1前端的概念 2](#_Toc8306543)

[2.2前端的历史 2](#_Toc8306544)

[2.3前端的现状 2](#_Toc8306545)

[2.4 前端的发展趋势 3](#_Toc8306546)

[3.系统前端设计 4](#_Toc8306547)

[3.1 系统布局分析 4](#_Toc8306548)

[3.2 产品布局和页面设计 4](#_Toc8306549)

[3.3 技术分析 9](#_Toc8306550)

[4.系统前端实现 10](#_Toc8306555)

[4.1 目录结构解析 10](#_Toc8306556)

[4.2 部分代码及功能 10](#_Toc8306556)

[4.3 vue.js特性解析 13](#_Toc8306558)

[4.4 axios的功能解析 15](#_Toc8306563)

[5. 总结 16](#_Toc8306564)

[参考文献 17](#_Toc8306565)

[致谢 18](#_Toc8306566)

**基与HTML的移动社交APP的设计与实现**

摘要：伴随着移动终端技术的不断成熟，4G网络的全面普及，以及刚刚到来的5G网络。手机上网以高性能、低成本、快速便捷的优势成为大部分人们上网的首选。本课题以HTML5新一代web标准为基础，使用了主流的vue框架配合weui框架，开发的一款高质量的移动社交应用，利用html5的开发周期短、成本低、兼容性高的优势，达到跨平台运行效果。真正实现“一次编写，到处运行”。

关键字：HTML5；VUE；移动终端

**Design and Bmplementation of Mobile Social APP Based on HTML**

**Abstract:** With the continuous maturity of mobile terminal technology, the comprehensive popularization of 4G network and the arrival of 5G network. Mobile Internet with its high performance, low cost, fast and convenient advantages become the first choice of most people online. This project is based on the new generation HTML5 web standard and adopts the mainstream vue framework and weui framework to develop high-quality mobile social applications. It takes advantage of the features of short development cycle, low cost and strong compatibility of HTML5 to achieve the effect of cross-platform operation. The real implementation is "write once, run everywhere."

**Key Words:** Html5; Vue; Mobile terminals

# 1.绪论

## 1.1研究意义

随着移动互联网的飞速发展，手机已经成为了人们生活中不可缺少的部分，手机app的开发需求也变的庞大起来，本作品采用的是HTML5应用开发，是利用Web技术进行的App开发， web技术本身需要浏览器的支持才能进行展示和用户交互。主要用到的技术是HTML5、JavaScript、CSS等。H5开发的好处是可以跨平台，编写的代码可以同时在Android、IOS、Windows上进行运行。解决了原生开发开发成本高、开发周期长的问题，同时H5开发的代码维护方便、版本更新快，降低产品成本。

## 1.2研究目标

目前我们使用的许多手机app是采用的原生技术开发，原生应用开发，是利用Android、IOS等移动平台上提供的官方开发语言、类库和开发工具进行App开发。比如android是利用java、Android studio，IOS是利用Objective-C 和Xcode进行开发。原生应用由于是利用的官方提供的语言和工具并且能够直接操控硬件设备，在应用性能上和交互体验上目前属最好的，但是原生应用的可移植性比较差，特别是开发一款原生的App，在不同的而操作系统中，同样的逻辑、界面要写两套代码。这就使得一款产品的开发成本大大增加，提高了互联网创业的门槛。而H5开发跨平台的优势对于开发来说就很友好了，虽然说由于Web技术本身的限制，H5移动应用不能直接访问设备硬件和离线存储[1]，所以在体验和性能上有很大的局限性。但是随着H5技术的不断成熟，未来H5开发逐渐取代原生开发是非常有可能的。

# 2.前端的发展与现状

## 2.1前端的概念

在软件体系结构和编程领域，前端是软件系统中直接和用户交互的部分。对于一个网站来说，前端通常是指网站的前端部分，包括:网站的表现层和结构层[2]。因此，前端技术一般分为:前端设计和前端开发两部分。前端设计可以理解为网站的视觉设计，包括但不限于品牌Logo、用户界面和交互逻辑的设计。前端开发是web页面的代码实现，涉及的技术包括但不限于:HTML、CSS、JavaScript。

