

# PaperPass旗舰版检测报告

## 简明打印版

### 比对结果(相似度):

总体: 18% (总体相似度是指本地库、互联网的综合对比结果)  
本地库: 8% (本地库相似度是指论文与学术期刊、学位论文、会议论文、图书数据库的对比结果)  
期刊库: 6% (期刊库相似度是指论文与学术期刊库的对比结果)  
学位库: 5% (学位库相似度是指论文与学位论文库的对比结果)  
会议库: 1% (会议库相似度是指论文与会议论文库的对比结果)  
图书库: 3% (图书库相似度是指论文与图书库的对比结果)  
互联网: 11% (互联网相似度是指论文与互联网资源的对比结果)

报告编号: 5CB222DC5C947QWPV

检测版本: 旗舰版

论文题目: [降重]基于vue的个人博客的设计与实现

论文作者: 周会艳

论文字数: 12726字符(不计空格)

段落个数: 395

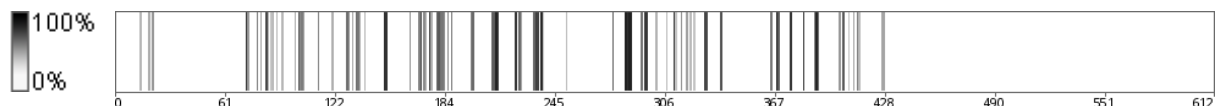
句子个数: 612句

提交时间: 2019-4-14 1:56:44

比对范围: 学术期刊、学位论文、会议论文、书籍数据、互联网资源

查询真伪: <http://www.paperpass.com/check>

### 句子相似度分布图:



### 本地库相似资源列表(学术期刊、学位论文、会议论文、书籍数据):

1. 相似度: 1% 篇名: 《博客文化中的自我认同与价值审视》  
来源: 学术期刊 《内蒙古农业大学学报(社会科学版)》 2009年6期
2. 相似度: 1% 篇名: 《电视艺术原理》  
来源: 书籍数据 浙江大学出版社 2003-10-01
3. 相似度: 1% 篇名: 《基于Vue.js的后台单页应用管理系统的研究与实现》  
来源: 学术期刊 《现代计算机》 2017年30期
4. 相似度: 1% 篇名: 《博客兴起的传播学意义及其问题研究》  
来源: 学位论文 大连理工大学 2006
5. 相似度: 1% 篇名: 《试论博客引发的文化新现象与美学新意义》  
来源: 学位论文 山东大学 2007

### 互联网相似资源列表:

1. 相似度: 3% 标题: 《单页面应用 - yangyang的专栏 - CS...》  
<https://blog.csdn.net/u014465934/article/details/82948700>
2. 相似度: 1% 标题: 《Vue.js学习系列四 —— Webpack学习...》  
<https://www.jianshu.com/p/aef34acd111f>

- 3.相似度：1% 标题：《vue2.0学习04之vue-cli的解析 - ...》  
<https://www.jianshu.com/p/a68b3a2fe46b>
- 4.相似度：1% 标题：《Vue.js学习系列四——Webpack学习实践...》  
<https://blog.csdn.net/violetjack0808/article/details/54915825>
- 5.相似度：1% 标题：《Vue-MVVM框架理解 - 简书》  
<https://www.jianshu.com/p/9a25bf786aa7>
- 6.相似度：1% 标题：《RESTful 架构详解 菜鸟教程》  
<https://www.runoob.com/w3cnote/restful-architecture.html>
- 7.相似度：1% 标题：《对于restful的简单理解 - ...》  
[https://blog.csdn.net/Poison\\_\\_biting/article/details/77072695](https://blog.csdn.net/Poison__biting/article/details/77072695)
- 8.相似度：1% 标题：《Node.js GET/POST请求 -...》  
<https://justcoding.iteye.com/blog/2311912>
- 9.相似度：1% 标题：《Node.js 第十五章- GET/POST请求...》  
<https://blog.csdn.net/u014085502/article/details/85341709>
- 10.相似度：1% 标题：《npm模块管理器入门 - 李某龙 - 博客园》  
<https://www.cnblogs.com/slly/p/7927760.html>
- 11.相似度：1% 标题：《使用vue-router页面跳转 - sunsh...》  
<https://blog.csdn.net/sunshao904/article/details/78829635>
- 12.相似度：1% 标题：《Node.js模块系统 - CSDN博客》  
[https://blog.csdn.net/little\\_nai/article/details/70197853](https://blog.csdn.net/little_nai/article/details/70197853)
- 13.相似度：1% 标题：《NPM安装使用，以及package.json解释...》  
<https://www.jianshu.com/p/68f2608c7976>
- 14.相似度：1% 标题：《npm的下载、使用，及切换镜像源的三种方式 - ...》  
<https://blog.csdn.net/qq940853667/article/details/70837646>
- 15.相似度：1% 标题：《09\_Node.js GET 与 POST 请求...》  
[https://blog.csdn.net/weixin\\_33720956/article/details/88010444](https://blog.csdn.net/weixin_33720956/article/details/88010444)
- 16.相似度：1% 标题：《Vue页面跳转动画效果的实现方法\_vue.js\_...》  
<https://www.jb51.net/article/147883.htm>

