山东大学软件学院

Web技术 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：  如右 | 姓名：  陈智伟  杨茜云  李泽宇 202000300021  李安祺 | | 班级：7班 |
| 实验题目：多人博客平台 | | | |
| 实验学时：32 | | 实验日期：2021年10月1日至12月12日 | |
| 实验目的：  熟练掌握和灵活应用网站的规划设计方面的理论知识和技能，规划和设计两个网站，使设计的网站能够对网页布局有充分的掌控能力，合理的选择颜色、字体、分辨率以及图片、声音、动画等多媒体元素，能最终实现网站的前端设计目标。  熟练掌握和灵活应用后端技术，使用spring boot实现与前端的数据交互，实现对数据库的访问。 | | | |
| 硬件环境： PC/Mac，八核、64GB内存 | | | |
| 软件环境：  操作系统：（Win/Mac）  开发工具：（IDEA、VSCode）  数据服务：（MySql-8）  应用服务：（Tomcat） | | | |
| 输出成果：  202099123456\_张三\_实验1报告.docx  202099123456\_张三\_实验1.zip，内容包括：  /demo\_fe.zip 前端源码（名称不限。不要node\_modules）  /demo\_be.zip 后端源码（不要带out、JDK、Tomcat等）  demo\_be.jar 后端产物（用SpringBoot的话输出成JAR，自带Tomcat。传统架构可输出WAR包）  demo\_fe\_dist.zip 前端产物（也可集成到后端，做成一个包）  init.sql DB初始化脚本（建表结构、初始数据等） | | | |
| 项目说明：  项目客户为软件学院，工期为三周天，计划投入4人21天/8工时。  项目组共有4人，分工如下：  陈智伟：负责前端。  杨茜云：负责前端。  李泽宇：负责一部分功能的后端开发、数据库设计。  李安祺：负责一部分功能的后端开发、数据库设计。 | | | |
| 需求分析 博客是网络交流的一种重要形式。国内拥有许多专业的博客平台，如CSDN等。  为了实现小规模用户之间文章的分享、多人之间的交流互动，熟悉博客平台的开发流程，本团队打算进行一个多人博客项目的开发。 | | | |
| 概要设计 功能简述如下：  大体分为用户模块、文章模块、搜索模块和交互模块。 用户模块 包括基本的用户登录注册功能、用户个人信息的修改维护功能。  登陆后用户的个人主界面可以按时间顺序查看到自己历史发布的博客记录。 文章模块 包括文章的编辑发布、文章的修改、文章分类标签的设计、文章的删除功能  文章分为标题、摘要、标签、主体内容四部分。  使用markdown格式的一个插件，使得用户可以编辑博客主体的内容。  文章发布后，若是文章的拥有者可以随时修改文章的内容，并且可以设置文章是否可以为其他用户所见（即可见性），也可以随时删除文章。 交互模块 包括点赞、收藏、发表评论、查看他人文章的功能。  用户点赞和收藏过的文章会按时间顺序分别放入到用户个人主页下的我的点赞和我的收藏栏目。  所有的用户都可以在文章下面发表评论，并可无限制地对评论进行再评论。显示用户的评论时间和评论的用户名。 搜索模块 用户可在博客大厅里的搜索栏里按照用户或者博客的分类，来搜索用户或者博客。  或者直接使用一旁的标签分类直接搜索对应分类的博客。  搜索功能会将所有包含用户搜索的关键字的用户或者博客搜索出来，供用户查看。 | | | |
| 系统架构 系统采用前后端分离的架构。  前端使用：vue2 + element–ui + axios  后端使用：Spring Boot + mybatis plus + shrio + redis + jwt + hibernate validater    目标用户环境为Google, Fire Fox, Edge浏览器，建议分辨率1080p。 | | | |
| 数据设计用户表m\_user 包括以下字段：id，username，avatar，email，password，status，created，last\_login。  分别为如下含义：id号，用户名，头像，邮箱，口令，状态，注册时间，上次登录时间。  主键为id。  username字段有唯一约束。  created，password，字段有非空约束。 博客表m\_blog 包含以下字段：id，user\_id，title，description，content，created，status，praise，label，favorite  分别为如下含义：博客id号，对应所属用户的id，标题，摘要，内容，创建时间，状态，点赞数，标签，收藏者。  主键为id。  user\_id，title，description，有非空约束。  user\_id为外键，参见用户表id字段。 