

PaperFree检测报告简明打印版

比对结果（相似度）：

总体：18%

网络库：18%

编号：JSKB374NGCBDIXXI

标题：基于单页面应用的MEAN架构的个人博客系

作者：谢昆

长度：20759字符

标题：基于单页面应用的MEAN架构的个人博客系

时间：2015-05-28 20:43:16

比对库：学术期刊（1999-2015）、学术论文（硕博库1990-2015）、互联网资源

相似资源列表(学术期刊、学位论文)

1、相似度：(6.48%)

来源：学术期刊

篇名：《Node.js开发指南》试读：第一章：Node.js简介

2、相似度：(1.29%)

来源：学术期刊

篇名：nodeJs入门篇之认识nodejs

3、相似度：(0.85%)

来源：学术期刊

篇名：MongoDB - cendy

4、相似度：(0.77%)

来源：学术期刊

篇名：nosql篇mongodb初识与安装整理以及与关系型数据库对比

5、相似度：(0.66%)

来源：学术期刊

篇名：MongoDB安装及C#开发测试

6、相似度：(0.59%)

来源：学术期刊

篇名：MongoDB数据库从基础到高级应用详细说明

7、相似度：(0.57%)

来源：学术期刊

篇名：nosql篇mongodb初识与安装整理以及与关系型数据库对比

8、相似度：(0.54%)

来源：学术期刊

篇名：Ancient Chinese urban planning

9、相似度：(0.52%)

来源：学术期刊

篇名：NoSQL之MongoDB开篇

10、相似度：(0.48%)

来源：学术期刊

篇名：《用AngularJS开发下一代Web应用》试读

11、相似度：(0.4%)

来源：学术期刊

篇名：Free program user login logout internet Download

12、相似度：(0.31%)

来源：学术期刊

篇名：node.js入门及express.js框架

13、相似度：(0.13%)

来源：学术期刊

篇名：angularJS 笔记

14、相似度：(0.1%)

来源：学术期刊

篇名：mongodb简单使用

相似资源列表(互联网)

1、相似度：(2.8%)

标题：《数据绑定_用AngularJS开发下一代Web应用_红黑联盟读书频道》

<http://book.2cto.com/201312/37753.html>

2、相似度：(2.03%)

标题：《分布式文件存储数据库MongoDB_服务器知识学堂-中关村在线》

<http://server.zol.com.cn/270/2700424.html>

3、相似度：(2%)

标题：《NodeJS学习笔记 - 入门简介 | IT宅.com》

<http://www.itzhai.com/nodejs-study-notes-introduction-to.html>

4、相似度：(1.49%)

标题：《开源软件PK:Hadoop vs Apache 谁与争锋- 第3页- 开源资讯- LUPA...》

<http://www.lupaworld.com/article-215468-3.html>

5、相似度：(1.43%)

标题：《NodeJS学习笔记-JavaScript-第七城市》

<http://www.th7.cn/web/js/201310/13299.shtml>

6、相似度：(0.92%)

标题：《Java RESTful Web Service 实战 - 下载频道 - CSDN.NET》

<http://download.csdn.net/detail/hysxchina/8514421>

7、相似度：(0.84%)

标题：《译者序 - 51CTO.COM》

<http://book.51cto.com/art/201311/416400.htm>

8、相似度：(0.38%)

标题：《Rest|Restful初步认识 | 风芯の小站》

<http://1.liangtao.sinaapp.com/?p=639>

9、相似度：(0.29%)

标题：《【NOSQL版块推广】MongoDB 学习 - NoSQL论坛 - 51CTO技术论坛...》

<http://bbs.51cto.com/thread-1132453-1.html>

10、相似度：(0.19%)

标题：《[AngularJS] 1. Angular JS的五大特性- Javascript教程_JS教程_技术...》

<http://www.2cto.com/kf/201304/205069.html>

11、相似度：(0.12%)

标题：《AngularJS HTML编译器介绍_AngularJS_脚本之家》

<http://www.jb51.net/article/58248.htm>

全文简明报告：

山东工商学院

SHANDONG INSTITUTE OF BUSINESS AND TECHNOLOGY

毕业论文(设计)

GRADUATION THESIS (DESIGN)

论文(设计)题目

Title Of Thesis(Design) 基于单页面应用的MEAN架构的个人博客系统

分院(系别)

Department 计算机科学与技术学院

专业

Speciality 网络工程 班级

Class 112

论文(设计)作者

Author of Thesis(Design) 谢昆 论文完成日期

Date 2015年5月25日

论文(设计)指导教师

Advisor 刘志猛 指导教师职称

The Title of Advisor 讲师

基于单页面应用的MEAN架构的个人博客系统

A Personal Blog System Based on A Single Page Application Using MEAN Architecture

作者:谢昆

山东工商学院

指导教师对毕业论文(设计)的评语

Advisor' s Comments on Graduation Thesis (Design)

评语:

指导教师(签章)

Signature of Advisor

日期

Date

评阅人意见

评阅人姓名: 职称:

选项标准: A很同意 B同意 C基本同意 D不同意

分

项

评

价 评价项目 A B C D

选

题

质

量 1 选题符合专业培养目标,体现综合训练基本要求

2 题目难易适度

3 题目工作量适当

4 有理论意义或实际价值

能

力

水

平 5 查阅文献资料能力强

6 综合运用知识能力强

7 研究方案的设计能力强

8 研究方法和手段的运用能力强

9 外文应用能力强

成

果

质

量 10 文题相符

11 写作水平高

12 写作规范

13 篇幅适度

14 成果有理论或实际价值

总体评价: 优 ☐ 良 ☐ 中 ☐ 及格 ☐ 不及格 ☐

评阅人评语

评阅人签字: 年 月 日

答辩(评审)委员会意见

Appraisal of Defence Commission

答辩(评审)成绩

Mark of Defence

鉴定意见

Appraisal & Comments

主任(签章)