## 全文简明报告:

### 基于vue的个人博客的设计与实现

姓名： 周会艳

院系： 信息工程学院

专业： 计算机科学与技术

导师： 郑颖

完成时间： 2019-00-00

题目

摘要：

互联网成为了当今世界人们对外交流，快速获取、发布和传递信息的最常用也是最便捷的渠道， {45%：互联网在人们的生活，工作和学习的各个方面都发挥着重要作用。} 而博客网站正适应这种人际与技术交流方式的改变，满足个人技术与信息共享的需求。 本文对博客的功能与需求进行了完整分析，设计出了一个简单、易用的个人博客系统。 为了提高开发效率和代码复用率，本系统使用采用SPA（单页面应用）思想的vue.js进行系统的开发。

Vue.js是一个轻量级的基于MVVM模式的渐进式框架。 {48%：组件和数据绑定思想简化了前端开发的复杂性。} 后端将node.js技术与express结合使用，以创建符合RESTful API设计规范的接口。 {54%：实现前端和后端的分离，并且有效地执行去耦。}

关键词 博客，vue.js ， node.js ， 前端框架

英文题目

有道的：

Abstract: the Internet has become the most common and convenient channel for people to communicate with others in the world. The Internet plays an important role in people's life, work and study. Blog sites are adapting to this change in interpersonal and technical communication, to meet the needs of personal technology and information sharing. This article has carried on the complete analysis to the blog function and the demand, has designed a simple, easy to use personal blog system. In order to improve the development efficiency and code reuse rate, This system uses the idea of SPA( single page application) vue. Js to develop system. Vue. js is a lightweight progressive framework based on the MVVM pattern. The idea of component and data binding simplifies the complexity of front-end development. The back end USES node. js technology in conjunction with express to create interfaces that conform to RESTful API design specifications. The front and rear ends are separated and decoupled effectively.

关键词

开发背景8

1.1博客的背景8

1.2系统设计的意义9

系统相关技术介绍9

SPA (Single Page App) 9

单页应用9

与传统网页的比较9

单页应用的好与坏11

NPM11

2.1vue.js12

2.1.1介绍12

Vue-cli12

2.2.1声明式渲染13

2.2.1创建实例14

2.2.2 vue响应式原理16

2.2.5 与其他框架相比优点17

2.2路由17

2.3状态管理17

axios : 19

2.4node.js19

2.4.1Node.js模块系统20

Node.js Express 框架20

2.4Node.js GET/POST请求20

2.3Node.js RESTful API20

2.4Node.js 连接 MySQL20

系统分析20

3.1需求分析20

3.23.2 可行性分析21

系统设计21

前台21

后台21

数据库设计21

概念结构设计21

逻辑结构设计21

系统实现22

序言

博客（blog）作为个人知识管理工具，越来越受人们日常生活的欢迎

{83%：互联网已成为人们快速获取，发布和传输信息的重要渠道。} {54%：通用博客技

术的出现使每个人都可以零成本和零维护创建自己的在线媒体。} Blog技术缔造了“博客”文化。一般博客是当今网络的热点。与朋友或其他观众分享，博客是托管他们的平台。因此，开发了一种有效的通用博客系统，使越来越多的人能够更方便快捷地向他人展示自己。