评论表comments 包含以下字段：id，user\_id，content，blog\_id，follow，time  分别为如下含义：评论id号，评论者(用户)的id号，内容，评论的博客id，回复的哪条评论，评论时间。  主键为id。  content，user\_id，blog\_id，follow，time都不能为空  外键为user\_id，参照用户表id字段。  外键为blog\_id，参照博客表id字段。 收藏表m\_favorite 包括以下字段：    Fid为收藏者id，fblogid为收藏的博客id，ftime为收藏时间  Fid，fblogs，ftime有非空约束 点赞表m\_praise 包括以下字段：    pid为点赞者id，pblogid为点赞的博客id，ptime为点赞时间  Pid，pblogid，ptime有非空约束 关注表m\_subscription 包括以下字段：    Id为关注者id，sid为被关注者id，stime为关注时间  id，sid，stime有非空约束 | | | |
| 功能接口搜博客  1. **根据关键字搜索博客**   功能：输入关键字搜索博客  地址：/blog/selectByKey  传入参数：String key  返回：  搜索成功：    搜索失败：     1. **根据预定义的标签分类搜索博客**   请求地址：  请求参数  {  "label": "java"  }  返回结果  {  "code": 200,  "msg": "获取用户点赞博客成功",  "data": [  {  "id": 9,  "userId": 1,  "title": "真正理解Mysql的四种隔离级别@",  "description": "事务是应用程序中一系列严密的操作，所有操作必须成功完成，否则在每个操作中所作的所有更改都会被撤消。也就是事务具有原子性，一个事务中的一系列的操作要么全部成功，要么一个都不做。\n\n事务的结束有两种，当事务中的所以步骤全部成功执行时，事务提交。如果其中一个步骤失败，将发生回滚操作，撤消撤消之前到事务开始时的所以操作。",  陈智伟 15:40:15  "content": "### 什么是事务 \n\n> 事务是应用程序中一系列严密的操作，所有操作必须成功完成，否则在每个操作中所作的所有更改都会被撤消。也就是事务具有原子性，一个事务中的一系列的操作要么全部成功，要么一个都不做。\n> \n> 事务的结束有两种，当事务中的所以步骤全部成功执行时，事务提交。如果其中一个步骤失败，将发生回滚操作，撤消撤消之前到事务开始时的所以操作。\n\n\*\*事务的 ACID\*\*\n\n事务具有四个特征：原子性（ Atomicity ）、一致性（ Consistency ）、隔离性（ Isolation ）和持续性（ Durability ）。这四个特性简称为 ACID 特性。\n\n> 1 、原子性。事务是数据库的逻辑工作单位，事务中包含的各操作要么都做，要么都不做\n> \n> 2 、一致性。事 务执行的结果必须是使数据库从一个一致性状态变到另一个一致性状态。因此当数据库只包含成功事务提交的结果时，就说数据库处于一致性状态。如果数据库系统 运行中发生故障，有些事务尚未完成就被迫中断，这些未完成事务对数据库所做的修改有一部分已写入物理数据库，这时数据库就处于一种不正确的状态，或者说是 不一致的状态。",  "created": "2021-12-10 23:46:10",  "status": 0,  "praise": 1,  "favorite": 0,  "label":"[\"java\"]"  },  {  "id": 23,  "userId": 2,  "title": "java",  "description": "我真不错啊啊哈哈哈",  "content": "java",  "created": "2021-12-16 01:10:37",  "status": 0,  "praise": 1,  "favorite": 1,  "label": "[\"java\"]"  }  ]  } 获取所有博客内容 地址：/blogs  传入参数：Integer currentpage  返回：   维护个人信息 功能：更新个人信息  地址：/userinfo/renewinfo  传入参数：user    返回：    功能：获取个人信息  地址：/userinfo/getinfo  传入参数：Long id  返回：  获取信息成功：    失败：   博客点赞 功能：返回用户点赞信息  地址：/praise/getpraiseinfo  传入参数：int userid  返回：  {  "code": 200,  "msg": "获取用户id：2博客点赞信息成功",  "data": [  {  "pid": 2,  "pblogid": 3,  "ptime": "2021-12-06"  },  {  "pid": 2,  "pblogid": 