Signature of Dean

日期Date

基于单页面应用的MEAN架构的个人博客系统

[摘要] 本文以技术积累的个人博客所需的主要功能为出发点,阐述搭建个人博客系统的全过程。本系统采用HTML+CSS+JavaScript编写单页面应用程序,并使用 Bootstrap、AngularJS、Node.js、Express、MongoDB等技术和框架,同时系统整体上采用开发快速的、灵活性强的JavaScript 全栈式开发框架——MEAN,支持对文章的增删改查、标题搜索、点赞、人气排序、回收站、创建分类、分类浏览、文章前后翻篇;用户的登录、自动登录、退出;登录时密码的显隐、忘记密码、回到顶部按钮、页面路由切换时的动画效果等功能。

[关键词] 个人博客 单页面应用 MEAN架构 JavaScript

A Personal Blog System Based on A Single Page Application Using MEAN Architecture

Abstract:{ 56% : This article illustrates the whole process of establishing a personal blog system, ;} based on the main functions of individual skill accumulating needs. this system adopts HTML, CSS, JavaScript to create a single page application, and overall adopts the rapid in development, scalable JavaScript full-stack developing framework -- MEAN by using Bootstrap, AngularJS, Node.js, Express,{ 50% : MongoDB technologies and frameworks, ;} and supports the create, retrieve, update and delete operations of articles, the searching of title of article, favoring article, the sorting with the favor of articles, recycle bin, establishing category, classify browsing, article turning previous and next; user login,{ 54% : user auto login, user logout; ;} display password when filling login form, forgetting password, back to the top, the animation of page route changing functions.

Keywords: personal blog; single page application; MEAN; JavaScript

目录

第一章 绪论 1

1.1 问题的提出	1
1.2 课题研究现状及意义	1
1.3 课题研究的目的是内容	2
1.4 本文的主要工作	2
第二章 系统使用技术简介	3
2.1 单页面应用	3
2.2 MEAN堆栈的组成	3
2.3 MEAN堆栈的适用范围	6
第三章 系统的功能需求分析	8
3.1 系统的运行环境	8
3.2 用户管理	8
3.3 分类管理	10
3.4 文章管理	11
第四章 系统的概要设计	14
4.1 系统组织结构	14
4.2 系统的功能模块	15
4.3 系统的数据库设计	16
第五章 系统的详细设计	18
5.1 前端路由的设计	18
5.2 RESTFUL接口的设计	19
第六章 系统的前端单元测试	27
6.1 FILTER单元测试	27
6.2 SERVICE单元测试	28

6.3 CONTROLLER单元测试 29

6.4 DIRECTIVE单元测试 30

第七章 系统的部署与运行 31

7.1 将项目部署到LINUX UBUNTU服务器 31

7.2 使用FOREVER让项目持续运行 32

结论 33

致谢语 34

参考文献 35

第一章 绪论

1.1 问题的提出

技术积累对于技术人员来说是非常重要的事情,虽然平时在工作时解决了不少问题,但在很长时间以后,由于缺乏知识的积累和汇总,再遇到类似问题或相同问题时,不得不再回想甚至在众多资料中查找以前的解决方案,这对于个人来说既是时间上的浪费,也是知识上的遗忘。虽然网络上有着众多的网站都提供博客的功能来供大家发表技术博客,如:ITeye、CSDN、开源中国等,但是许多都是一个模子刻出来的,虽然可以个性化设置,但是无法尽可能地体现博主的个性和偏好。因此拥有一个自己搭建的博客系统来做技术积累,不仅可以让自己随意地改变博客的展示形式、沉淀知识以及方便他人的知识学习,也能让自己从零开始体验全栈开发的过程,进一步了解Web应用开发的流程。

1.2 课题研究现状及意义

在这个互联网知识更新换代如此频繁的时代,不断学习的能力和解决问题的能力对于技术人员来说是必须的,于是积累知识就显得额外重要。然而许多博客界面大体相似对于每个人可修改的设置内容少,于是,编写自定义的个人博客便由此出现,自己编写个人博客不仅可以将自己的技术积累以自己喜欢的方式记录下来,也能提升自己对Web开发一整套流程出现的问题的处理能力。

随着Web开发技术的不断发展,与以ASP、PHP、JSP为代表的动态页面技术不同的单页面应用产生,单页面应用是只有一张Web页面的应用。浏览器在第一次访问应用的时候会加载必需的HTML、CSS和JavaScript,之后所有的操作都通过JavaScript控制来在这张页面上完成。单页面应用有望成为未来Web开发技术的发展趋势,很多传统网站都在或者已经转型为单页面应用,新的单页面应用网站(包括移动平台上的)也如雨后春笋般呈现在人们的面前,如Gmail、Evernote、Trello等

相对于传统的Web开发堆栈LAMP(Linux、Apache、MySQL、PHP/Python), MEAN(MongoDB、Express、Node.js、AngularJS)这种新兴堆栈没有操作系统平台的

限制,它运行于JavaScript运行时环境上,MEAN开发堆栈采用JavaScript作为开发语言,并在应用的客户端、服务端、数据库的逻辑编写都采用该种语言。这种优势降低了开发人员在编写前后端逻辑时出错的可能性,也在一定程度上降低了应用响应的延迟,同时也极大地提高了开发效率。

1.3 课题研究的目的是内容

该个人博客系统实现用户未登录情况下对文章的查看、搜索,用户登录时的密码显隐、重置密码,用户登录后对文章的增删改查、回收站到用户退出等主要功能。文章查看时采用动画进行过渡,对进入和退出的子页面分别采用不同的切换动画,使得视觉上富有层次感;文章编写采用Simditor富文本编辑器自动生成HTML格式的文章内容,以使文章在页面上显示时与在编写时的格式一致,避免普通的HTML多行文本的内容不能自定义格式的问题;对文章标题的搜索采用简单的正则匹配方式,不区分大小写地将出现的关键字标红,以减少采用第三方框架的麻烦;删除文章时使用假删除的方式将文章放入回收站,而在回收站中可撤销删除和彻底删除,这种方式可避免用户误操作带来的难以恢复的后果。