{61%：博客（Blogger）的出现使我们第一次在互联网世界中拥有知识积累和文化取向。} 让人类从广泛的数字化中生存，并过渡到个性化，准确的目录生存。{51%：博客结合了工作，生活和学习，并通过博客文章记录和发布日常思想。} 提取并链接世界上最有价值，最相关和最有趣的信息和资源。在其形成的旗帜下，更多的知识工作者可以从零距离和零障碍中学习这些最生动的想法。{79%：在博客传播出现之后，标志着以“信息共享”为特征的第一代门户网站，以“理想共享”为特征的第二代门户网站正在兴起。}{45%：互联网已经开始真正揭示知识的无限价值。} 博客通信是网络通信领域中一种鲜明的通信现象。{42%：博客的出现也打破了传统媒体对媒体内容的长期控制。}{46%：Lu Yin是美国社会心理学家的四位先驱之一，也是四位传播先驱之一，受到1947年首次提出的“守门人”理论的强烈影响。} 以至于更有甚者宣称，“守门人”时代结束。{44%：从某种意义上说，理论的建立是实践结论的升华，而实践又必须指导实践活动。} 理论的终结对现实生活具有重要意义。 博客技术开发的博主，博主和博主是否会重新定义互联网的界限.....

{45%：博客的特点是自由，开放和分享，它们通过图形和音频表达来代表个人网络的五大功能。} 一种社交个人服务模型，提供阅读和写作，组织沟通和评估交流。它不是纯粹的技术创新，但它确实是Web应用程序的逐步演变，是一种变化形式。从某种意义上说，它也是一种开放的计算机源代码，它就像一个傻瓜相机，让所有用户都成为程序员。这大大减少了网络与人之间的距离，也消除了人与人之间的距离。博客的普遍性使其沟通方式成为每个人对每个人的传播。但是，博客并不像它们那样繁荣。{47%：本文以非独立博客网站为研究对象，从发送者，接收者，内容，博客服务平台和互动反馈等几个方面对博客进行分析。} 并从中得出了以下结论：{73%：博客参与者的盲目性导致了博客行为的混乱和无聊;} {58%：由于入门门槛为零，缺乏监管，互联网现有模式的博客已经完全被颠覆，变成了一个新的信息垃圾场。}{62%：博客遭受庸俗肤浅，网络侵权等因素的影响;} {52%：博客平台忽视了服务，管理和价值取向方面的用户体验;} 博客之间的互动也呈现出两种极端尴尬的局面。一般来说，我们的博客处于一个混乱的时期，隐藏在虚拟热量的表面下是各种危机。

## 开发背景

### 1.1博客的背景

Blog是继Email，BBS和ICQ之后的第四种网络通信方式。它是互联网时代的个人“读者文摘”。更代表着新的学习方式。{58%：简而言之，博客是以网络为载体，快速，轻松地发布您自己的想法，并及时有效地与他人沟通。} 一个集彩色和个性化显示于一体的综合平台。

博客的分类有很多。比如，按功能分：基本博客、微型博客；按用户分：个人博客、企业博客；按存在方式分：{47%：托管博客，Bloggers建立在独立网站，附属博客和独立博客上。} 本系统就是按用户划分的个人博客。

博客有以下特点：

简单的操作是博客开发的动力。这是博客从许多网民那里收到的最大特色。许多博客托管网站使用的口号就是“在一分钟内轻松拥有一个博客”。

{67%：不断更新的功能是博客生活的催化剂。}{57%：在现代社会中，信息传递速度非

常快，更新博客可以与生物的新陈代谢进行比较。}

3、开放互动功能是博客传播的推动。 {49%：互联网为博客提供了开放性，博客不再是纯粹的私人空间。} 访问者和博主对我们的文章撰写评论和评论，如果我们回复并通过链接地址返回，则可以实现交互效果。 {66%：因此，利用博客开放互动的特点，可以用于沟通推广，形成博客朋友的固定圈子。}

{64%：4，展示个性的特点是博客激动人心的动力。} {47%：博主可以从日志内容，博客界面，文章数量和日志分类中简单地反映博主的个性。}

## 1.2系统设计的意义

在科技快速进步的今天，互联网成为了当今世界人们对外交流，快速获取、发布和传递信息的最常用也是最便捷的渠道， {45%：互联网在人们的生活，工作和学习的各个方面都发挥着重要作用。} 当今时代，技术日新月异，作为互联网技术从业者，需要不断的汲取新知识，走在技术的前沿， 但是大部分新技术文档与书籍都是英文，对于英语能力有限的互联网技术从业者，不可避免的需要借助搜索引擎或者其他人的帮助， 在搜索引擎的搜索结果中大部分的技术问题解决方案的贡献者都来自博客， 所以此次毕业设计自己也想要做一个记录自己学习与工作过程的心得体会的博客网站， 去适应这种人际与技术交流方式的改变，同时也实现个人技术与信息共享的需求。

### 系统相关技术介绍

SPA (Single Page App)

