

PaperPass旗舰版检测报告

简明打印版

比对结果(相似度):

总体 : 52% (总体相似度是指本地库、互联网的综合对比结果)
本地库 : 21% (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 : 15% (期刊库相似度是指论文与学术期刊库的对比结果)
学位库 : 14% (学位库相似度是指论文与学位论文库的对比结果)
会议库 : 4% (会议库相似度是指论文与会议论文库的对比结果)
图书库 : 6% (图书库相似度是指论文与图书库的对比结果)
互联网 : 50% (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 : 5CB2221656C7BGP0W

检测版本 : 旗舰版

论文题目 : 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 : 13782字符(不计空格)

段落个数 : 395

句子个数 : 617 句

提交时间 : 2019-4-14 1:53:26

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 : 4% 篇名 : 《透过博客传播方式看把关人理论》
来源 : 学位论文 东北师范大学 2006
- 2.相似度 : 2% 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
- 3.相似度 : 2% 篇名 : 《博客数据的属性拓扑分析》
来源 : 学术期刊 《燕山大学学报》 2015年1期
- 4.相似度 : 2% 篇名 : 《中国官员博客的传播学解读》
来源 : 学位论文 山东大学 2009
- 5.相似度 : 2% 篇名 : 《“博客”与博客传播新探》
来源 : 学位论文 华中科技大学 2004
- 6.相似度 : 2% 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
- 7.相似度 : 2% 篇名 : 《电视艺术原理》
来源 : 书籍数据 浙江大学出版社 2003-10-01
- 8.相似度 : 1% 篇名 : 《高校辅导员如何利用博客开展大学生思想政治教育工作》
来源 : 学术期刊 《才智》 2014年2期
- 9.相似度 : 1% 篇名 : 《网络文化探究》
来源 : 书籍数据 五洲传播出版社 2005-6-30

- 10.相似度：1% 篇名：《黑客闪客之后,博客崛起》
来源：学术期刊《管理与财富》2002年12期
- 11.相似度：1% 篇名：《网络博客的主流化研究》
来源：学位论文 四川大学 2009
- 12.相似度：1% 篇名：《基于vue.js的共享空间平台的设计》
来源：学术期刊《数字通信世界》2018年3期
- 13.相似度：1% 篇名：《今天你"博客"了没有?》
来源：学术期刊《人力资源》2003年3期
- 14.相似度：1% 篇名：《基于Blog的高校图书馆个性化服务》
来源：学术期刊《科技情报开发与经济》2008年17期
- 15.相似度：1% 篇名：《大学生博客现象初探》
来源：学术期刊《科技信息(学术版)》2007年24期
- 16.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
- 17.相似度：1% 篇名：《基于MVVM框架VueJS的汽车销售管理系统的设计与实现》
来源：学位论文 广东工业大学 2017
- 18.相似度：1% 篇名：《博客对高等职业教育的辅助功能探析》
来源：学术期刊《中国新技术新产品》2011年4期
- 19.相似度：1% 篇名：《对档案学界博客现象的观察与思考》
来源：学术期刊《湖北档案》2010年6期
- 20.相似度：1% 篇名：《Internet网络博客的现状与发展》
来源：学术期刊《东方企业文化》2011年10期
- 21.相似度：1% 篇名：《基于开源的轻量级WebGIS开发框架的研究与实现》
来源：学术期刊《测绘与空间地理信息》2015年5期
- 22.相似度：1% 篇名：《我国博客编辑的职能研究》
来源：学位论文 河南大学 2008
- 23.相似度：1% 篇名：《基于路由模式的前端框架设计与改进》
来源：学术期刊《电脑知识与技术》2018年12期
- 24.相似度：1% 篇名：《大度也是一种美德》
来源：学术期刊《教育教学论坛》2013年6期
- 25.相似度：1% 篇名：《论网络文学创作的自由性》
来源：学位论文 山东大学 2007