博客系统采用JavaScript的全栈式架构MEAN,使开发的过程更快捷,同时使前端到后端到数据库的数据格式统一。

1.4 本文的主要工作

本文将介绍一个个人博客系统从概要设计到详细设计环境到部署完成的实施过程,该过程中将完成一个个人博客所需的主要功能。

第二章 系统使用技术简介

2.1 单页面应用

单页面应用也称为SPA(Single Page Application),在Web开发中,与多页面应用不同的是单页面应用会在用户第一次访问时将前端展示需要的HTML模板、CSS、JavaScript等文件一次性发送到用户的浏览器,当然,并不是所有应用都是这么多,也可以通过一系列前端优化,对于不是立刻用到的组件或文件实现懒加载。由于HTML模板被直接发送到了客户端,那么服务器就不需要负责数据和模板的混合之类的事,{ 54% : 反而只需要为客户端提供需要使用的数据即可。;}而在前端,组拼模板和数据便交给了前端框架,如AngularJS、EmberJS之类的MVC/MVVM框架。前端对页面显示和路由的控制采用JavaScript管理,向后端请求数据多采用异步AJAX方式。通常在后端搭建RESTful API,使得前后端更方便、快捷地实现功能。前端在访问RESTful API时,采用promise的编程方式,将异步处理的逻辑以同步形式编写。

2.2 MEAN堆栈的组成

MEAN是由MongoDB、Express、AngularJS和Node.js组合而成,同时本系统中采用Bootstrap美化前端界面。

2.2.1 MongoDB

MongoDB是目前非常流行的一种NoSQL(Not only SQL)数据库,它采用非常灵活的方式存储数据,这也是它能备受业内人员亲睐的原因之一。{ 50% : MongoDB作为一个文档型数据库, }很好的实现了面向对象的思想,而它最大的优势是所有的数据持久化操作都可以通过直接调用相应的方法来完成,不需要开发人员编写SQL语句。{ 74% : MongoDB是一个基于分布式文件存储的数据库,由C++语言编写而成。 :}{ 88% : 其目的在于为WEB应用提供可扩展的高性能数据存储解决方案。 :}MongoDB是面向集合存储的,{ 90% : 即数据被分组存储在数据集中,被称为一个集合每个集合在数据库中都有一个唯一的标识名,并且可以包含无限数目的文档。 :}{ 100% : 集合的概念类似关系型数据库里的表,不同的是它不需要定义任何模式。 :}{ 94% : 模式自由,意味着对于存储在MongoDB数据库中的文件,我们不需要知道它的任何结构定义。 :}{ 95% : 如果需要的话,我们完全可以把不同结构的文件存储在同一个数据库里。 :}{ 96% : 存储在集合中的文档,被存储为键/值对的形式。 :}{ 100% : 键用于唯一标识一个文档,为字符串类型,而值则可以是各种复杂的文件类型。 :}这种存储形式为BSON(Binary JSON)。高性能、易部署、存储数据非常方便是MongoDB主要的特点。

MongoDB适合作为信息基础设施的持久化缓存层,{ 76% : 适合实时的插入、更新与查询,并具备应用程序实时数据存储所需的复制及高度伸缩性; }适合文档化格式的存储及查询,{ 95% : 适合由数十或数百台服务器组成的数据库 :}

对于要求高度事务性的系统。例如对于银行或会计等需要大量原子性的复杂事物的应用程序来、传统的商业智能应用以及复杂的表级联查询MongoDB不是最优的选择。

2.2.2 Express

{ 67% : Express是目前最稳定的、使用最广泛的, :}{ 84% : 而且也是Node.js官方推荐的唯一的一个Web开发框架。 :}Express是一个简洁、灵活的Node.js Web应用开发框架,它提供一系列强大的特性,能够帮助我们创建各种WEB和移动设备应用。Express的API(Application Programming Interface)中提供了丰富的 HTTP 快捷方法和任意排列组合的Connect中间件,使我们创建健壮、友好的API变得既快速又简单。Express 不对Node.js已有的特性进行二次抽象,而只是在它之上扩展了Web应用所需的基本功能。{ 100% : Express除了为HTTP模块提供了更高层的接口外, :}还实现了路由控制、模板解析支持、动态视图、用户会话、静态文件服务、CSRF(Cross-Site Request Forgery)保护、访问日志、缓存、插件支持、错误控制器等功能。虽然Express提供了如此多的功能和接口,但是它并不是一个全能的框架,{ 71% : 它只是一个轻量级的Node.js Web开发框架, :}它做的更多的是对常用操作的封装,而具体的实现则是要依靠其他的模块或者插件来完成。

2.2.3 AngularJS

{ 82% : AngularJS是一款来自Google的前端JavaScript框架, :}{ 96% : 该框架已经被应用到了Google的多款产品中。 :}{ 88% : 这款框架最核心的特性有:MVC/MVVM、模块化、自动化双向数据绑定、语义化标签、依赖注入等等。 :}

AngularJS主要用于构建单页面应用。多页面Web应用会在服务端创建HTML,{ 58% : 并把HTML和数据装配并混合起来, :}然后再把生成的页面发送到浏览器中。在某种程度上,大部分的单页面应用,即AJAX应用,也会做同样的事。然而AngularJS的处理方式完全不

同,{ 71% : 在AngularJS中,模板和数据都会被发送到浏览器中,;}然后在客户端进行装配。于是,服务器便只需要为这些模板提供一些静态的资源,然后为这些模板提供所需要的正确数据即可。

使用MVC(Model-View-Controller)模型能够让应用更加易于扩展、维护和测试。MVC模型核心理念是业务逻辑、数据、界面显示分离。{ 77% : 在AngularJS应用中,;}视图就是文档对象模型DOM(Document Object Model),{ 66% : 控制器就是JavaScript类,数据则是被存储在对象的属性之中。;}