#### 单页应用

单页应用（英语：单页面应用程序，缩写为SPA）是通过动态重写当前页面与用户交互的Web应用程序或网站的模型。 {85%：而不是传统上从服务器重新加载整个新页面。} {85%：这种方法避免了在页面之间切换以中断用户体验，使应用程序更像桌面应用程序。} 在单页面应用程序中，通过加载单个页面来检索所有必需的代码（HTML，JavaScript和CSS）。或者根据需要动态加载适当的资源并通过路由程序将它们添加到页面中（通常是响应用户操作）。 网站所有的页面内容都包含在这个主页面中。 但是在实际开发的时候，还是会分开写（页面片段）。 SPA支持丰富的客户端功能，很少需要重新加载整个页面，因此当用户执行操作或在应用程序的各个区域之间导航时，无需重新加载页面。 因此，它加载速度更快，在后台提取数据，并对单个用户操作响应更快。

#### 与传统网页的比较

为了更好的理解什么是单页应用，我们先来了解传统的网页应用。 传统网页应用的工作模式如下图

/

#### 传统网页应用的工作模式

目前大多数网站都使用此表单。 {55%：单个HTML页面用作功能组件，这些HTML页面以刷新，超链接和表单提交的组合排列，以便为用户提供服务。} 此种方式作为网页应用的传统形式长久不衰，很多流行的开发框架都以之作为范式设计的。 比如 Ruby on Rails, Spring MVC, Express 等等



/

## 传统网页应用的工作分布

{61%：在传统的Web应用程序中，浏览器更像是表示层，路由进程，服务调用和页面跳转过程都由服务器处理。} {77%：也就是说，MVC被放置在服务器端，并且V作为用户界面通过网络被发送到浏览器以作为UI与用户交互。}

这样的范式有以下特点：

{58%：在服务器端，由于 MVC存在于服务器上，因此开发资源的重点和此类应用程序的开发偏向于后端，} {46%：并且后端工程师通常会领导整个项目的开发；}

该页面经常刷新。 {82%：这可能导致资源浪费，用户需要花费额外的时间等待页面刷新，并且用户体验不佳。}

{47%：与传统的Web应用程序相比，单页面应用程序将MVC放在浏览器端：}

/

## 单页应用的工作分布

{69%：1个控制器前端，单页面应用程序将路由处理放在浏览器端，即直接响应浏览器端浏览器地址的变化。} {73%：分发到相应的路由并向用户呈现相应的接口。}

{64%：2，小组件作为功能组件，类似于传统网页中的 Ajax组件，单组件应用程序，小组件作为功能组件，} {64%：当路径更改时，不再刷新整个页面，但组合小组件以替换更改的部件。}

{58%：3数据层预定位，与Ajax组件的显著区别在于单页面应用程序通常在浏览器端有一层真实数据。} 而服务端则退化成了完全的数据 API。 {58%：浏览器端的数据层封装了要调用的上层视图层的服务器API。}

## 单页应用的好与坏

{63%：每种技术都有其优点和缺点，单页应用程序也是如此。} 单页面开发的优点：

1良好的用户体验： 用户无需刷新页面，减少了TTFB请求所需的时间，并且还通过Ajax异步获取数据，页面显示流畅。

2前后端分离： 前端负责界面显示，后端负责数据存储和计算，每个都执行其职责，不混合前端和后端的逻辑。

3减轻服务端压力： 为了减轻服务器的压力，服务器只需要提供一个API接口，这可以在没有页面逻辑和页面拼接的情况下将吞吐量提高几倍。

4共享一组后端程序代码，适应多个目的： 同一组后端程序代码可以应用于Web，移动电话和平板电脑而无需修改。

5组件共享： {61%：一些场景，性能体验并不苛刻，或者产品处于快速试验和错误阶段，采用了一些技术（ Hybrid，} {68%： React Native），组件可以在多个端共享，有

助于快速迭代产品并节省资源。}

单页面开发的缺点：

1首屏加载过慢： 当第一次加载单个页面时，需要组合所有页面所依赖的css和js并以统一的方式加载它们。

2 SEO： 由于页面数据是前端异步加载的一种方式，因此不利于搜索引擎的爬行。

如果应用程序要求包含超出典型HTML表单提供的功能的丰富功能集，则应选择SPA样式应用程序。

NPM

由于使用到的技术都是通过npm来安装的，所以先来介绍npm到底是什么：

NPM 即 Node 包管理器（Node Package Manager）。它是一个用JavaScript编写的包管理系统。 {71%：可以解决NodeJS代码部署中的很多问题，常见的使用场景如下：}