互联网相似资源列表：

- 1.相似度：5% 标题：《单页面应用 - yangyang的专栏 - CS...》
<https://blog.csdn.net/u014465934/article/details/82948700>
- 2.相似度：4% 标题：《透过博客传播方式看把关人理论论文》
<http://www.doc88.com/p-7107707130685.html>
- 3.相似度：4% 标题：《透过博客传播方式看把关人理论八》
<http://www.doc88.com/p-3847776407398.html>
- 4.相似度：4% 标题：《Node.js快速入门 - 程裕强的专栏 - C...》
<https://blog.csdn.net/chengyuqiang/article/details/56487806>
- 5.相似度：4% 标题：《我对博客的理解。_森森_新浪博客》
<http://blog.sina.com.cn/u/b4be591a0102vuwo>
- 6.相似度：4% 标题：《透过博客传播方式看把关人理论》
<http://d.wanfangdata.com.cn/Thesis/Y888287>
- 7.相似度：4% 标题：《简析透过博客传播方式看把关人理论.pdf 全文-...》
<https://max.book118.com/html/2015/0113/11338053.shtm>
- 8.相似度：3% 标题：《透过博客传播方式看把关人理论》
<http://m.doc88.com/p-73542830292.html>
- 9.相似度：3% 标题：《个人博客系统的设计与实现(1)-资源下载_天...》
<http://www.wenku365.com/d-7282822.html>
- 10.相似度：3% 标题：《个人博客系统的设计与实现(1) 南京廖华》
http://www.wodefanwen.com/lhd_86cdz1x1zt371qy5cnk3_2.html
- 11.相似度：3% 标题：《个人博客系统的设计与实现.doc》

- <https://max.book118.com/html/2016/1118/63818565.shtm>
- 12.相似度：3% 标题：《个人博客系统的设计与实现 (1)》
<https://www.docin.com/p-1501883385.html>
- 13.相似度：2% 标题：《博客文化中的自我认同与价值审视》
<http://www.doc88.com/p-9562932246585.html>
- 14.相似度：2% 标题：《博客数据的属性拓扑分析》
<http://m.doc88.com/p-7778261194096.html>
- 15.相似度：2% 标题：《restful 架构详解 - 姜小妖 - 博客园...》
<https://www.cnblogs.com/jiangxiaoyaoblog/p/5826582.html>
- 16.相似度：2% 标题：《Restful API 架构与设计参考原则 - ...》
<https://blog.csdn.net/zwqjoy/article/details/78788915>
- 17.相似度：2% 标题：《RESTful 架构详解 菜鸟教程》
<https://www.runoob.com/w3cnote/restful-architecture.html>
- 18.相似度：2% 标题：《RESTFUL接口 - hicjiajia - ...》
<https://www.cnblogs.com/hicjiajia/p/4658790.html>
- 19.相似度：2% 标题：《REST面向资源架构 RESTful架构详解 -...》
<https://blog.csdn.net/h330531987/article/details/72724111>
- 20.相似度：2% 标题：《Node.js GET/POST请求 菜鸟教...》
<http://www.runoob.com/nodejs/node-js-get-post.html>
- 21.相似度：2% 标题：《Node.js开发Web后台服务 - 张果 - ...》
<https://www.cnblogs.com/best/p/6204116.html>
- 22.相似度：2% 标题：《Node.js开发Web后台服务 - 学习笔记 ...》
<https://blog.csdn.net/zhangguo5/article/details/62423469>
- 23.相似度：2% 标题：《前端面试题（框架部分） - 前端学堂 - 博客园...》
<https://www.cnblogs.com/wu-web/p/8048029.html>
- 24.相似度：2% 标题：《Vue系列（1）：单页面应用程序 - ahill...》
<https://blog.csdn.net/ahill1/article/details/82931161>
- 25.相似度：2% 标题：《如何用Vue构建大型单页面应用？-八戒服务购》
<https://go.zbj.com/news/14138.html>

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， {52%：互联网在人们生活、工作与学习的方方面面都发挥着重要的作用。} 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。 {51%：组件与数据绑定思想，简化了前端开发的复杂度。} 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。 {51%：实现前后端分离，有效的进行了解耦。}

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

{94%: Blog (博客) 作为一种个人知识管理工具, 在人们日常生活中越来越盛行}

{100%：互联网成为人们快速获取、发布和传递信息的重要渠道，它在人们政治、经济、生活等各个方面发挥着重要的作用。} {100%：通用博客技术的出现使得每个人可以零成本、零维护地创建自己的网络媒体，Blog站点所形成的网状结构促成了不同于以往社区的Blog文化，} Blog技术缔造了“博客”文化。 {100%：通用博客是当今网络的热点，越来越多的网民开始将自己的想法、感受、心情等以文字、图片或多媒体的形式发布到网络中，} {100%：以便于与朋友或其他浏览者分享，而博客正是承载这些内容的平台。} {100%：所以开发出一个高效的通用博客系统，使越来越多的人更能方便、快捷的向别人展示自己。}