使用jQuery我们可以单独刷新DOM中的局部内容,而不是刷新整个页面。{ 90% : 在jQuery中,我们把HTML模板字符串和数据混合起来,然后把获得的结果插入DOM中指定的位置,插入的方式是把结果设置给一个占位符元素的innerHTML属性。;}{ 90% : 但是如果需要把最新的数据插入到UI中,或者根据用户输入的来修改数据的时候,就需要做很多非常繁琐的工作来保证数据的状态的正确,并且UI和JavaScript的属性要同时正确。;}{ 67% : 而在AngularJS中,;}可以使用数据绑定实现这一过程,{ 71% : 通过声明UI中的某个部分需要映射到某个JavaScript属性,然后两者便能自动同步。;}{ 84% : 这样一来,在编写视图和模型的时候可以减少代码量。;}

{ 72% : AngularJS可以通过指令来扩展HTML,;}从而让浏览器识别新的语法。通过AngularJS内置的指令,可以把常用的视图定义成模板,以创建可服用的组件。

2.2.4 Node.js

Node.js既不是一种独立的语言,也不是一个JavaScript框架,更不是浏览器端的库,{ 79% : 而是一个让JavaScript运行在服务端的开发平台,它让JavaScript在服务端能与PHP、Python、Perl和Ruby平起平坐。;}

Node.js是一个为实时WEB应用开发而诞生的平台,{ 97% : 从诞生之初就充分考虑了在实时响应、超大规模数据要求下架构的可扩展性。;}{ 98% : 这使得它摒弃了传统平台依靠多线程来实现高并发的设计思路,而采用了单线程、异步式IO、事件驱动式的程序设计模型。;}{ 100% : 这些特性不仅带来了巨大的性能提升,还减少了多线程程序设计的复杂性,进而提高了开发效率。;}

{ 75% : Node.js让JavaScript可以运行在浏览器之外,;}{ 84% : 它实现了文件系统、模块包、操作系统API、网络通信等核心JavaScript没有或者不完善的功能。;}{ 54% : Node.js的运行环境是Chrome的JavaScript V8引擎,;}这个目前世界上公认的JavaScript执行速度最快的引擎。在数次浏览器引擎革命之后,{ 79% : V8的即时编译JIT(Just-In-Time Compilation)执行速度已经快接近本地代码的执行速度。;}{ 56% : 由于Node.js不运行于浏览器端,因此不存在浏览器兼容的繁琐问题,;}这样一来便可以使用JavaScript的全部特性。

虽然浏览器端的JavaScript做的事是与WEB应用相关的,而运行于Node.js平台上的JavaScript则可以轻松地开发命令行工具、单元测试工具、客户端JavaScript编译器、具有复杂逻辑的网站、基于社交网络的大规模Web应用等等。{ 57% : Node.js内建了HTTP服务器支持,可以仅通过几行代码就能轻易地搭建起一个HTTP服务器。;}

{ 60% : Node.js采用了异步式IO与事件驱动, ;}这也成为其最大的亮点。对于解决高并发的问題,{ 85% : 传统架构是采用多线程的方式,为每个业务逻辑提供一个系统线程,通过系统线程的切换来弥补同步式IO调用时的时间开销。 ;}{ 78% : 而Node.js使用的是单线程,对所有的IO都采用异步式的请求方式,从而避免了上下文的切换。 ;}{ 95% : Node.js在执行过程中会维护一个事件队列,程序在执行时进入事件循环等待下一个事件到来,每个异步式IO请求完成后会被推送到事件队列,等待程序进程进行处理。 ;}

2.2.5 Bootstrap

Bootstrap是由Twitter工程师开发的一款目前最受欢迎的HTML、CSS和JavaScript框架。由于现在终端设备的多样化,如:笔记本电脑、台式机、平板电脑、智能手机等。不同的终端设备往往设备显示器的大小不一样,对于同一张网页在不同的终端设备上的显示效果自然也会有不同,笔记本电脑屏幕多是14寸,台式机则有19、20甚至更高的尺寸,而手机上的分辨率与屏幕尺寸则更加繁多。对于众多的屏幕尺寸导致网页显示不尽人意的问題,最终出现了一种解决办法,即响应式布局。响应式布局做的就是适配不同设备下的展示形式,使得形式与效果统一。而Bootstrap正是这样一款框架,响应式布局的框架并不少,如:Foundation、Skeleton等等,不过Bootstrap凭借其巧妙地设计理念和样式风格成为其中最受欢迎的选择,从而成为了开发多终端下的响应式布局的首选前端框架。同时作为一个CSS框架,使用Bootstrap能够让CSS基础薄弱的开发人员也能编写出简洁而又美观的页面。

2.3 MEAN堆栈的适用范围

软件工程史上没有银弹。JavaScript全堆栈的MEAN也不例外。MEAN有它的局限性,同时也有它的优越性。MEAN中的每个成员都有其局限的地方,只有当它们优势的方面产生交集时,才能将总体的优势发挥到极致。

MongoDB由于是非关系型数据库,对于要解决高度事务性的系统、传统的商业智能以及SQL的问题时,会显得非常劣势,而这种情况应该是转向SQL的时候了。Express是基于Node.js的轻量级WEB开发框架,其局限性取决于Node.js,AngularJS在需要做SEO(Search Engine Optimization)的内容网站、交互频繁的游戏体验等应用不是最好的选择。{ 60% : Node.js由于是单线程,采用异步式IO和事件驱动, ;}在计算密集型的应用、单用户多任务型应用、逻辑十分复杂的事务以及Unicode与国际化方面不适用。

了解了各成员不适用的场景后,从中可以总结一些适用的场景,即处理逻辑简单、访问频繁、CRUD(Create Retrieve Update Delete)、SPA、数据存储格式自由且伸缩性强、IO密集等等。由此可以用于创建社交网站、博客、即时聊天等应用。

