

东 华 大 学

毕业设计（论文）开题报告

课 题 名 称 : 基于 React 技术栈的个人博客系统

学 院 : 计算机科学与技术学院

专 业 : 卓越软件 1402

姓 名 : 毛 泉

学 号 : 141320222

指 导 教 师 : 孙 莉

2018 年 3 月 3 日

基于 React 技术栈的个人博客系统

1、背景介绍

随着近几年个人自媒体的发展，一些乐于分享自身学习和实践心得的技术人员已经不再满足于一些互联网厂商提供的网络 blog 服务，希望能够通过自己购买服务器、域名等必要设备搭建属于自己，可以使用方便，去除冗余功能，且无广告能够提供完整交互和完美体验的个人网站，亦本次所需要研究和搭建的个人博客系统。

关于 React 框架，这是由美国 Facebook 公司创建和发布的一套前端框架，由于其易用的特性也已经广泛运用在当今的互联网社会中的各种各样的网站上。

2、研究现状

近几年，在互联网上也有着一些可以提供简单搭建个人网站的服务和程序，例如 wordpress 或者 github 的静态网站服务等，这些现有的程序只要在服务器上安装，并稍作配置即可搭建完成一个简单的博客系统。

这一类型的服务主要提供的服务有以下几点：

- 2.1 设定用户的管理权限
- 2.2 可以通过富文本编辑器编辑发表文章，发表的同时可以选择文章标签，设定访问链接等各种必要参数和设置。
- 2.3 可以提供静态 html 页面可供访问
- 2.3 提供访客的评论功能，并且可以做出审核。
- 2.4 可以切换自己喜欢的展示主题等。

关于 Wordpress 的功能点网上已经有很多的资料，便不再赘述。

3、课题研究内容

- 3.1 研究学习 React 技术框架以及 Node.js 的后端构建技术等

这一部分内容由于课堂上并没有过多设计，通过课外资料的学习进行补充自学，基本具备完善和搭建具体应用的能力。

3.2 确定本博客系统需要完成的基本和进阶功能点

通过与现有程序的对比比较,充分汲取现有系统的优点并且对于自己使用经验,对希望可以优化使用体验的点在本系统中进行修改和完善。

3.3 利用学习的技术和确定的功能点具体搭建整个博客系统

利用React技术栈、Node.js与Express后端技术、Mongodb的NoSQL数据库技术,并将前端部分利用Webpack进行工程化的构建,以建设本课题的主体部分。

3.4 对搭建起的系统进行测试,获得测试结果

利用一系列的测试用例对已经搭建起的系统进行测试,及时完善系统并且解决在测试中发现的问题

4、技术路线

4.1 将现在较为流行的敏捷开发的方法运用到系统的开发中去。

对于每个阶段,需要一定的user story,在具体的编程中通过功能的逐渐完善来完成user story。

在构建的部分,首先约定前后端交互的API,用于方便之后的编程和功能的完善。

4.2 在构建的全程中使用Git进行代码和版本管理。

4.3 将React技术栈(React、Redux等),Node.js(Express框架)以及MongoDB的NoSQL数据库应用到系统中去,这些框架和内容均利用JavaScript语言。

4.4 部分界面的布局将会使用到CSS布局以及HTML网页搭建的相关技术。

4.5 整个系统将会利用RESTful架构进行搭建,进行前后端分离。

5、关键技术介绍

5.1 React

在Web技术的发展之后,网页已经不仅仅是展现的工具,更可作为交互的工具。为了满足如此的需求,便有了React等一系列的前端交互框架。

React是一个声明式、组件化的前端框架,其主要负责的是类MVC软件结构中的V部分,也就是UI部分。通过编写拥有各自状态的组件,再由组件构成更

加复杂的界面。无需再用模版代码，通过使用 JavaScript 编写的组件你可以更好地传递数据，将应用状态和 DOM 拆分开来。更好的完成前后端开发的分离，对个人甚或是企业可以进行行业工作的细分和术业专攻。并且无论现有项目正在使用什么技术栈，你都可以随时引入 React 开发新特性。

React 的主要特征就是有状态的组件，React 组件使用一个名为 `render()` 的方法，接收数据作为输入，输出页面中对应展示的内容。状态主要体现在除了使用外部传入的数据以外（通过 `this.props` 访问传入数据），组件还可以拥有其内部的状态数据（通过 `this.state` 访问状态数据）。当组件的状态数据改变时，组件会调用 `render()` 方法重新渲染。

React 作为一个基础的框架，在其基础上也发展了很多基于其的功能扩展，本课题中也将进行使用。

5.2 Redux

随着基于 React 框架的开发日趋复杂，React 框架需要管理很多的 **state**（状态）。这些 **state** 可能包括服务器响应、缓存数据、本地生成尚未持久化到服务器的数据，也包括 UI 状态，如激活的路由，被选中的标签，是否显示加载动效或者分页器等等。