## 2.2前端的历史

在二十年前，前端还不存在，作为一个从后端细分出来的工种。网站开发人员主要是使用Flash做动画，Dreamweaver“可视化”编写web页面，以及Firework来剪切图像[3]。所以长期以来，业内人士俗称或自嘲为“切图仔”。随着Web 2.0概念的兴起，用户对交互更丰富、体验更极致的产品的需求越来越强烈，前端在网站开发中扮演着越来越重要的角色。从那时起，前端进入了蓬勃发展的十年，各种新技术和新思想层出不穷。

## 2.3前端的现状

2010年Backbone、Angular.js等前端JavaScript框架相继出现，前端开发人员逐渐有了信心，因为他们也可以借助MVC、MVVM来构建具有可靠范式支撑的SPA(单页应用)。

而2009年出现、2011年后不温不火的Node.js把JavaScript直接从浏览器端带入到服务器端，继而npm社区方兴未艾，进而出现React、Polymer等推动W3C规范或多方互相影响、借鉴、学习的“新概念”前端JavaScript框架，让从业人员意识到：前端领域也可以实现组件化，反向推动行业规范的制定。而与此同时，Grunt、Gulp、Webpack等前端开发自动化、前端打包工具的相继出现，使得前端工程化告别了刀耕火种之状态，一同步入了初级阶段。

早早地，Node-WebKit（现名为NW.js）和Electron已在桌面端攻城掠地多时。早有像Atom编辑器这样由Github社区开发维护并广受好评的成功开源项目。而后Facebook的React Native把矛头和目标直指Android和iOS开发领域，其愿景不可谓不宏大，React Native引入虚拟DOM（Virtual DOM）的概念，借助中间的抽象层，理论上可以横跨所有平台，支持使用HTML、CSS、JavaScript构建跨平台产品和应用，大大提高开发效率，缩短开发周期[4]。期间，不乏像Meteor、MEAN这样的全栈平台和全栈框架大展身手。至此，前端领域在短短的四五年间，可谓变幻莫测、生机勃勃，也不免显得稚气未脱，远未成熟。

## 2.4 前端的发展趋势

步入2018年下半年，ECMAScript标准继续深化，此时思考前端的发展趋势，可大致作出如下判断：

1. W3C标准将继续深化，对HTML、CSS、JavaScript分别提供更具语义、更强大、更丰富的标签、属性、接口等；
2. 开源社区将继续蓬勃发展，只是前端工具将趋于理性和成熟。过时的思想和工具将被淘汰；
3. 使用node实现跨平台的JavaScript将继续流行。未来任何可以用JavaScript实现的产品都将用JavaScript实现[5]。因此，前端将成为大多数开发领域的基本技能之一。换句话说，前端将慢慢成为通用技术;
4. 前后端分离的开发模式将得到更多的重视，前端从业人员的工作职责将进一步扩大，同时入门门槛将进一步提高，只有具备一定后端知识的前端开发人员，才能真正胜任未来前端领域的工作；
5. 前端工程化的理念将越来越被行业接受，前端领域将步入全面工程化阶段。

# 3.系统前端设计

## 3.1 系统布局分析

本作品采用前后端分离的开发模式，主要探究的是前端的构成：前端页面架构、静态页面设计和前端的性能优化，借助前端JavaScript框架Vue.js+移动端前端UI框架WeUI+前端打包工具WebPack+npm插件，完成的一个单页应用（SPA）形式的移动社交app。