{100%：博客(Blogger)的出现，使我们在互联网世界，第一次有了知识积累和文化指向。} {100%：使人类由粗放的数字化生存，过渡为个人化的精确的目录式生存。} {100%：博客们将工作、生活和学习融为一体，通过博客日志(Blog)，将日常的思想精华及时记录并发布，} {100%：萃取并链接全球最有价值、最相关、最有意思的信息与资源。} {100%：在其形成的博客媒介下，更多的知识工作者能够零距离、零壁垒地汲取这些最鲜活的思想。} {100%：博客传播的出现，标志着以“信息共享”为特征的第一代门户之后，追求“思想共享”为特征的第二代门户正在浮现，} {100%：互联网开始真正凸现无穷的知识价值。} {100%：博客传播是网络传播领域出现的个性鲜明的传播现象，其出现改变了网络传播的秩序，重新划分了网络传播的界限。} {100%：博客的出现还打破了长久以来传统媒体对媒介所传播内容的控制，} {100%：使得由美国社会心理学家、传播学四大先驱之一的卢因1947年率先提出的“把关人”理论受到强烈冲击，} 以至于更有甚者宣称，“守门人”时代结束。 {100%：一个理论的建立，在某种意义上讲是现实实践的结论升华，它反过来必然要指导实践活动。} {100%：一个理论的结束，对于现实生活是具有重大意义的。} {100%：博客技术下发展起来的博客群体、博客媒介和博客传播，是否将重新定义互联网的界限...}

{100%：博客是以自由、开放和共享为文化特征，通过图文音象等表现形式，围绕个人网络存在的五大功能，} {100%：提供存取读写、组织沟通、评价交换等服务的一种社会化个人服务模式。} {100%：它并不是纯粹的技术创新，但确是一种逐渐演变的网络应用，一种形式的变化。} {100%：从某种意义上说，也是一种计算机源代码开放，它像一部傻瓜照相机一样，让所有使用者都成了程序员，} {100%：这就大大消解了网络与人的距离，也消解了人与人的距离。} {100%：博客的全民性，让它的传播方式成为所有人对所有人的传播。} {100%：然而，博客并不如表面般繁荣，当博客毫不掩饰地在大众面前喧闹的时候，接踵而来的问题使得博客乱了方寸。} {49%：本文以非独立博客网站为研究对象，从传者、受者、内容、博客服务平台以及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： {100%：博客参与者的盲目性导致了博客行为过程中的迷茫与厌倦；} {100%：由于进入的零门槛和缺少监管，彻底颠覆互联网既有模式的博客，变成了新的信息垃圾场。} {96%：博客正遭受低俗肤浅成风、网络侵权等因素的困扰；} {96%：博客平台在服务、管理以及价值导向上忽略了用户体验；} {90%：博客间的互动也呈现两个极端的尴尬局面。} {79%：从总体上说，我国博客正处在一个混乱期，隐藏在虚热表面下的是种种的危机。}

开发背景

1.1 博客的背景

{81%：Blog是继Email、BBS、ICQ之后出现的第四种网络交流方式，至今为止十分受大家的欢迎，} {96%：是网络时代的个人“读者文摘”，是以超级链接为入口的网络日记，是代表着新的生活方式和新的工作方式，} 更代表着新的学习方式。 {100%：简言之，Blog就是以网络作为载体，简易迅速便捷地发布自己的心得，及时有效轻松地与他人进行交流，} {100%：再集丰富多彩的个性化展示于一体的综合性平台。}

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：{92%：托管博客、自建独立网站的Blogger、附属Blogger、独立博客。} 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。{72%：这是博客收到众多网友青睐的最大特点。}
许多博客托管网站使用的口号就是{61%：“一分钟内就可以轻松拥有一个博客”}。