9,  "ptime": "2021-12-06"  },  ]  }  功能：博客点赞信息添加（点赞）  地址：/praise/addpraiseinfo  传入参数：Praise praise  返回：“添加博客点赞信息成功”  功能：博客点赞信息删除（取消点赞）  地址：/praise/delpraiseinfo  传入参数：Praise praise  返回：“删除博客点赞信息成功”  功能：点赞数添加  地址：/blog/addpraisenum  传入参数：Blog blog  返回“添加点赞数成功”  功能：点赞数减少  地址：/blog/addpraisenum  传入参数：Blog blog  返回“减少点赞数成功”  功能：获取用户所有点赞的博客  地址：/blog/getuserpraiseblog  传入参数：int userid  返回：  {  "code": 200,  "msg": "获取用户点赞博客成功",  "data": [  {  "id": 9,  "userId": 1,  "title": "真正理解Mysql的四种隔离级别@",  "description": "事务是应用程序中一系列严密的操作，所有操作必须成功完成，否则在每个操作中所作的所有更改都会被撤消。也就是事务具有原子性，一个事务中的一系列的操作要么全部成功，要么一个都不做。\n\n事务的结束有两种，当事务中的所以步骤全部成功执行时，事务提交。如果其中一个步骤失败，将发生回滚操作，撤消撤消之前到事务开始时的所以操作。",  陈智伟 15:40:15  "content": "### 什么是事务 \n\n> 事务是应用程序中一系列严密的操作，所有操作必须成功完成，否则在每个操作中所作的所有更改都会被撤消。也就是事务具有原子性，一个事务中的一系列的操作要么全部成功，要么一个都不做。\n> \n> 事务的结束有两种，当事务中的所以步骤全部成功执行时，事务提交。如果其中一个步骤失败，将发生回滚操作，撤消撤消之前到事务开始时的所以操作。\n\n\*\*事务的 ACID\*\*\n\n事务具有四个特征：原子性（ Atomicity ）、一致性（ Consistency ）、隔离性（ Isolation ）和持续性（ Durability ）。这四个特性简称为 ACID 特性。\n\n> 1 、原子性。事务是数据库的逻辑工作单位，事务中包含的各操作要么都做，要么都不做\n> \n> 2 、一致性。事 务执行的结果必须是使数据库从一个一致性状态变到另一个一致性状态。因此当数据库只包含成功事务提交的结果时，就说数据库处于一致性状态。如果数据库系统 运行中发生故障，有些事务尚未完成就被迫中断，这些未完成事务对数据库所做的修改有一部分已写入物理数据库，这时数据库就处于一种不正确的状态，或者说是 不一致的状态。",  "created": "2021-12-10 23:46:10",  "status": 0,  "praise": 1,  "favorite": 0,  "label":"[\"java\"]"  },  {  "id": 18,  "userId": 2,  "title": "最值得学习的博客项目eblog",  "description": "视频中所用到的xshell和navicat直接获取哈！去问驱蚊器围墙是",  "content": "请1212",  "created": "2021-12-10 23:46:14",  "status": 0,  "praise": 1,  "favorite": 0,  "label": "[]"  },  {  "id": 23,  "userId": 2,  "title": "java",  "description": "我真不错啊啊哈哈哈",  "content": "java",  "created": "2021-12-16 01:10:37",  "status": 0,  "praise": 1,  "favorite": 1,  "label": "[\"java\"]"  }  ]  } 博客收藏 功能：博客收藏信息增加  地址：/favorite/addfavoriteinfo  传入参数：Favorite favorite  返回：添加博客收藏信息成功  功能：博客收藏信息删除  地址：/favorite/delfavoriteinfo  传入参数：Favorite favorite  返回：删除博客收藏信息成功  功能：获取博客收藏信息  地址：/favorite/getfavoriteinfo  传入参数：int id  返回：{  "code": 200,  "msg": "获取用户id：2博客收藏信息成功",  "data": [  {  "fid": 2,  "fblogid": 3,  "ftime": "2021-12-06 07:54:20"  },  {  "fid": 2,  "fblogid": 16,  "ftime": "2021-12-06 07:55:29"  },  ]  } 博客关注 功能：博客添加关注信息  地址：/subscribe/addsubscribe  传入参数：Subscription subscription  返回：添加关注信息成功  功能：删除博客关注信息  地址：/subscribe/delsubscribe  传入参数：Subscription subscription  返回：删除关注信息成功  功能：获取博客关注信息  地址：/subscribe/getsubscribe  传入参数：int id  返回：获取关注信息成功  {  "code": 200,  "msg": "获取用户id：2关注信息成功",  "data": [  {  "id": 1,  "username": "markerhub",  "avatar": "https://image-1300566513.cos.ap-guangzhou.myqcloud.com/upload/images/5a9f48118166308daba8b6da7e466aab.jpg",  "email": null,  "password": "96e79218965eb72c92a549dd5a330112",  "status": 0,  "created": "2020-04-20 10:44:01",  "lastLogin": null  },  ]  }  功能：获取所有用户被关注信息  地址：/subscribe/getAllsubscribe  传入参数：Subscription subscription  返回：获取所有用户被关注信息成功  功能：获取该用户所有用户被关注数  地址：/subscribe/get1usersubscribe  传入参数：Subscription subscription  返回：获取该用户所有用户被关注数成功 登录注册基本功能接口 地址：/login  请求参数：    返回参数：  登陆成功返回：    否则：   博客发布修改接口 请求地址：/blog/edit  请求参数：    返回参数：     修改密码接口 请求地址：/user/editPassword  请求参数：    返回参数：   用户搜索接口  1. 根据用户名模糊查询所有相似用户   请求地址：/user/search  请求参数：  {  "username": "lize"  }  返回参数：  返回参数：  {  "code": 200,  "msg": "操作成功",  "data": [  {  "id": 2,  "username": "lizeyu",  "avatar": null,  "email": "2869036143@qq.com",  "password": "96e79218965eb72c92a549dd5a330112",  "status": 0,  "created": null,  "lastLogin": null  },  {  "id": 3,  "username": "lizeyu1",  "avatar": null,  "email": "2869036143@qq.com",  "password": "96e79218965eb72c92a549dd5a330112",  "status": 0,  "created": null,  "lastLogin": null  }  ]  }  (2) 根据用户名搜索用户博客接口(准确匹配)  请求地址：/search  请求参数：  {  "username": "markerhub"  }  用户存在返回结果：  {  "code": 200,  "msg": "操作成功",  "data": [  {  "id": 1,  "userId": 1,  "title": "title",  "description": "des",  "content": "content",  "created": "2021-11-29",  "status": 0,  "praise": null  },    {  "id": 7,  "userId": 1,  "title": "你真的会写单例模式吗?",  "description": "单例模式可能是代码最少的模式了，但是少不一定意味着简单，想要用好、用对单例模式，还真得费一番脑筋。本文对 Java 中常见的单例模式写法做了一个总结，如有错漏之处，恳请读者指正。",  "content": "> 作者：吃桔子的攻城狮 来源：http://www.tekbroaden.com/singleton-java.html\n\n\n单例模式可能是代码最少的模式了，但是少不一定意味着简单，想要用好、用对单例模式，还真得费一番脑筋。本文对 Java 中常见的单例模式写法做了一个总结，如有错漏之处，恳请读者指正。