具体采用的技术栈为：Vue.js+WeUI+Webpack+Npm

## 3.2 产品布局和页面设计

下面是本作品部分页面展示：



图1 聊天对象

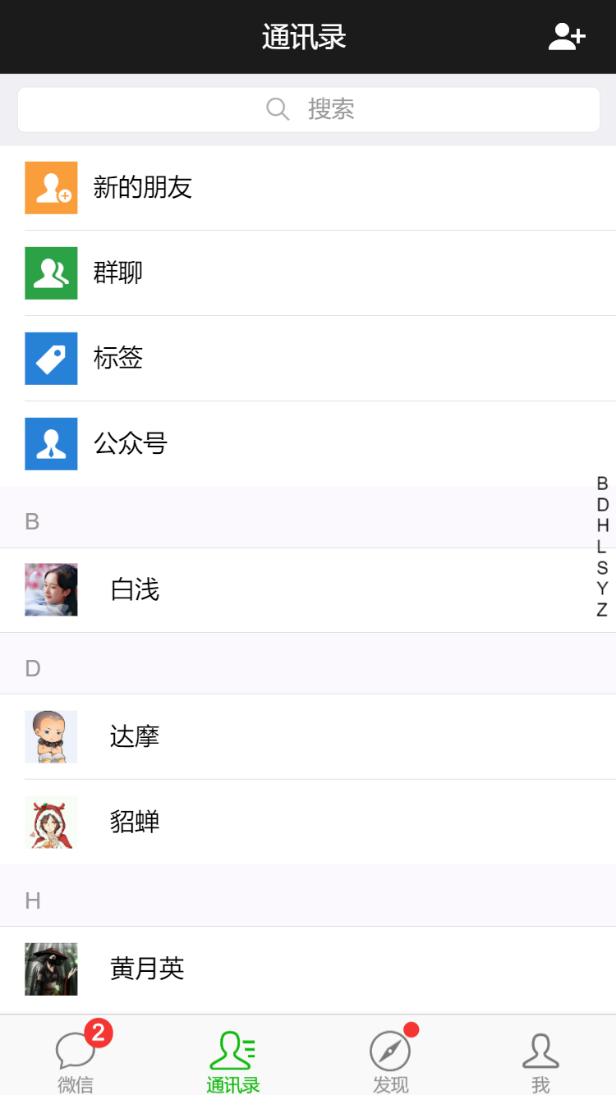


图2 通讯录

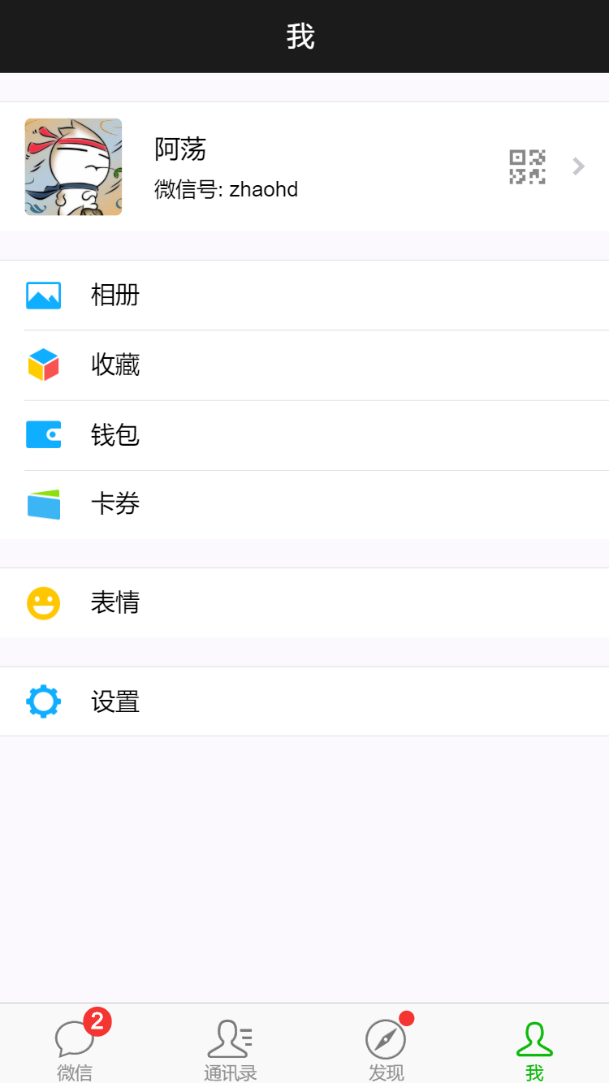


图3 个人信息



图4 群聊信息



图5 朋友圈

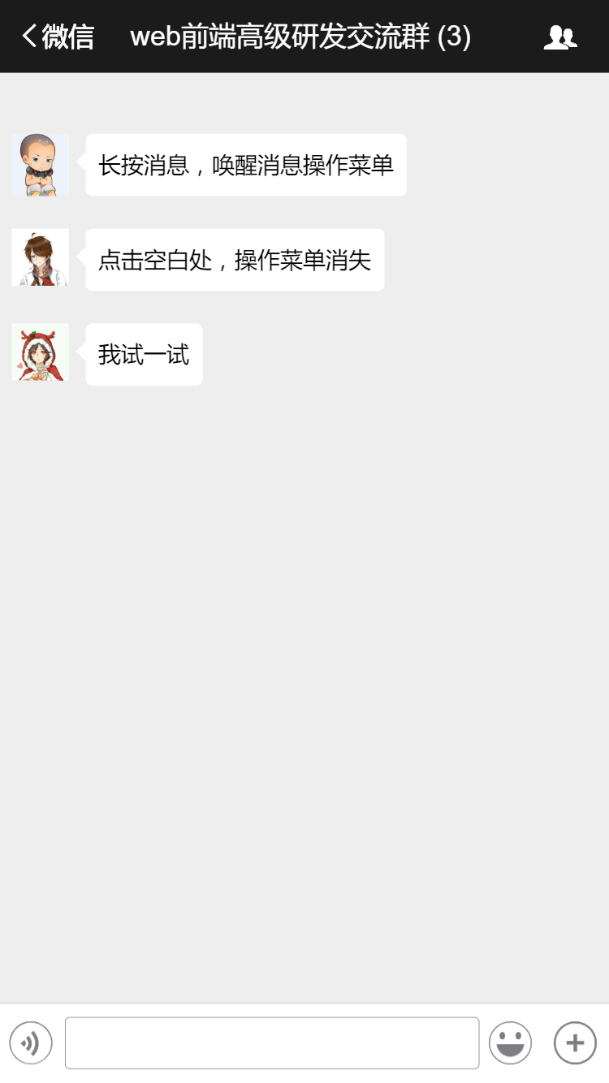


图6 群聊

## 3.3 技术分析

### 3.3.1 Vue.js

Vue.js是基于MVVM范式的轻量级高效前端JavaScript框架，为现代Web接口提供数据驱动的Web组件构建功能。Vue.js提供双向数据绑定、计算属性、组件、自定义指令和过滤器等特性。与此同时，Vue.Js可以围绕生态插件构建大中小型的应用程序。扩展性、代码重用性、可维护性都非常高，是一个难得的优秀的前端开源JavaScript框架。

### 3.3.2 WeUI

WeUI 是由微信官方设计并推出的一套同微信原生视觉体验一致的基础样式库。

### 3.3.3 Webpack

Webpack是一个新兴的前端模块打包工具，它提供了诸如模块依赖关系管理之类的特性。与Webpack插件生态系统一起，Webpack扩展了前端自动化和前端工程的前沿。Webpack现在非常流行，许多优秀的开源JavaScript框架都喜欢将Webpack作为默认的打包和构建工具。

### 3.3.4 Npm

npm 是 JavaScript 世界的包管理工具,并且是 Node.js 平台的默认包管理工具。 npm 为开发人员提供了安装、共享、分发代码,管理项目依赖关系的功能。对于前端开发人员来说，npm提供及极大地便利，同时也是展示自己能力的平台。