{63%：2、不断更新的功能是博客生活的催化剂。}{83%：现代社会，信息传递超级快速，更新博客可以比作生物的新陈代谢，生物没有新陈代谢就代表着生命的终结，而博客没有更新，博客也就失去了生命力。}

3、开放互动功能是博客传播的推动。{90%：网络赋予了博客的开放性，博客也就不再是一个纯粹的私人空间。}{65%：游客与博主给我们的文章写留言与评论，如果我们对他们进行了回复，并通过链接地址进行回访，就达到了互动效应。}{76%：因此，利用博客开放互动的特点，可以用于交流推广，形成一个固定的博客朋友圈。}

{87%：4、展示个性的特点是博客精彩的原动力。}{77%：博主可以简单地从日志内容，博客界面，文章数量，日志分类体现出博主的个性。}

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，{52%：互联网在人们生活、工作与学习的方方面面都发挥着重要的作用。} 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：{100%：single-page application，缩写 SPA}）是一种网络应用程序或网站的模型，它通过动态重写当前页面来与用户交互，{100%：而非传统的从服务器重新加载整个新页面。}{100%：这种方法避免了页面之间切换打断用户体验，使应用程序更像一个桌面应用程序。}{100%：在单页应用中，所有必要的代码（HTML、JavaScript和 CSS）都通过单个页面的加载而检索，}{80%：或者根据需要（通常是为用户操作）由路由程序动态装载适当的资源并添加到页面。} 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）。{66%：SPA可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的各区域间导航时无需重新加载页面，}{94%：所以加载速度更快，可在后台提取数据，并且对单个用户操作的响应更快。}

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

{100%：目前绝大部分的网站都还采用这种形式。} {97%：将单个 HTML 页面作为功能元件，通过刷新、超链接、表单提交等方式，组合排列这些 HTML 页面，来为用户提供服务。} 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

{100%：在传统的网页应用中，浏览器更多的是充当一个展示层，路由处理、服务调用、页面跳转流程都由服务端来处理。} {100%：即 MVC 都放在服务器端，而 V 作为用户界面则通过网络发送到浏览器端，作为 UI 与用户交互。}

这样的范式有以下特点：

{100%：重服务端，由于 MVC 都存在于服务器上，因此这类应用在开发资源和开发的重心都偏向后端，往往是后端工程师来主导整个项目开发；}

{100%：页面频繁刷新，由于浏览器端只是一个展现层，当页面功能有所变化的时，页面就刷新，} {100%：这会导致资源的浪费，用户需要花费额外的时间等待页面刷新，用户体验不佳。}

{80%：相较于传统网页应用，单页应用将 MVC 前置到了浏览器端：}

/

单页应用的工作分布

{98%：1控制器前置，单页应用将路由处理放在浏览器端，即在浏览器端直接响应浏览器地址的变化，} {100%：分发到对应的路由，向用户呈现对应的界面。}

{98%：2以小块组件为功能元件，类似于传统网页中的 Ajax组件，单页应用以小的组件为功能元件，} {100%：在路由变化时，不再刷新整个页面，而是组合这些小的组件，替换变化的部分。}

{98%：3数据层前置，与 Ajax组件一个明显的区别是，单页应用在浏览器端通常有一层实实在在的数据层，} 而服务端则退化成了完全的数据 API。 {100%：浏览器端的数据层会封装服务端 API，供上层的视图层调用。}

单页应用的好与坏

{100%：每种技术都有其利弊，单页应用也是如此。} 单页面开发的优点：

1良好的用户体验： {94%：用户不需要重新刷新页面，减少TTFB的请求耗时，获取数据

也是通过Ajax异步获取，页面显示流畅。}

2前后端分离： {100%：前端负责界面显示，后端负责数据存储和计算，各司其职，不会把前后端的逻辑混杂在一起。}

3减轻服务端压力： {100%：减轻服务器压力，服务器只需要提供API接口，不用管页面逻辑和页面的拼接，吞吐能力会提高几倍。}

{89%：4共用一套后端程序代码，适配多端：} {100%：同一套后端程序代码，不用修改就可以适用于Web、手机、平板。}

5组件共享： {100%：某些对性能体验要求不高的场景，或者产品处于快速试错阶段，借助于一些技术（Hybrid、React Native），} {100%：可以在多端共享组件，便于产品的快速迭代，节约资源。}