第三章 系统的功能需求分析

3.1 系统的运行环境

3.1.1 服务器环境

1.硬件环境

(1) CPU数:1个或以上

(2) 数据库数据存储所在分卷的容量:4G以上

(3) 内存容量:1G或以上

2.软件环境

(1) gcc

(2) build-essential

(3) JavaScript服务端运行时环境Node.js v0.10.32

(4) NoSQL数据库MongoDB 2.6.6

(5) 操作系统:Linux Ubuntu

3.1.2 客户端环境

1.硬件环境

(1) CPU数:2个或以上

(2) 内存容量:2G或以上

2.软件环境

(1) Internet Explorer 9以上

(2) Chrome 15.0.874以上

(3) Firefox 8.0.1以上

(4) Opera 11.52以上

(5) 操作系统:Windows 7

3.2 用户管理

实现用户权限控制,将用户分为普通用户和管理员用户。对普通用户只能查看文章,搜索文章,给文章点赞;而管理员用户可以登录系统,并通过忘记密码功能修改密码,以及自动登录等功能。

图3.1 找回密码

图3.2 用户登录

图3.3 查看密码

图3.4 密码重置邮件

图3.5 重置密码

3.3 分类管理

通过用户的权限控制,所有用户都能查看分类,只有管理员用户才能创建分类。

图3.6 查看分类

图3.7 创建分类

3.4 文章管理

所有用户都能查看文章,并通过分页功能查看不同分页的文章,分页支持文章的前后翻页和上下级的翻页以及直接选页。

图3.8 查看文章

点击文章标题可进入文章详情页面。

图3.9 文章详情

在任意页面都可以使用搜索功能,对文章标题进行搜索,搜索的返回的结果中对关键字标红,文章点赞。

图3.10文章标题搜索

图3.11文章点赞

而管理员用户登录后可以对文章做删除、更新、创建、操作文章回收站等功能

图3.12 新建文章

图3.13 更新文章

图3.14 文章回收站

图3.15文章回收站操作

第四章 系统的概要设计

本章主要介绍系统的组织结构、功能模块、数据库设计。

4.1 系统组织结构

MEAN是一个全栈式的JavaScript解决方案,是由MongoDB、Express、AngularJS和Node.js组合而成,它可用于创建快速、健壮和可维护的WEB应用程序。本系统采用前后端分离,前端让AngularJS处理与用户交互的逻辑和页面的展示,并用Bootstrap作为CSS框架统一界面风格,后端则采用Node.js的轻量级Web开发框架Express处理用户请求,对数据的存储使用灵活性强的MongoDB。

图4.1 MEAN的基本架构

{ 60% : 由于是一个JavaScript的架构, :}其前端-后端-数据库之间的交互使用的都是JSON(JavaScript Object Notation),采用JSON交互数据,易于阅读和编写,同时也能方便机器的解析和生成,加快数据传输。从浏览器端用户的输入,经过AngularJS将数据组合成JavaScript对象以JSON的方式传递到Express,Express处理JSON数据时,无需对数据进行类型转换,直接可将数据存入数据库中,而MongoDB中采用BSON(Binary JSON)的方式存储数据。需要取数据时,Express向数据库发出请求,并获得数据,此时数据格式为JSON,Express将数据以JSON的方式发送到前端的AngularJS,收到数据的AngularJS便能按访问JavaScript对象的方式使用JSON,并将内容通过浏览器呈现给用户。

图4.2 前端-后端-数据库交互

采用MEAN的架构之所以能如此快速的进行开发与构成这种架构的框架和技术是紧密相关的。

4.2 系统的功能模块

根据个人博客需求分析,系统可划分出三大功能模块:分类管理、文章管理、用户管理。

系统采用内置用户的方式,在系统启动后,数据库自动生成特定用户的信息,对于默认密码支持密码找回以对其进行修改。登录时,用户不确定是否输入了正确密码,可通过密码显隐查看当前输入的密码明文。登录时使用自动登录,省去下次登录时仍需要输入密码的烦恼。

文章在展示时,任何人都可以直接使用文章列表、文章分类、人气排行、标题搜索、文章浏览的功能,对于敏感操作,使用权限控制,只有用户登录后才能看到回收站、文章删除、文章发表、文章更新的操作链接。

文章的分类存在于分类管理中,分类管理中包含对分类的创建和浏览,同样使用权限控制,创建分类只能由用户进行。

图4.3 系统功能模块结构图

4.3 系统的数据库设计

系统采用的是NoSQL数据库存储数据,{ 66% : MongoDB是一个开源的、文档型数据

库,;它为简化开发和提升扩展性而设计。{ 61% : 在文档型的MongoDB中,一条记录就是一个文档,;文档是由域/值对组成的,并且类似于JSON,一个域的值可以是别的文档、数组或者文档组成的数组。

图3.4 数据库文档结构图

文档中的_id和__v为数据库自动生成的。

图3.5 分类文档的数据格式图

文章文档中delete用于文章的标识删除,favori记录文章的被点赞数,而content中则存储包含HTML标记的文章内容。

图3.6 文章文档的数据格式图

用户文档中使用key保存重置密码的标识,而key_generate_time用于处理标识过期,auto_login用于保存自动登录的cookie。

图3.7 用户文档的数据格式图

第五章 系统的详细设计

5.1 前端路由的设计

前端采用MVC模型的AngularJS框架,在SPA中,通过JavaScript控制路由的访问和页面的切换,由于AngularJS自带的ng-Route对于嵌套使用路由支持的不够好,在此使用UI-Router管理路由的配置。

UI-Router中State表示路由的名称,abstract表示是否为只能通过继承才可访问的路由,URL即浏览器地址栏上显示的内容。由于路由可以聚合,也就是说URL也能聚合。以app作为根节点的路由并设置可嵌套路由的特定区域,使user和articles路由可以继承它,从而使不同类型的操作使用不同的界面,而user和articles中各设置可嵌套的区域,使得具体操作的页面可以通过路由聚合的方式在其父路由中呈现。对于不同的嵌套区域可采用不同的页面加载的展示动画,使页面动态性更强。