{73%：允许用户从NPM服务器下载其他人编写的第三方软件包以供本地使用。}

{91%：允许用户从NPM服务器下载并安装某人编写的命令程序以供本地使用。}

{80%：允许用户将自己的包或命令程序上载到NPM服务器以供其他人使用。}

使用npm安装包的命令是： `npm install 包名 [--save-dev]`

--save与-dev都是可选的

--save： 将安装的包放在依赖项dependencies中

--save-dev： 安装的包将出现在devDependencies中

2.1vue.js

2.1.1介绍

{83%：Vue是用于构建用户界面的渐进式框架。} 与其他大型框架不同，Vue设计为自下而上应用。 {70%：Vue的核心库仅关注视图层，使其易于入门并易于与第三方库或现有项目集成。} {51%：另一方面，当与现代工具链和各种支持库结合使用时，Vue还完全能够驱动复杂的单页应用程序。}

/

Vue.js中View与Model的交互原理

此图不仅描述了MVVM模式（Model-View\_ViewModel），还描述了在Vue.js中ViewModel是如何和View以及Model进行交互的。

ViewModel是Vue.js的核心，它是一个Vue实例。 Vue实例应用于HTML元素。



当创建了ViewModel后，双向绑定是如何达成的呢？

{69%：首先，我们将上图中的DOM Listener和Data Bindings视为两个工具，这是实现双向绑定的关键。}

{81%：查看View端，ViewModel中的DOM Listeners工具将帮助我们监控页面上DOM元素的更改，如果有更改，则更改Model中的数据。}

{80%：从Model侧，当我们更新Model中的数据时，Data Bindings工具将帮助我们更新页面中的DOM元素。}

### 2.2.1 组件化应用构建

{88%：组件系统是Vue的另一个重要概念，因为它是一种抽象，允许我们使用小型，独立且通常可重用的组件来构建大型应用程序。} {40%：几乎任何类型的应用程序接口都可以在Web页面构造中抽象到组件树中：}

/

页面结构的树形示意

### 2.2.2 vue响应式原理

Vue最独特的功能之一是其非侵入式响应系统。数据模型仅仅是普通的JavaScript对象。而当你修改它们时，视图会进行更新。这使得状态管理非常简单直接。Vue响应式系统的底层的细节如下图：

/

vue响应式原理

vue如何追踪变化呢？将普通JavaScript对象传递给Vue实例的数据选项时，Vue将迭代此对象的所有属性，并使用Object.defineProperty将所有这些属性转换为getters / setter。{43%：由于Object.defineProperty是ES5中无法填充的功能，因此Vue不支持IE8及更低版本的浏览器。} 这些getter / setter对用户是不可见的，但内部Vue通过它们跟踪依赖关系，通知在访问和修改属性时的更改。每个组件实例都有一个相应的观察器实例对象，该对象在组件呈现期间将该属性记录为依赖项。稍后，当调用依赖项的setter时，会通知观察者重新计算，从而更新其关联的组件。这就是vue响应式更新组件的原理。

### 2.2.5 与其他框架相比优点

1.易用。只要您学习了HTML，CSS，JavaScript，就可以阅读vue开发指南并开始构建应用程序。不像其他前端框架一样有比较高的使用门槛。

2.灵活。一个繁荣的生态系统，可以在库和完整的框架之间进行扩展。

Vue-cli

Vue-cli是vue官方开发的脚手架工具，基于webpack构建，很好地规范了其开发生态，开发者只需要关注业务代码本身，而不必操心复杂的webpack配置，可以说是对开发人员尤其是新手是相当友好的。

Vue CLI是一个基于Vue.js的完整系统，用于快速开发并提供以下功能：

1通过 @vue/cli 搭建交互式的项目脚手架。

2通过 @vue/cli + @vue/cli-service-global 快速开始零配置原型开发。

3一个运行时依赖 (@vue/cli-service)，该依赖：

1.可升级；

2基于 webpack 构建，并带有合理的默认配置；

3可以通过项目内的配置文件进行配置；

4可以通过插件进行扩展。

5丰富的官方插件集，集成了前端生态系统中的最佳工具。

6用于创建和管理Vue.js项目的完全图形用户界面。

上边提到了webpack，那么什么是webpack呢？ 请看以下介绍：

{70%: Webpack是一种前端资源模块化管理和打包工具，可以将许多松散模块打包成依赖关系和规则，以满足生产环境中部署的前端资源。} 还可以将按需加载的模块进行代码分割，等到实际需要的时候再异步加载（如图），而要它自动实现这些功能，你得提前编辑好配置文件。