管理不断变化的 **state** 也变得逐渐困难。如果一个 **model** 的变化会引起另一个 **model** 变化，那么当 **view** 变化时，就可能引起对应 **model** 以及另一个 **model** 的变化，依次地，可能会引起另一个 **view** 的变化。直至你搞不清楚到底发生了什么。**state** 在什么时候，由于什么原因，如何变化已然不受控制。当系统变得错综复杂的时候，想重现问题或者添加新功能就会变得举步维艰。

为了解决这样的问题，Redux 这个 React 状态容器就出现了，它提供可预测化的状态管理。

Redux 是一个混合产物。它和一些设计模式及技术相似，但也有不同之处。它利用 React 提供的 Flux 思路进行状态管理，并且融合了一些当前比较新潮的函数式编程思想搭建了这个单向数据流管理系统。

5.2 Node.js 与 Express

Node.js 是 JavaScript 语言从前端向后端发展的产物。

简单的说 **Node.js** 就是运行在服务端的 **JavaScript**。**Node.js** 是一个基于 **Chrome JavaScript** 运行时建立的一个平台。

Node.js 是一个事件驱动 I/O 服务端 **JavaScript** 环境，基于 **Google** 的 **V8** 引擎，**V8** 引擎执行 **Javascript** 的速度非常快，性能非常好。而 **Express** 是一个简洁而灵活的 **node.js** **Web** 应用框架，提供了一系列强大特性帮助创建各种 **Web** 应用，和丰富的 **HTTP** 工具。使用 **Express** 可以快速地搭建一个完整功能的网站的后端部分。

通过 **Node.js** 和 **Express** 可以将前后端均利用 **JavaScript** 进行处理，大大减少了不同语言之间进行转换可能导致的错误和不便。

5.3 MongoDB

MongoDB 是由 **C++**语言编写，是一个基于分布式文件存储的开源 **NoSQL** 数据库系统。在高负载的情况下，添加更多的节点，可以保证服务器性能。**MongoDB** 旨在为 **WEB** 应用提供可扩展的高性能数据存储解决方案。**MongoDB** 将数据存储为一个文档，数据结构由键值(key=>value)对组成。**MongoDB** 文档类似于 **JSON** 对象。字段值可以包含其他文档，数组及文档数组。

正因为类似于 **JSON** 对象这个特性，其可以与 **JavaScript** 进行简单的交互和获取数据，不用进行多一步的转换，大大减低了过去 **SQL** 语言或者 **SSH** 框架的复杂性。

5.4 RESTful 架构

RESTful 架构，就是目前最流行的一种互联网软件架构。它结构清晰、符合标准、易于理解、扩展方便，所以正得到越来越多网站的采用。**REST** 这个词，是 **Roy Thomas Fielding** 在他 2000 年的博士论文中提出的。

REST，即 **Representational State Transfer** 的缩写。主要有如下几个表现特征：每一个 **URI** 表示一个资源，客户端和服务端之间传递这种资源，通过 **HTTP** 动词对服务器资源进行操作。

通过以上介绍的集中技术，基本可以通过 **ES6** 的 **JavaScript** 完成整个系统，提供了完善和一致的开发体验。

6、要解决的技术问题

- 6.1 前端后端的协调并进，在一个服务器上运行前端和后端两套服务。
- 6.2 决定前后端交互的 API，对于 RESTful 的深度理解。
- 6.3 数据在 React 框架中的转化和表现。
- 6.4 界面布局的具体风格以及界面框架的选择。
- 6.5 对于 Redux 构建系统产生的数据流动的理解。
- 6.6 利用 Webpack 进行前端工程化的实践，前端和后端的热更新简化调试操作
- 6.7 浏览数的具体统计

7、日程安排

序号	各阶段名称	起止日期
1	对于技术的学习和知识完善	2018/3/5-2018/3/18
2	User Story的确定和API约定	2018/3/19-2018/3/25
3	用户注册登录等基础功能	2018/3/26-2018/4/1
4	构建系统的管理具体功能(发布文章、文章用户管理等)	2018/4/2-2018/4/22
5	构建系统的浏览部分具体功能 (浏览文章，发表评论等)	2018/4/23-2018/5/6
6	系统整体测试	2018/5/7-2018/5/13
7	论文的撰写和完善	2018/5/13-2018/5/20

8、参考文献

- 1 <https://doc.react-china.org/> React 中文官方网站

2 <http://www.ruanyifeng.com/blog/2011/09/restful> 阮一峰 理解 RESTful 架构

3 陈 屹 著 《深入 React 技术栈》 人民邮电出版社 2016

4 <http://www.runoob.com/> 菜鸟教程网站

.....