表 5.1 前端路由

state	abstract	URL
app	true	/app
app.articles	true	/articles
app.user	true	/user
app.user.password	false	/password/{token}

app.articles.article_list false /article_list/{page}/{category}/{keyword}

app.articles.article_detail false
/article_detail/{page}/{position}/{id}/{category}/{keyword}

app.articles.edit false /edit/{page}/{position}/{id}

app.articles.article_trash false /article_trash

图5.1 前端路由结构图

5.2 RESTful接口的设计

RESTful是一种软件架构风格,{ 100% : 设计风格而不是标准,只是提供了一组设计原则和约束条件。 :}{ 100% : 它主要用于客户端和服务端交互类的软件。 :}{ 100% : 基于这个风格设计的软件可以更简洁,更有层次,更易于实现缓存等机制。 :}

REST要求客户端和服务端是无状态通信,每次通信中应包含请求所需的全部内容。而HTTP方法是无状态的,通过URI确定每个资源的位置,并使用HTTP的PUT、DELETE、POST、GET请求方式对资源作相应的增删改查操作。

对于本应用中的通信采用响应正文中包含status自定义状态码的方式处理自定义错误,使用message来传递响应数据和提示信息。

5.2.1 自定义状态码

状态码 提示信息

200 操作成功/获得数据

500 服务器内部错误

用户接口

10101 用户名或密码错误

10102 用户已在线

10103 缺少参数 id

10104 无用户登录

10105 邮件发送失败

10106 标识不存在

10107 标识过期

10108 修改密码失败

分类接口

10201 保存失败

10202 缺少参数 name

文章接口

10301 暂无数据

10302 缺少参数

10303 文章未找到

10304 缺少参数 id

10305 更新失败

10306 参数不能为空

10307 删除失败

10308 拒绝服务,请勿输入#、(、)、\\等不合法字符

10309 不正确的请求

5.2.2 用户接口

1. 用户登录

url <http://xxx.xxx.xxx/user/login>

请求方式 POST

支持格式 JSON

请求参数 名称 类型 说明

username string 用户名

password string md5加密后的密码

cookie string 用户自动登录标识

返回结果字段 名称 类型与范围 说明

status int 请求响应的状态码

message string/object 提示信息/当前用户信息(非敏感)

2. 用户退出

url http://xxx.xxx.xxx/user/logout

请求方式 GET

支持格式 JSON

请求参数 名称 类型 说明

id string 用户id

返回结果字段 名称 类型与范围 说明

status int 请求响应的状态码

message string 提示信息

3. 获取当前用户信息

url http://xxx.xxx.xxx/user/current_user

请求方式 GET

支持格式 JSON

请求参数 名称 类型 说明

无 无 无

返回结果字段 名称 类型与范围 说明

status int 请求响应的状态码

message string/object 提示信息/当前用户信息(非敏感)