单页面开发的缺点：

1首屏加载过慢： {92%：单页面首次加载，需要将所有页面所依赖的css和js合并后统一加载，所以css和js文件会较大，影响页面首次打开时间。}

2 SEO： {100%：因为页面数据都是前端异步加载的方式，不利于搜索引擎的抓取。}

{96%：如果应用要求包括丰富的功能，且超出了典型HTML窗体提供的功能，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）。 {86%：它是一个以JavaScript编写的软件包管理系统，默认环境为Node.js，从Node.js 0.6.3版本开始，NPM被自动附带在安装包中。} {100%：能解决NodeJS代码部署上的很多问题，常见的使用场景有以下几种：}

{100%：允许用户从NPM服务器下载别人编写的第三方包到本地使用。}

{100%：允许用户从NPM服务器下载并安装别人编写的命令行程序到本地使用。}

{100%：允许用户将自己编写的包或命令行程序上传到NPM服务器供别人使用。}

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

{100%: Vue 是一套用于构建用户界面的渐进式框架。} {100%: 与其它大型框架不同的是, Vue 被设计为可以自底向上逐层应用。} {100%: Vue 的核心库只关注视图层, 不仅易于上手, 还便于与第三方库或既有项目整合。} {100%: 另一方面, 当与现代化的工具链以及各种支持类库结合使用时, Vue 也完全能够为复杂的单页应用提供驱动。}

/

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。 {100%: Vue实例是作用于某一个HTML元素上的, 这个元素可以是HTML的body元素, 也可以是指定了id的某个元素。}

当创建了ViewModel后, 双向绑定是如何达成的呢?

{98%: 首先, 我们将上图中的DOM Listener和Data Bindings看作两个工具, 它们是实现双向绑定的关键。}

{99%: View侧看, ViewModel中的DOM Listeners工具会帮我们监测页面上DOM元素的变化, 如果有变化, 则更改Model中的数据。}

{100%: 从Model侧看, 当我们更新Model中的数据时, Data Bindings工具会帮我们更新页面中的DOM元素。}

2.2.1 组件化应用构建

{100%: 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、独立和通常可复用的组件构建大型应用。} {81%: 在网页构建中几乎任意类型的应用界面都可以抽象为一个组件树:}

/

页面结构的树形示意

2.2.2 vue响应式原理

{100%: Vue 最独特的特性之一, 是其非侵入性的响应式系统。} 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? {99%: 当把一个普通的 JavaScript 对象传给 Vue 实例的 data 选项, Vue 将遍历此对象所有的属性, 并使用 Object.defineProperty 把这些属性全部转为 getter/setter。} {78%: 由于Object.defineProperty 是ES5中一个无法 shim 的特性, 所以 Vue 不支持 IE8 及以下低版本的浏览器。} {91%: 这些 getter/setter 对用户来说是不可见的, 但是在内部 Vue 通过它们追

踪依赖，在属性被访问和修改时通知变化。} {100%：每个组件实例都有相应的 `watcher` 实例对象，它会在组件渲染的过程中把属性记录为依赖，} {100%：之后当依赖项的 `setter` 被调用时，会通知 `watcher` 重新计算，从而致使它关联的组件得以更新。} 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。 {70%：只要已经学会了 `HTML`、`CSS`、`JavaScript` 就可以阅读 `vue` 开发指南开始构建应用。} 不像其他前端框架一样有比较高的使用门槛。

2灵活。 {100%：不断繁荣的生态系统，可以在一个库和一套完整框架之间自如伸缩。}

`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。

{86%： `Vue CLI` 是一个基于 `Vue.js` 进行快速开发的完整系统，提供了以下功能：}

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；

{94%：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。

{96%：5一个丰富的官方插件集合，集成了前端生态中最好的工具。}

{97%：6一套完全图形化的创建和管理 `Vue.js` 项目的用户界面。}

上边提到了 `webpack`，那么什么是 `webpack` 呢？ 请看以下介绍：

{82%：`webpack` 就是前端资源模块化管理和打包工具，它可以将很多松散的模块按照依赖和规则打包成符合生产环境部署的前端资源，} 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。 {100%：而我们写的代码后缀大多是 `.vue` 的，在每个 `.vue` 文件中都可能 `html`、`js`、`css` 甚至是图片资源；} {100%：并