\n\n饿汉法\n===\n\n顾名思义，饿汉法就是在第一次引用该类的时候就创建对象实例，而不管实际是否需要创建。代码如下：\n\n```\npublic class Singleton { \n private static Singleton = new Singleton();\n private Singleton() {}\n public static getSignleton(){\n return singleton;\n }\n}\n\n```\n\n这样做的好处是编写简单，但是无法做到延迟创建对象。但是我们很多时候都希望对象可以尽可能地延迟加载，从而减小负载，所以就需要下面的懒汉法：\n",  "created": "2020-05-22",  "status": 0,  "praise": null  },  {  "id": 15,  "userId": 1,  "title": "title",  "description": "description",  "content": "content",  "created": "2021-11-24",  "status": 0,  "praise": null  }  ]  }  否则：  {  "code": 400,  "msg": "failed",  "data": null  }  注：查询结果为 用户名（字符串）中包含传递的参数的用户。 评论功能接口 **(1) 获取评论接口**  请求地址：/comment/get  请求参数：如下图    返回结果：  {  "code": 200,  "msg": "操作成功",  "data": [  [  {  "id": 6,  "userId": 6,  "blogId": 1,  "content": "用户6评论博客1",  "follow": 0,  "username": "6号用户",  "time": "2021-12-10 13:33:45"  }  ],  [  {  "id": 1,  "userId": 2,  "blogId": 1,  "content": "用户2评论博客1",  "follow": 0,  "username": "2号用户",  "time": "2021-12-10 13:33:45"  },  {  "id": 2,  "userId": 4,  "blogId": 1,  "content": "用户4回复用户2对博客1的评论",  "follow": 1,  "username": "4号用户",  "time": "2021-12-10 13:33:45"  },  {  "id": 3,  "userId": 3,  "blogId": 1,  "content": "用户3回复用户4对用户2对博客1的评论的评论",  "follow": 2,  "username": "3号用户",  "time": "2021-12-10 13:33:45"  },  {  "id": 8,  "userId": 6,  "blogId": 1,  "content": "用户6回复用户3对用户4对用户2对博客1的评论的评论的评论",  "follow": 3,  "username": "6号用户",  "time": "2021-12-10 13:33:45"  },  {  "id": 5,  "userId": 1,  "blogId": 1,  "content": "用户1(博主)回复用户2对自己的博客1的评论",  "follow": 1,  "username": "markerhub",  "time": "2021-12-10 13:33:45"  }  ]  ]  }  **(2)发表评论接口**  请求地址：/comment/save  请求参数：  {  "userId": 1,  "blogId": 2,  "content": "content",  "follow": 0  }  返回结果：  {  "code": 200,  "msg": "操作成功",  "data": "success"  }  **(3)删除评论接口**  ​ 请求地址：/comment/delete  请求参数：/blogs/getByClass  {  "userId": 1,  "commentId": 9  }  返回结果  //请求成功返回结果  {  "code": 200,  "msg": "操作成功",  "data": "success"  }  //请求失败返回结果  {  "code": 400,  "msg": "只能删除自己的评论。",  "data": null  } | | | |