4. 发送重置密码邮件

url http://xxx.xxx.xxx/user/reset_password

请求方式 GET

支持格式 JSON

请求参数 名称 类型 说明

无 无 无

返回结果字段 名称 类型与范围 说明

status int 请求响应的状态码

message string 提示信息

5. 通过key的存在与否和过期时间检查key的有效性

url <http://xxx.xxx.xxx/user/password/token/>

请求方式 GET

支持格式 JSON

请求参数 名称 类型 说明

key string 重置密码链接中的token

返回结果字段 名称 类型与范围 说明

status int 请求响应的状态码

message string 提示信息

6. 重置为新密码

url <http://xxx.xxx.xxx/user/password/new>

请求方式 POST

支持格式 JSON

请求参数 名称 类型 说明

p string md5加密后的新密码

key string 重置密码链接中的token

返回结果字段 名称 类型与范围 说明

status int 请求响应的状态码

message string 提示信息

5.2.3 分类接口

1. 获取所有的分类

url http://xxx.xxx.xxx/category

请求方式 GET

支持格式 JSON

请求参数 名称 类型 说明

无 无 无

返回结果字段 名称 类型与范围 说明

status int 请求响应的状态码

message string/object 提示信息/分类信息

2. 新增分类

url http://xxx.xxx.xxx/category/:name

请求方式 PUT

支持格式 JSON

请求参数 名称 类型 说明

name string 分类的名称

返回结果字段 名称 类型与范围 说明

status int 请求响应的状态码

message string 提示信息

5.2.4 文章接口

1. 文章回收站

url `http://xxx.xxx.xxx/article_trash`

请求方式 GET

支持格式 JSON

请求参数 名称 类型 说明

name string 分类的名称

返回结果字段 名称 类型与范围 说明

status int 请求响应的状态码

message string/

object数组 提示信息/

已标记删除的所有文章信息

2. 文章人气排行

url `http://xxx.xxx.xxx/article_hot`

请求方式 GET

支持格式 JSON

请求参数 名称 类型 说明

无 无 无

返回结果字段 名称 类型与范围 说明

status int 请求响应的状态码

message string/

object数组 提示信息/

点赞数排名前9的文章信息

3. 文章点赞

url http://xxx.xxx.xxx/article_favor

请求方式 GET

支持格式 JSON

请求参数 名称 类型 说明

id string 文章的id

返回结果字段 名称 类型与范围 说明

status int 请求响应的状态码

message string 提示信息

4. 新建文章

url http://xxx.xxx.xxx/article_op

请求方式 GET

支持格式 JSON

请求参数 名称 类型 说明

content string 文章内容

create_time date 文章创建日期

category string 文章分类

title string 文章标题

返回结果字段 名称 类型与范围 说明

status int 请求响应的状态码

message string 提示信息

5. 更新文章

url http://xxx.xxx.xxx/article_op/:id

请求方式 POST

支持格式 JSON

请求参数 名称 类型 说明

content string 文章内容

id string 文章的id

category string 文章分类

title string 文章标题

返回结果字段 名称 类型与范围 说明

status int 请求响应的状态码

message string 提示信息

6. 删除文章

url http://xxx.xxx.xxx/article_op/:id/:position

请求方式 DELETE

支持格式 JSON

请求参数 名称 类型 说明

position int -1:彻底删除,其他标记删除

id string 文章的id

返回结果字段 名称 类型与范围 说明

status int 请求响应的状态码

message string 提示信息

7. 分页获取所有文章

url http://xxx.xxx.xxx/article_list

请求方式 GET

支持格式 JSON

请求参数 名称 类型 说明

p int 分页的页码

keyword string 搜索关键字

返回结果字段 名称 类型与范围 说明

status int 请求响应的状态码

message string/

object数组 提示信息/

分页返回的文章信息

8. 获取具体文章

url http://xxx.xxx.xxx/article_op/:id/:flip/:position

请求方式 GET

支持格式 JSON

请求参数 名称 类型 说明

id int 文章的id

返回结果字段 名称 类型与范围 说明

status int 请求响应的状态码

end int -1文章为第一篇,1文章为最后一篇,0文章处在中间,-2文章既是第一也是最后

position int 当前文章所处的页码

message string/

object数组 提示信息/

分页返回的文章信息

9. 获取具体文章

url http://xxx.xxx.xxx/article_op/:id/:flip/:position

请求方式 GET

支持格式 JSON

请求参数 名称 类型 说明

id int 文章的id

返回结果字段 名称 类型与范围 说明

status int 请求响应的状态码

end int -1文章为第一篇,1文章为最后一篇,0文章处在中间,-2文章既是第一也是最后

position int 当前文章所处的页码

message string/

object数组 提示信息/

分页返回的文章信息

10. 上传图片

url http://xxx.xxx.xxx/article/img_upload

请求方式 POST

支持格式 JSON

请求参数 名称 类型 说明

file file 上传的图片信息

返回结果字段 名称 类型与范围 说明

status int 请求响应的状态码

success boolean true上传成功,false上传失败

file_path string 图片的引用

message string 提示信息

第六章 系统的前端单元测试

本章主要介绍使用Karma对AngularJS前端功能进行单元测试,Karma是一个基于Node.js的JavaScript测试执行过程管理工具。该工具可用于测试所有主流Web浏览器,也可集成到持续集成工具,也可和其他代码编辑器一起使用。Karma测试工具的一个强大特性就是可以监控文件的变化,然后自动重新执行测试,并通过控制台界面显示测试结果。Jasmine是单元测试框架。

通过Karma与Jasmine可以对AngularJS项目中的controller、service、filter、directive进行单元测试,使用依赖注入的方式将需要的服务注入到测试环境,并在Karma的配置文件中配置需要进行单元测试的JS文件。

图6.1配置需要测试的JS文件

6.1 filter单元测试

对filter中的日期转换过滤器测试其对正常日期格式的转换结果是否正确和测试HTML转换过滤器对HTML转换为可信任的HTML的结果。

图6.2 filter单元测试逻辑

6.2 service单元测试

对service中的articleRest的GET、DELETE、PUT、POST分别测试其返回结果,并使用\$httpBackend模拟后端数据请求的响应。

图6.3 articleRest GET方法的单元测试

图6.4 articleRest DELETE方法的单元测试

图6.5 articleRest PUT方法的单元测试

图6.6 articleRest POST方法的单元测试

6.3 controller单元测试

对controller中的loginCtrl登录控制器进行单元测试,测试其登录控制器中数据的初始化和密码显隐功能是否正常。

图6.7 loginCtrl逻辑的单元测试

6.4 directive单元测试

对directive中的rightSideBar指令进行单元测试,测试其指令被使用时调用的模板是否符合指令定义时的模板。

图6.8 测试rightSideBar的模板

第七章 系统的部署与运行

7.1 将项目部署到Linux Ubuntu服务器

7.1.1 部署项目

NPM(Node Package Manager)是Node.js的包管理器,其功能类似于Java开发时使用的MAVEN。NPM用于管理Node.js项目的模块之间的依赖。通过读取项目根目录下的package.json文件,并解析其中的依赖关系,根据每个依赖去NPM官网上下下载相关模块。

使用SecureCRT软件通过SSH(Secure SHell)登录到Linux服务器,使用filezilla软件通过ftp将项目上传到服务器。启动MongoDB并设置好数据文件存储位置。通过Linux命令进入本项目的根目录下,执行npm install安装项目中使用的依赖模块,并执行nodemon bin/www启动项目,项目启动后,编写一篇文章,并且创建一个分类,待分类和文章创建成功后,使用MongDB命令编写db.articles.ensureIndex({create_time:1}); db.articles.ensureIndex({create_time:-1});db.categories.ensureIndex({name:1},{unique:true});三条语句为articles和categories集合创建索引,至此,项目部署完成。

7.1.2 让项目共享80端口

通过Nginx可以让我们在访问本项目的网站时不用再输入开发调试时的端口号,利用Nginx反向代理让项目与系统共享80端口。在安装好Nginx后,修改配置文件/etc/nginx/nginx.conf,将其中listen为80的server下的server_name修改为我们项目部署的IP或域名,再将server下的location / 下的proxy_pass设置为我们项目的IP和端口形式。此后只需要输入域名或者IP就能够访问到该项目。Nginx可以针对不同的请求传递给不同的服务,如下图:

图7.1 Nginx反向代理流程图

7.2 使用forever让项目持续运行

通过SecureCRT部署好项目之后,项目确实正常运行了,但是一旦断开与Linux服务器的连接,刚刚部署好的项目立马就停下了,要想让项目不因为断开SSH与服务器的连接而停止运行的话,{ 52% : 可以通过Node.js社区中提供的forever模块。}forever不仅能让你的应用在你断开SSH连接之后继续运行,同时也能使应用在崩溃的时候重新启动。

使用npm install forever -g将该模块安装到全局环境中,同时在项目的根目录下使用npm install forever-monitor -save将其监视模块保存到项目中。随后使用forever start bin/www启动项目即可。

结论

本论文针对博客记事的主要需求,采用前后端分离,同时使用RESTful API设计了个人博客系统。系统实现分类管理、文章管理、用户管理三大主要模块。包含分类创建、分类查询;