# 4.系统前端实现

## 4.1 目录结构解析

前端目录结构如下：

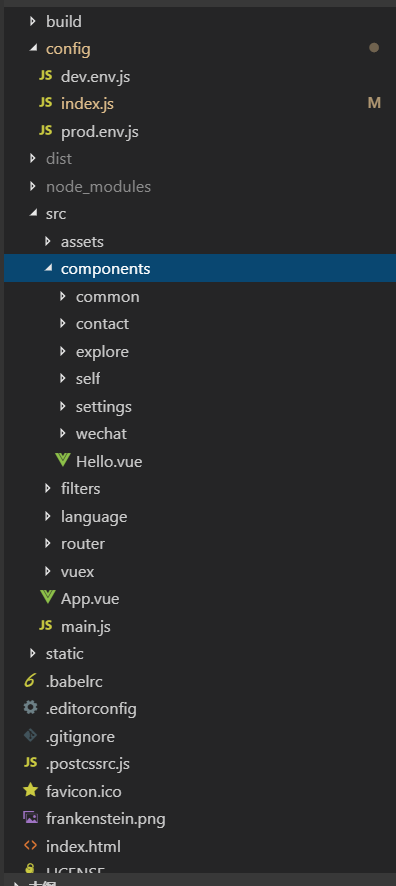


图7 前端项目工程目录

* build 文件夹：用来存放 webpack 相关配置信息和脚本。开发中可以在webpack.base.conf.js中配置less、sass等css预编译库，以及配置一些引用的第三方类库和脚本文件。
* config 文件夹：主要用来存放配置文件，比如一些常见的开发环境的端口号、是否开启热加载、生产环境的静态资源相对路径以及打包时生成的静态资源路径和名称等。
* dist 文件夹：用来存放npm run build 命令打包生成的静态资源文件，一般来说，打包完成后会默认生成。
* node\_modules：用来存放使用npm命令下载的开发环境和生产环境的依赖包。
* src: 用来存放项目源码及需要引用的资源文件。
* src下assets：用来存放项目中需要用到的资源文件，例如创建css、js、images文件夹等将对应的文件分别存放，易于查看。
* src下componets：存放vue开发中一些公共组件，以及存在vue页面组件的文件夹。
* src下router：vue-router vue路由的配置文件。
* src下vuex：存放 vuex 为vue专门开发的状态管理器。
* src下app.vue：使用标签<route-view></router-view>渲染整个工程的.vue组件。
* src下main.js：vue-cli工程的入口文件。
* index.html：设置项目的一些meta头信息和提供<div id="app"></div>用于挂载 vue 节点。

· package.json：用于 node\_modules资源部 和 启动、打包项目的 npm 命令管理。

## 4.2 部分代码及功能解析

下面代码是app.vue中的一部分：

<div id="app">

<welcome></welcome>

<div class="outter"

:class="{'hideLeft':$route.path.split('/').length>2}">

<!--通用头部-->

<header class="app-header"

:class="{'header-hide':!$store.state.headerStatus}">

<wx-header :pageName="pageName"></wx-header>

</header>

<!--搜索框 只在“微信”和“通讯录”页面下显示-->

<search v-show="$route.path.indexOf('explore')

===-1&&$route.path.indexOf('self')===-1"></search>

<!--四个门面页 “微信” “通讯录” “发现” “我”-->

<section class="app-content">

<keep-alive>

<router-view name="default" ></router-view>

</keep-alive>

</section>

<!--底部导航 路由 -->

<footer class="app-footer">

<wx-nav></wx-nav>

</footer>

</div>

<!--其他店内页集合 有过渡效果-->

<transition name="custom-classes-transition"

:enter-active-class="enterAnimate"