| 界面说明登录界面 登录界面使用动态插件vue-particles,自定义过渡色背景色，其中粒子特效可以实现自动连接运动，还可以与鼠标进行互动。（动态GIF图与该报告一起打包在文件夹中）  判空    如果用户在本地保存密码，可以显示用户已保存的密码    3）用户不存在、密码错误等情况的判断      (密码在输入时设置了具体数字不可见的保密功能，保证登录的安全性)  4）登录成功  弹出登录成功的提示框并将输入框加动画浮出，显示登录的账户名称后，进入界面（gif有演示）   浏览、管理博客相关功能  1. 浏览博客并实现点赞、收藏、关注作者等功能   博客界面采用鼠标感应动态展示的显示方式，博客卡片可以根据鼠标的移动方向进行旋转，当鼠标悬浮在卡片之上时，博客标题上移，显示出博客内容（具体实现动态GIF图与该报告一同打包与文件夹中）主界面的分页按钮样式进行修改匹配风格  A、  点击博客卡片下方的收藏或点赞按钮实现收藏和点赞功能，若已收藏或已点赞，再点击则取消点赞或收藏，同时按钮上会动态显示当前该博客的点赞收藏数。    （点赞前后）  B、点击卡片进入博客具体界面      如果该博客为其他用户发布，当前账号可关注该作者       1. 评论功能     发表后：    若为该博客作者发表评论，评论有相应的作者标记：    回复评论（可以实现多用户交叉评论）：      （任何发布的博客或是评论都会显示发布时间）   1. 管理博客功能 2. 发布博客     判空功能：    如果未输入必要信息会给出相应提示  可选择相关标签用于博客发表后的分区搜索：    进行博客内容的编辑：    设置了字体加粗、倾斜、增添下划线、向左对齐、居中对齐、添加连接、添加图片等功能  创建成功：    2、作者发布博客后，仍然可以对博客内容进行修改：      3、设计博客的可见性或是删除     搜索博客或是用户的功能  1. 分区检索（采用标签）   上面在发表博客时，我们可以为博客设置标签，接下来可以通过标签搜索：  下图为各种分区（标签）    以c/c++区为例：  下图为该分区的所有博客     1. 关键字搜索   1、博客搜索        2、用户搜索        该功能支持反复搜索  (也可以在搜索出的界面点击博客卡片进入博客，点击用户卡片进入用户的个人界面，或是实现点赞收藏关注的功能) 用户个人界面 A、用户自己的界面      在右方的的个人资料卡片中，可以展示用户的个人信息，如昵称、邮箱、粉丝数、关注数。  主体部分主要有以下四部分组成：  1、  我的动态：根据时间线显示用户发布的博客    2、  我的收藏：根据收藏的先后顺序显示收藏内容    3、  我的关注：显示关注的用户，并可通过点击进入对方名片    4、  点赞过：显示点赞过的博客    B、其他用户的个人界面  可以通过点击搜索功能搜索出来的名片或者关注的用户的名片进入对方个人界面，出于安全与保密考虑，仅显示部分个人信息   游客模式（未登录模式） 游客模式下仍可浏览博客内容，但是无法进行点赞收藏关注等功能，也无法查看个人界面、评论、写博客。      无法关注    无法评论    无法点赞    无法收藏    无法写博客   注册新账号   判空功能：    不能重名：    判断邮箱格式是否正确：    判断两次密码是否输入一致：    注册成功，有动画效果：     其他功能 利用localstorage，对当前登录的用户的登录状态进行保持，即使在关闭页面之后，为账号主人提供了便利。用户如果在不能保证安全的情况下退出时，可点击退出登录来关闭当前状态，清除缓存。  并通过路由等实现刷新数据不消失的功能 | | | |
| 安全设计 口令采用SecureUtil.*md5*()方式加密。  账号采用基于token的身份验证。   部署说明 环境需求：按照100用户量估算，需要1台服务器，每台4核、1GB内存。初始存储0.5GB，每年扩充多少1GB。网络带宽50MB。  部署方法：  1.        要求win10系统、DB版本mysql8.0.23  2.        创建DB blog，创建DB用户root 密码965288QQczw，执行blog.sql脚本。  3.        后端JAR运行方式java -jar vueblog-0.0.1-SNAPSHOTr.jar。  4 后端使用8081端口，建数据库时为3306端口 | | | |
| 总结讨论 李安祺总结：通过团队项目开发，更加熟悉了后端的从数据设计到编写功能接口的开发过程，也锻炼了团队协作能力和沟通能力。  陈智伟的总结：前端的合作开发，让我更加了解前端的开发流程，如何搭建基础的模块、如何分工，如何分配每个界面之间的关联。更好地进行数据的处理，接口请求的处理。掌握前端开发部分技术，学会自己从0开始。  杨茜云总结：很好地提高了自己的合作能力，自己的设计能力。掌握前端的开发技术vue、elemenUI等。更好地了解合作开发。掌握前端的接口连接、界面样式设计等开发功能流程。  李泽宇总结：懂得了后端数据库的设计——外键的应用等，初步了解了shrio，redis的基本使用方法，锻炼了团队交流协作能力，学会了如何使用Postman测试后端接口，如何编写开发文档。 | | | |