文章的增删改查、文章标题搜索、文章回收站、文章人气排行、文章分页列表;用户登录、用户退出、用户密码找回和用户自动登录等功能。系统采用HTML、CSS、JavaScript开发,使用JavaScript全栈式开发框架MEAN,采用Bootstrap作为前端CSS框架解决前端响应式布局的问题,以及部分浏览器兼容性问题,采用AngularJS作为前端MVC框架,处理页面展示和与用户的交互逻辑,以及与后端RESTful API的联系,采用Node.js作为服务端搭建HTTP服务,同时采用Node.js推荐的轻量级Web开发框架Express搭建RESTful API处理前端的请求,同时访问数据库,数据库采用文档型的MongoDB,使得后端逻辑的编写非常方便、快捷,并创建数据库索引,加速数据库对某些服务的查询速度。本系统采用MEAN的全栈式架构而非一味的使用重量级的架构或框架,目的在于对症下药,实现快速开发。

虽然,MEAN架构在本系统的使用中有其明显的优势,但是,本文中所述的内容并没有涵盖全部的系统开发内容,系统在许多细微的处理上是不到位的。其中最重要的是系统的安全方面。由于网络中对一些信息摘要算法如md5、sha1、sha256、sha512等算法采用建立巨型数据库的方式暴力破解,导致这些算法的安全性产生问题,虽然可以通过对明文多次加密,但也不排除被破解的可能。然而对于一些现有的加强版算法将成为下一个选择。信息加密不仅仅是对传输过程中的某部分信息加密,同时应该对整个信息加密,系统采用了http这种虽然快但是不安全的方式,容易受到恶意的攻击,转向使用https的连接方式将能够更好地保护信息的内容。然后光只有前端做加密是远远不够的,因为前端只是一个客户端,它不是必需的,而真正提供服务的是后端,不仅仅是前端需要加密,后端更要加密,包括对每次请求的最大字节数、文件上传时的最大体积、访问者的身份校验、后端的错误日志、异常处理、请求的记录,数据库定期备份等等。当然,不到位细节不仅限于安全,还有用户体验和系统总体的结构优化。如:采用静态文件服务器提供CSS、JavaScript等静态的文件、采用懒加载的方式提高页面的响应速度、使用浏览器缓存加速页面显示等等。

对于上述的这些问题,在以后我都会进一步完善,并加入新的功能使其能够成为一个让自己满意的作品。

致谢语

大学四年匆匆而过,时间在流去的同时也给我带来了我想要的东西,总的来说我的大学生活中没有遗憾,不论是从思想上、能力上或是生活上我都比以前有很大的进步。四年的大学生活中,我收获到了益友、遇到了良师,在等待中抓住了良好的机遇、在自己最能吃苦的时候选择奋斗,用学习作为对自己最好的投资、用行动证明了自己的能力。我能有这样的成就,是与我身边的每个人息息相关。

人的一生需要有人帮助,有人监督和贵人指点。

感谢刘志猛老师的细心指导和大力支持,从毕业设计的选题到论文的完成,刘志猛老师使我在整个毕业设计的完成进度上有总体的把握,并且在许多实际问题为我提供指导,刘志猛老师严谨治学的态度值得我去学习。

环境影响了人同时塑造了人。感谢我们的计算机科学与技术学院,给我提供了良好的人际交流平台、学习环境和氛围以及能力提升的机会。

感谢帮助过我的同学、以及为我论文写作提出建议的同学,你们的付出为我在毕业设计的

过程中收获不少。

最后,感谢百忙之中抽出时间来审阅此文和参加本论文答辩的老师,向老师们致以深深的谢意。

参考文献

- [1] Brad Green, Sbyam Sesbadri. 用AngularJS开发下一代Web应用[M]. 大漠穷秋译. 北京:电子工业出版社,2013.
- [2] BYVoid. Node.js开发指南[M]. 北京:人民邮电出版社,2012.
- [3] Kristina Chodorow. MongoDB 权威指南[M]. 邓强,王明辉译. 北京:人民邮电出版社, 2014.
- [4] Shyam Seshadri, Brad Green. OReilly.AngularJS.Up.and.Running[M]. O' Reilly Media, Inc, 2014.
- [5] Tom Hughes-Croucher Mike Wilson. Node即学即用[M]. 郑达韩译. 北京:人民邮电出版社, 2013.
- [6] Mike Cantelon, Marc Harter, T.J. Holowaychuk, Nathan Rajlich. Node.js in Action[M]. Manning Publications, 2013.
- [7] Douglas Crockford. JavaScript语言精粹[M]. 赵泽欣 鄢学鹏译. 北京:电子工业出版社,2009.
- [8] David Flanagan. JavaScript权威指南[M]. 淘宝前端团队译. 北京:机械工业出版社,2012.
- [9] 陶国荣. HTML5 实战[M]. 北京:机械工业出版社,2011.
- [10] UI Router. [http://angular-ui.github.io/ui-router/site/#/api/ui.router.state.\\$state](http://angular-ui.github.io/ui-router/site/#/api/ui.router.state.$state)
- [11] Andris Reinman. andris9/Nodemailer. <https://github.com/andris9/Nodemailer>
- [12] MEAN - Full-Stack JavaScript Using MongoDB, Express, AngularJS, and Node.js. <http://mean.io/#/>
- [13] foreverjs/forever. <https://github.com/foreverjs/forever>
- [14] 快速搭建Web环境 Angularjs + Express3 + Bootstrap3 | 粉丝日志. <http://blog.fens.me/angularjs-express3-bootstrap3/>

-
- [15] Tim Wood moment/moment. <https://github.com/moment/moment/>
 - [16] 全局 CSS 样式•Bootstrap 中文文档. <http://v3.bootcss.com/css/>
 - [17] AngularJS — Superheroic JavaScript MVW Framework. <https://angularjs.org/>
 - [18] The MongoDB 3.0 Manual — MongoDB Manual 3.0.3.
<http://docs.mongodb.org/manual/>
 - [19] Mongoose ODM v4.0.3. <http://mongoosejs.com/>
 - [20] About Documentation | Node.js. <https://nodejs.org/documentation/>
 - [21] Express - 中文文档 - Node.js Web 应用开发框架. <http://www.expressjs.com.cn/>
 - [22] Simditor. <http://simditor.tower.im/>

检测报告由PaperFree文献相似度检测系统生成
Copyright2007-2015 PaperFree