:leave-active-class="leaveAnimate">

<router-view name="subPage" class="sub-page"></router-view>

</transition>

</div>

app.vue页面是所有页面的入口，是整个页面的核心部分。在这里我将页面分为四部分：通用头部、主体页面、底部导航、其它页面。通用头部控制页面的头部的显示与隐藏以及动态载入页面的名称，主体部分使用<keep-alive>包裹路由进行缓存，主体部分页面属于四个导航部分的页面及子页面。底部导航是简单的一个nav组件（微信、通讯录、发现、我）。其它页面是不属于主体部分页面的页面。

## 4.3 vue.js特性解析

本作品重点使用Vue.js框架来实现数据驱动的组件。下面重点讲解Vue.js的基本用法，也就是Vue.js框架的核心：列表渲染、条件渲染、双向数据绑定和vue-route。

### 4.3.1 vue.js列表渲染

Vue.js的列表渲染指的是把从服务器上面请求回来的JSON数据或者页面中的data数据渲染成用户可见的列表：

<ul class="wechat-list">

<!--props传递消息对象 baseMsgObj -->

<msg-item

v-for="baseMsgObj in $store.state.msgList.baseMsg"

:item="baseMsgObj"

class="list-row line-bottom" :key="baseMsgObj.mid">

</msg-item>

</ul>

这段代码的用处是：渲染微信页的聊天对象。使用vue.js提供的v-for方法，将vuex中缓存的聊天对象的以一个个msg-item组件的形式渲染到界面上

### 4.3.2 vue.js条件渲染

<div class="operate-box">

<div class="operate-unread"

v-if="read" v-on:click="toggleMsgRead">标为未读</div>

<div class="operate-read"

v-else v-on:click="toggleMsgRead">标为已读</div>

<div class="operate-del" v-on:click="deleteMsgEvent">删除</div>

</div>

上面这段代码是以这种常见的消息列表左右滑动显示选项的功能，主要是通过v-if和v-else实现不同状态下显示不同的内容，标志为read，当read为真时，消息为已读，这时未读标志就会隐藏

### 4.3.3 vue.js数据双向绑定

<div id="app">

<div>{{message}}</div>

<input type="text" v-model="message">

</div>

上述代码在页面上渲染出了一个文本区域和一个输入框，实现了输入框内的值实时显示在文本区域中。值得注意的是，data中message的值也会随着输入的改变而改变，这种特性称为“双向数据绑定”。对比JavaScript中复杂的DOM操作，Vue.js的实现方式显得更简洁明了。

### 4.3.4 vue.js的路由

作为vue.js的官方路由插件，vue-router和vue.js深度集成，适合构建单页面应用程序。Vue的单页应用程序基于路由和组件，用来设置访问路径，并将路径和组件映射起来。传统的页面应用，是使用一些超链接来实现页面切换和跳转。在vue-router单页应用程序中，则是路径之间的切换，即组件之间的切换。路由模块的本质是建立url与页面之间的映射关系：

import Vue from "vue"

import Router from "vue-router"

const weixin = resolve => require(["../components/wechat/wechat.vue"], resolve) // 微信

const friendChat = resolve =>

require(["../components/wechat/dialogue.vue"], resolve) // 好友聊天页面

const routes = [

{

path: "/",

name: "微信",

component: weixin

},

{

path: "/wechat/dialogue",

name: "",

components: {

default: weixin,

subPage: friendChat

}

},

]

以上代码是本作品中部分路由的配置信息，根据这从而实现了页面组件的跳转、页面间传参以及页面与页面间切换动画的配置等功能。

## 4.4 axios的功能解析

axios 是一个基于Promise 用于浏览器和 nodejs 的 HTTP 客户端，主要是用于向后台发起请求。它支持浏览器、node.js和promise，能提供对请求和响应进行拦截、转换请求和响应数据、自动转换JSON数据等功能：

export function fetch (options) {

return new Promise((resolve, reject) => {

const instance = axios.create({

// instance创建一个axios实例，在这里可以自定义配置自己需要的信息

// 所有的请求都会带上这些配置，比如请求头信息、全局都要用的身份信息等，

baseURL: url,

headers: { 'Content-Type': 'application/x-www-form-urlencoded' },

transformRequest: [

function (data, config) {

if (!config['Content-Type']) return Qs.stringify(data)

switch (config['Content-Type'].toLowerCase()) {

// 返回得到一个对象（可以为json或者是一个数组）

case 'application/json;charset=utf-8': {

return JSON.stringify(data)