且由于组件化，这些 .vue 文件之间还有错综复杂的关系。} {100%：所以项目要被浏览器识别，我们就要使用 webpack 将它们打包成 js 文件以及相应的资源文件。}

{98%：或者这么理解，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来运行项目。} {100%：webpack 把我们的 vue 项目想表达的所有意图传递给浏览器让浏览器去运行。} WebPack 可以看做是模块打包机： {97%：它做的事情是，分析你的项目结构，找到 JavaScript 模块以及其它的一些浏览器不能直接运行的拓展语言（less，} {97%：TypeScript 等），并将其打包为合适的格式以供浏览器使用。}

2.2 路由

本系统的前端路由使用 vue-router 来实现。

{83%：vue-router 是一个前端路由管理器，是 Vue.js 官方的路由插件，它和 vue.js 是深度集成的，适合用于构建单页面应用。} {100%：vue 的单页面应用是基于路由和组件的，路由用于设定访问路径，并将路径和组件映射起来。} {100%：传统的页面应用，是用一些超链接来实现页面切换和跳转的。} {100%：在 vue-router 单页面应用中，则是路径之间的切换，也就是组件的切换。}

2.3 状态管理

{97%：Vuex 是专门为 Vue.js 设计的状态管理库，以利用 Vue.js 的细粒度数据响应机制来进行高效的状态更新。} 本系统使用 vuex 来进行系统组件之前公共数据的管理。

{100%：vuex 是一种集中式状态管理模式，什么意思呢？} {81%：我们在模块化开发过程中，我们以组件来作为模块单位，模块之间存在于不同的命名空间，作用域互不干预，} {87%：这样保证了我们模块之间变量函数名称等不会冲突，但是有时候我们需要组件之间共享一些数据或者状态，通常用传参来实现，} 但是传参的做法至少有两个弊端， {100%：一是麻烦(尤其是当需要传递的参数很多时)，} {100%：二是不好管理且冗余(给多个组件传参就需要多份参数列表，} 而且容易出错)。 {100%：vuex 提供的集中式管理就解决了这个问题，通过把要共享的数据或状态集中起来管理，} {93%：别的组件需要时就去访问变更，大大提高了可维护性和开发效率。}

/

vuex 状态管理流程

{100%：需要注意，所有 store 中 state 的改变，都放置在 store 自身的 action 中去管理。} {100%：这种集中式状态管理能够被更容易地理解哪种类型的 mutation 将会发生，以及它们是如何被触发。} {100%：当错误出现时，我们现在也会有一个 log 记录 bug 之前发生了什么。}

axios :

2.4 node.js

{84%：Node.js 是一个事件驱动 I/O 的能够在服务器端运行 JavaScript 的开放源代码、跨平台 JavaScript 运行环境。} {80%：Node.js 由 Node.js 基金会持有和维护[4]，并与 Linux 基金会合作关系[5]。} {94%：Node.js 采用 Google 开发的 V8 运行代码，使用事件驱动、非阻塞

和异步输入输出模型等技术来提高性能，可优化应用程序的传输量和规模。} {62%：这些技术通常用于数据密集的即时应用程序。}

{100%：Node.js大部分基本模块都用JavaScript语言编写。} {94%：在Node.js出现之前，JavaScript通常作为客户端程序设计语言使用，以JavaScript写出的程序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。 {87%：Node.js含有一系列内置模块，使得程序可以脱离Apache HTTP Server或IIS，作为独立服务器运行。}

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?

{100%：模块是Node.js 应用程序的基本组成部分，文件和模块是一一对应的。}
{100%：换言之，一个 Node.js 文件就是一个模块，这个文件可能是JavaScript 代码、JSON 或者编译过的C/C++ 扩展。} {94%：每个模块都有自己的作用域，当我们使用var 来声明的一个变量，他并不是全局的，而是属于当前模块下。}

{98%：模块是构建应用程序的基础，也使得函数和变量私有化，不直接对外暴露出来，接下来我们就要介绍Node的模块化系统和它最常用的模式。} 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。

{100%：为了让Node.js的文件可以相互调用，Node.js提供了一个简单的模块系统。}
{95%：模块可以通过 module.exports 自由地隐藏内部实现、对外暴露接口 。}
{96%：我们只需要通过 require，就能实现模块加载引入。}