/

用vue项目来举例：浏览器它是只认识js，不认识vue的。我们编写的代码后缀主要是.vue，而html，js，css甚至图像资源可能都在每个.vue文件中；{93%：由于组件化，这些.vue文件之间存在错综复杂的关系。} {85%：因此项目需要被浏览器识别，我们将使用webpack将它们打包成js文件和相应的资源文件。}

{92%：或者理解，我们以vue项目的形式编写项目逻辑，浏览器以他理解的方式运行项目。}

{90%: Webpack将我们的vue项目想要表达的所有意图传递给浏览器以让浏览器运行。}

WebPack可以看做是模块打包机：它的作用是分析你的项目结构，找到JavaScript模块和浏览器无法直接运行的其他扩展语言（更少，TypeScript等）并将其打包成适合浏览器使用的格式。

## 2.2路由

本系统的前端路由使用vue-router来实现。

{65%: Vue-router是一个前端路由管理器，是Vue.js的官方路由插件。} Vue的单页应用程序基于路由和组件。 {84%：传统的页面应用程序使用一些超链接来实现页面切换和跳转。} {83%：在vue-router单页面应用程序中，它是路径之间的切换，即组件的切换。}

## 2.3状态管理

Vuex是一个专为Vue.js设计的状态管理库，可利用Vue.js的细粒度数据响应机制实现高效

的状态更新。 本系统使用vuex来进行系统组件之前公共数据的管理。

Vuex是一种集中的状态管理模型。 {46%：在模块化开发过程中，我们使用组件作为模块单元。} 这确保了我们的模块之间的变量函数名称不会发生冲突，但有时我们需要在组件之间共享一些数据或状态，通常使用参数。 但是传参的做法至少有两个弊端，一个是麻烦（特别是当需要传递很多参数时），其次，管理和冗余并不容易（将参数传递给多个组件需要多个参数列表。而且容易出错）。 vuex提供的集中管理通过集中要共享的数据或状态来解决此问题。 {43%：其他组件根据需要访问更改，大大提高了可维护性和开发效率。}

/

## vuex状态管理流程

请注意，所有商店中对状态的更改都放在商店自己要管理的操作中。 {83%：这种集中的状态管理使得更容易理解将发生哪种类型的突变以及它们是如何被触发的。} {41%：当错误发生时，我们现在还将记录错误之前发生的事件。}

axios :

2.4node.js

{54%：Node.js是一个事件驱动的I / O开源，跨平台的JavaScript运行时环境，在服务器端运行JavaScript。} Node.js由Node.js Foundation [4]持有并维护，并与Linux Foundation建立了合作伙伴关系[5]。 Node.js使用Google的V8运行时代码来提高性能并优化应用程序吞吐量，并使用事件驱动，非阻塞和异步输入和输出模型等技术进行扩展。 {51%：这些技术通常用于数据密集型即时应用程序。}

Node.js的大多数基本模块都是用JavaScript编写的。 {44%：在Node.js出现之前，JavaScript经常被用作客户端编程语言，用JavaScript编写的程序通常在用户的浏览器上运行。} Node.js的出现使JavaScript也能用于服务端编程。 {41%：Node.js包含一组内置模块，允许程序作为Apache HTTP Server或IIS的独立服务器运行。}

## 2.4.1Node.js模块系统

<http://www.runoob.com/nodejs/nodejs-module-system.html>

### 什么是模块系统

### 什么是模块？

{81%：模块是Node.js应用程序的基础部分，文件和模块是一对一的。} {78%：换句话说，Node.js文件是一个模块，可以是JavaScript代码，JSON或编译的C / C ++扩展。} 每个模块都有自己的范围。

模块是构建应用程序的基础，它们还使函数和变量私有化，而不是直接暴露。 JavaScript起初并没有内置的模块系统，CommonJS社区为了使JavaScript可以提供一个类似Python、Ruby等的标准库，自己实现了一套API填补了JavaScript没有内置模块的空白。

CommonJS规范本身涵盖了模块、二进制、Buffer、文件系统、包管理等内容，而NodeJS正是借鉴了CommonJS规范的模块系统，自身实现了一套非常易用的模块系统。 CommonJS对模块的定义可分为三部分：模块引用（require）、模块定义

(exports、module)、模块标识。

{75%：为了相互调用Node.js文件，Node.js提供了一个简单的模块系统。} 模块可以自由隐藏内部实现并通过module.exports公开接口。 我们只需要使用require来实现模块加载。