}

// 如果发送的是formdata属性的配置时，直接将当前的data返回,不做任何处理（默认当前请求头的内容为formdata格式）

case 'multipart/form-data;charset=utf-8': {

return data

}

// 默认返回QS序列化的内容，防止内容没用进行处理

default: {

return Qs.stringify(data)

}

}

}

],

params: {

sessionToken: state.sessionToken

}

})

一下代码是部分axios的配置信息，从中可以看出我们可以通过解析请求头对不同的数据类型进行不同的数据序列化，通过添加请求拦截器和响应拦截器对需要处理的请求个响应进行统一处理。比如限制重复请求的发送、减少向服务器请求数量从而减少服务器压力，根据不同情况对请求设置超时时间等等，达到性能优化的目的。

# 5. 总结

经过一个多月的努力我的毕业设计和毕业论文终于完成了。通过这次移动端app的设计，让我对自己的目前所掌握的理论和实践进行了一次全面的梳理。让理论与实践更加密切的结合在一起。虽然毕业设计的内容不多，但是在这个过程中让我经历了一个项目从0到1的历程，从一开始的无从下手到现在的胸有成竹，这给了我很大的信心，让我了解专业知识的同时也对前端目前的发展前景充满信心。

对于毕业设计的实现上，我以现有的掌握的知识为基础，在实现过程中，让我发现掌握的大量知识都是模糊的，只是知其然而不知其所以然。例如，本设计在页面的权限上面的处理一开始让我不知道该怎么处理，最后发现使用路由元信息设置标记，然后监听路由去获取标记达到控制权限的目的。这些发现对于我的帮助是非常大的，它让我能及时的发现自己的不足并改正。同时我也大胆的采用了一些新的技术，努力了解行业前沿技术，了解学习新型框架、插件，提高自己的开发效率和质量。提高是有限的也是全面的，正是这一次设计让我积累了无数的实际经验。今后我会更加关注行业新技术，掌握这些先进的知识，在前端的道路上且行且珍惜。

# 参考文献

[1] 潘志宏，罗伟斌，柳青.基于HTML5跨平台移动应用的研究与实践[J].电脑知识与技术，2013.

[2] 俞华锋.基于HTML5的网页设计与实现[J].科技信息，2012.

[3] 唐灿.下一代Web界面前端技术综述[J].重庆工商大学学报：自然科学版，2009（8）.

[4] 马新强，孙兆，袁哲.Web标准与HTML5的核心技术研究[J].重庆文理学院学报：自然科学版，2010.

[5] 刘春华.基于HTML5的移动互联网应用发展趋势[J].移动通信，2013.

# 致谢

大学生活一晃而过，回首走过的岁月，心中倍感充实，当我写完这篇毕业论文的时候，有一种如释重负的感觉，感慨良多。

首先诚挚的感谢我的论文指导老师高光老师。他在忙碌的教学工作中挤出时间来审查、修改我的论文。对该论文从选题，构思到最后定稿的各个环节给予细心指引与教导，使我得以最终完成毕业论文设计。

从开始写作至论文最终定稿，总共花费了我一个月以来所有的业余时间，虽说在繁忙的工作之余要完成这样一篇论文的确不是一件很轻松的事情，但我内心深处却满含深深的感激之情。感谢我的论文指导老师高光老师，感谢教过我的所有老师们。你们严谨细致、一丝不苟的作风一直是我工作、学习中的榜样，是你们让我能够静静地坐下来，在知识的海洋里吸取更多的营养，从而能够为自己进一步的加油充电。

谨以此致谢最后，我要向百忙之中抽时间对本文进行审阅的各位老师表示衷心的感谢。