在Node.js 中创建模块

```
// add.js

functionadd(a, b){

returna+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
```

{71%：我们只需要调用 `require('./add')` 就能实现对模块的引入。}

Node.js Express 框架

{100%：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你创建各种 Web 应用，和丰富的 HTTP 工具。} {100%：使用 Express 可以快速地搭建一个完整功能的网站。} Express 框架核心特性如下：

1可以设置中间件来响应 HTTP 请求。

{96%：2定义了路由表用于执行不同的 HTTP 请求动作。}

{96%：3可以通过向模板传递参数来动态渲染 HTML 页面。}

{92%：4丰富的HTTP工具以及来自Connect框架的中间件随取随用，创建强健、友好的API变得快速又简单}

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
```

{100%：在很多场景中，我们的服务器都需要跟用户的浏览器打交道，如表单提交。}

{92%：表单提交到服务器一般都使用 GET/POST 。

Node.js获取GET请求内容： {100%：由于GET请求直接被嵌入在路径中，URL是完整的请求路径，包括了?} {100%：后面的部分，因此你可以手动解析后面的内容作为GET请求的参数。} node.js 中 url 模块中的 parse 函数提供了这个功能。 {90%：parse 函数的作用是解析url，返回一个json格式的数组}

/

获取get参数

获取 POST 请求内容： {100%：POST 请求的内容全部的都在请求体中，http.ServerRequest 并没有一个属性内容为请求体，原因是等待请求体传输可能是一件耗时的工作。} {100%：比如上传文件，而很多时候我们可能并不需要理会请求体的内容，恶意的 POST请求会大大消耗服务器的资源，} {100%：所以 node.js默认是不会解析请求体的，当你需要的时候，需要手动来做。}

/

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。 {100%: 它首次出现在2000年Roy Fielding的博士论文中, Roy Fielding是HTTP规范的主要编写者之一。} 他在论文中提到: {98%: "我这篇文章的写作目的, 就是想在符合架构原理的前提下, 理解和评估以网络为基础的应用软件的架构设计, } {100%: 得到一个功能强、性能好、适宜通信的架构。} REST指的是一组架构约束条件和原则。 {100%: " 如果一个架构符合REST的约束条件和原则, 我们就称它为RESTful架构。}

{100%: REST本身并没有创造新的技术、组件或服务, 而隐藏在 RESTful背后的理念就是使用 Web的现有特征和能力, } {100%: 更好地使用现有 Web标准中的一些准则和约束。} {100%: 虽然REST本身受Web技术的影响很深, 但是理论上REST架构风格并不是绑定在HTTP上, 只不过目前HTTP是唯一与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。 {46%: 博主可以通过后台对用户进行管理以及文章与文章分类添加、删除、修改。}

针对博客系统的以上需求, 总结出如下信息:

{59%: 1.用户分为游客、普通用户和超级管理用户。}

{52%: 2.超级管理用户员可以用户进行管理和设置权限。}

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、发表评论, 游客只能进行文章的阅读。}

{50%: 5.一种文章类型可以有多篇文章, 一篇文章可以有多条评论。}

3.23.2 可行性分析

可行性分析(Feasibility Analysis)是在系统调查的基础上, {100%: 针对新系统的开发是否具备必要性和可能性, } {100%: 对新系统的开发从技术、经济、社会的方面

进行分析和研究，} {100%：以避免投资失误，保证新系统的开发成功。} {100%：可行性研究的目的是用最小的代价在尽可能短的时间内确定问题是否能够解决。} {97%：为了确定开发具有可行性，对本系统主要进行了以下几个方面的分析。}

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

{53%：超级管理员通过预设的账号登录后台。} {40%：超级管理员可以在后台对文章进行增删查改，对文章分类进行增删查改，对评论进行查看和删除；} {46%：对注册用户进行管理，可以进行的操作有查看和封禁。}

/

数据库设计

{42%：数据库是个人博客设计的主要部分，本系统选用了mysql数据库，使用navicat premium可视化数据库管理工具进行数据库的设计与管理。}

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

/

{52%：根据数据的概念结构设计为基础，设计出如下表结构}

(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