在Node.js 中创建模块

```
// add.js

function add(a, b) {

  return a+b}

module.exports={

  add}
```

当我们需要在其他地方使用 add 方法时，比如：

```
// app.jsconst {add}=require('./add' )

console.log(add(1, 2))// 3
```

我们只需要调用require（ './add' ）来实现模块。

Node.js Express 框架

Express是一个简单而灵活的node.js Web应用程序框架，它提供了一系列强大的功能，可帮助您创建各种Web应用程序以及丰富的HTTP工具。 使用Express快速构建功能齐全的网站。 Express 框架核心特性如下：

1可以设置中间件来响应 HTTP 请求。

图2定义了用于执行不同HTTP请求动作的路由表。

3您可以通过将参数传递给模板来动态呈现HTML页面。

Connect框架提供了4个丰富的HTTP工具和中间件，可以快速轻松地创建健壮，友好的API

5 性能更好。 Express 在node.js 上扩展了Web应用所需的功能

安装 Express 并将其保存到依赖列表中：

```
$ npm install express --save
```

2.4Node.js GET/POST请求

http: //www.runoob.com/nodejs/node-js-get-post.html

{64%：在许多情况下，我们的服务器需要处理用户的浏览器，例如表单提交。} 表单提交到服务器通常使用GET / POST。

Node.js 获取GET请求内容： {89%：由于GET请求直接嵌入到路径中，因此URL是完整的请求路径，包括？} {72%：后一部分，因此您可以手动将以下内容解析为GET请求的参数。}  
node.js 中 url 模块中的 parse 函数提供了这个功能。 解析函数的作用是解析url并以json格式返回数组。

/

获取get参数

获取 POST 请求内容： POST请求的内容都在请求正文中。 {83%：例如，上传文件，很多时候我们可能不需要注意请求体的内容，恶意的POST请求会大大消耗服务器的资源。}  
} 所以节点。

/

获取post请求

2.3Node.js RESTful API

REST全称是Representational State Transfer，中文意思是表述（编者注：通常译为表征）性状态转移。 {82%：它首次出现在2000年Roy Fielding的博士论文中，而Roy Fielding是HTTP规范的主要作者之一。} 他在论文中提到：“我写作的目的是理解和评估基于网络的应用程序的架构设计，符合架构原则。 获得功能强大，良好且适合通信的结构。 REST指的是一组架构约束条件和原则。 如果架构符合REST约束和原则，我们称之为RESTful架构。

{63%：REST本身不会创建新技术，组件或服务，RESTful背后的想法是使用Web的现有特性和功能。} {100%：更好地使用现有Web标准中的一些准则和约束。} {68%：尽管REST本身深受Web技术的影响，但理论上REST架构风格并未绑定到HTTP，但HTTP是当前唯一与REST相关的实例。} 所以我们这里描述的REST也是通过HTTP实现的REST。

2.4Node.js 连接 MySQL

连接数据库代码如图所示，其中connect()方法用来创建连接，end()方法用来关闭连接，query()方法用来向mysql传递参数。

/

连接mysql数据库

系统分析

3.1需求分析

想要完成一个系统，需求分析是第一步，明确系统是什么，要干什么，完成那些功能。

本系统需求如下：

博客的游客用户可以在网站上对文章进行常规访问。 以及在通过注册登录后，可以进行文章的发布与发表评论。 {54%：博主可以在后台管理用户并添加，删除和修改文章和文章。}

针对博客系统的以上需求，总结出如下信息：

{73%：用户分为游客，普通用户和超级管理用户。}

2.超级管理用户可以管理和设置用户的权限。

3.博客的超级管理员涉及对博客的文章类型管理、文章管理、评论管理、和用户管理。

{41%：4.用户可以阅读文章和发表评论，访问者只能阅读文章。}

5.文章类型可以包含多篇文章，文章可以有多个评论。

### 3.23.2 可行性分析

{50%：可行性分析( Feasibility Analysis)是在系统调查的基础上，无论是否有必要开发新系统，从技术，} 经济和社会角度分析和研究新系统的开发，避免投资错误并确保新系统的开发成功。 {60%：可行性研究的目的是确定问题是否可以在最短的时间内以最小的努力解决。} {44%：为了确定开发的可行性，主要分析了系统的以下几个方面。}

技术可行性分析： 本人对于VUE.JS与NODE.JS在实习中有所接触，对这方面有所了解，所以从技术方面看做个人博客是可行的。

时间可行性： 现在博客技术已经非常成熟，多方面的问题都有成熟的解决方案，结合技术可行性，在预定的时间里可以完成此次设计。

### 系统设计

/

个人博客的总体规划图如下：

/

#### 前台

在前台，游客浏览首页，文章详情页，文章列表与文章分类，注册用户在此基础上可以对文章进行评论。

/

#### 后台

{50%：超级管理员通过预设帐户登录后台。} {51%：超级管理员可以在后台添加，删除和修改文章，添加，删除和修改文章，以及查看和删除评论。} 管理注册用户，查看和禁止可执行的操作。

/

### 数据库设计

数据库是个人博客设计的主要部分。



## 概念结构设计

/

对于属性比较多的实体，在E\_R图中只列出了部分属性以做示意，具体的详细设计将体现在数据库表的结构中。 如下图所以：

### 数据库概念结构e-r图

## 逻辑结构设计

/

基于数据的概念结构设计，设计了下表结构。

(1) 用户表管理用户的信息。 表结构设计如下：

### 字段名称

### 数据类型

### 字段长度

### 说明

id

int

5

### 用户ID(主键)

account

varchar

50

### 登录名

password

varchar

50

### 密码

pwd\_salt

int

5

密码加盐随机数

nickname

varchar

50

用户名字

birthday

date

20

出生日期

gender

tinyint

1

性别

introduce

varchar

200

个人介绍

status

tinyint

1

用户状态

role\_id

int

5

角色id

角色表

字段名称

数据类型

字段长度

说明

id

int

5

角色ID(主键)

name

varchar

20

角色名称

dest

varchar

20

角色描述

permission\_list

varchar

255

权限列表

文章表

字段名称

数据类型

字段长度

说明

id

int

5

文章ID(主键)

type

int

20

文章类型

title

varchar

20

文章标题

content

mediumtext

255

文章内容

author\_id

int

5

文章作者id

status

tinyint

1

文章状态

views

int

20

文章浏览量

Comments

int

20

文章评论数

createtime

int

20

文章创建时间

updatetime

int

20

文章更新时间

dest

varchar

100

对文章的简要描述

文章分类表

字段名称

数据类型

字段长度

说明

id

int

5

角色ID(主键)

name

varchar

20

## 角色名称

dest

varchar

20

## 角色描述

permission\_list

varchar

255

## 权限列表

各个表之间的关系如下图：

## 数据库模型

## 接口

http: //www.shirdon.com/? p=426 参考网址

## 系统实现

## 文献：

- [1]李宇.前后端分离框架在软件设计中的应用[J]. 无线互联科技.2018, 15(17): 41-42.
- [2]汪彤.基于Node. js的图书共享平台的设计与实现[D]. 北京邮电大学.2018.
- [3]梁灏. Vue. js实战[M]. 清华大学出版社.2017.
- [4]麦冬.轻量级响应式框架Vue. js应用分析[J]. 信息与电脑(理论版).2017, (7): 58-59.
- [5]刘红卫.利用Node. js开发前后端分离的系统——以图书馆地方文献系统为例[J]. 天津科技.2018, 45(7): 67-70.
- [6]茆玉庭.基于Node. js和WebSocket的即时通信系统的设计与实现[D]. 南京邮电大学.2018.
- [7]程桂花, 沈炜, 何松林, 张珂杰. 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.
- [8]王伶俐, 张传国.基于NodeJS+Express框架的轻应用定制平台的设计与实现[J]. 计算机科学. 2017, 44(z2): 596-599.
- [9]聂鑫.前端编程与数据库设计的合理运用[J]. 信息与电脑(理论版).2011, (2): 100.
- [10]陈帅, 关玉蓉.基于Java Web的奖助学金系统设计与实现[J]. 科技广场.2017, (3):



190-192.

[11]李玉.Vue框架的前端交互性能优化解决方案的研究[D]. 华中科技大学.2017

[12]邹竞莹.Node. JS博客系统的设计与实现[D]. 黑龙江大学.2016.

[13]旷志光, 纪婷婷, 吴小丽.基于Vue. js的后台单页应用管理系统的研究与实现[J]. 现代计算机.2017, (30): 51-55.

[14]邓雯婷.基于Vue. js构建单页面GIS应用的方法研究[J]. 科技创新与应用.2018, ( 14 ) : 5-7, 10.

[15]朱二华.基于Vue. js的Web前端应用研究[J]. 科技与创新.2017, (20): 119-121.

[16]王志任.基于Vue. js的开发平台的设计与实现[D]. 广东工业大学.2018.

检测报告由PaperPass文献相似度检测系统生成

Copyright 2007-2019 PaperPass