

成 绩: _____

指导教师: _____

基于 Vue 和 SpringBoot 设计个人博客

计算机科学与技术二班 2020416240 彭清元

指导教师 黄宝贵 依托课程 JavaSE、Web 组件开发

摘要: 随着社会进步,记录知识的方法就不仅限于笔记,逐渐由纸质衍生到电子笔记,继而随着电子记录文档方式的多样性发展,大家将所学的知识进行回顾总结时发现, text、md 等格式已无法满足需求,大家需要将整个知识体系从头到尾贯穿起来,对于从事计算机行业的学生和社会人士来说,记录博客是学习和生活的一种重要方式。本文设计了一套后端技术基于 Spring Boot + Mybatis Plus , 前端技术基于 Element UI 框架 + Vue 的个人博客系统,该系统的平台开发语言是基于面向对象思想的 Java 语言,并使用 Mysql 作为本项目的后台数据库,开发出游客及管理者使用的博客网站。该系统的开发可以记录知识并分享的需求,及实现了自己的学习记录,完善了自己的学习系统。

关键字: 个人博客 Web 组件开发 MD5 加密 SSM 框架

Blog 是继 Email、BBS、ICQ 之后出现的第四种网络交流方式,是网络时代的个人“读者文摘”,是以超级链接为武器的网络日记,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。具体说来,博客(Blogger)这个概念解释为使用特定的软件,在网络上出版、发表和张贴个人文章的人。博客专注在特定的课题上提供评论或新闻,其他则被作为比较个人的日记。一个典型的博客结合了文字、图像、其他博客或网站的链接、及其它与主题相关的媒体。能够让读者以互动的方式留下意见,是许多博客的重要要素。大部分的博客内容以文字为主,博客是社交媒体网络的一部分。

1 系统总体方案设计

1.1 总体框架结构设计

个人博客系统设计采用前后端分离技术实现,使用 B/S 的设计模式[1], B/S 是“Browser/Server”的缩写,即“浏览器/服务器”模式,是随着 Internet 技术的兴起不同于 C/S 结构的新的体系架构,在这种结构下,用户操作与交互的界面是通过通用的浏览器(如,IE 浏览器、Google Chrome 浏览器、Firefox 浏览器等)来实现。前端使用当前流行的 Vue 框架进行开发,后端使用 Spring、SpringMVC、Mybatisplus 实现,使用 SpringBoot 技术进行整合,使用 MySQL 数据库进行存储。前端提供可视化页面给用户使用,进行浏览,后端对数据进行增删改查操作,创建接口通过 axios 技术提供前端调用,将数据全部存储在 MySQL 数据库中进行动态显示,使用 JDBC 技术实现后端与数据库进行交互,在 Idea 中使用 Tomcat

服务器运行，页面在浏览器中显示，如图 1.1 所示。

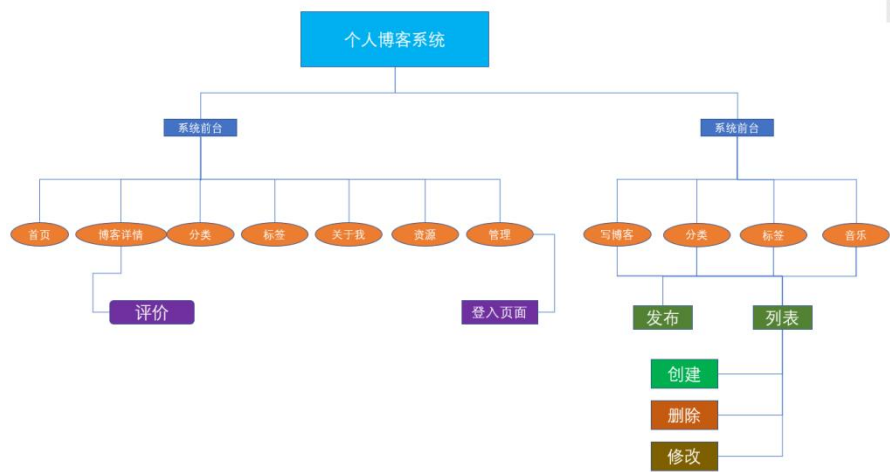


图 1.1 系统结构图

1.2 开发环境介绍

1.2.1 Tomcat 6.0 开发环境介绍

Tomcat 服务器是一个免费的开放源代码的 Web 应用服务器，属于轻量级应用服务器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试 JSP 程序的首选。对于一个初学者来说，可以这样认为，当在一台机器上配置好 Apache 服务器，可利用它响应 HTML（标准通用标记语言下的一个应用）页面的访问请求。实际上 Tomcat 是 Apache 服务器的扩展，但运行时它是独立运行的，所以当你运行 tomcat 时，它实际上作为一个与 Apache 独立的进程单独运行的，如图 1.2 所示



图 1.2 Tomcat

1.2.2 Idea 2021.3.3

intellij idea 2021 是一款市面上最好的 JAVA IDE 编程工具，该工具支持 git、svn、github 等版本控制工具，整合了智能代码助手、代码自动提示等功能，IntelliJ IDEA 强大的静态代码分析和人体工程学设计，让你的开发设计简单轻松，IntelliJ IDEA 将您的源代码编入索引之后，通过在各个环境中提供相



图 1.3 IDEA

关建议，提供快速和智能的体验：即时和智能的代码完成，即时代码分析和可靠的重构工具，如图 1.3 所示。

1.2.3 VsCode

Visual Studio Code（简称“Vscode” [2]）是 Microsoft 在 2015 年 4 月 30 日

Build 开发者大会上正式宣布一个运行于 Mac OS X、Windows 和 Linux 之上的，针对于编写现代 Web 和云应用的跨平台源代码编辑器，可在桌面上运行，并且可用于 Windows，macOS 和 Linux。它具有对 JavaScript，TypeScript 和 Node.js 的内置支持，并具有丰富的其他语言（例如 C++，C#，Java，Python，PHP，Go）和运行时（例如.NET 和 Unity）扩展的生态系统，[3] 如图 1.4 所示。



图 1.4 VsCode

2 前端页面设计及实现

前端主要是使用 Vue 工程进行编写页面样式，使用 axios 将数据获取展示内容，提供可视化界面给用户进行观看和查找内容，前端展示主要分为首页展示、内容分类、标签分类、作者介绍、资源分享、登入界面等功能。在浏览整个页面时引入全局音乐播放盒，播放音乐。

2.1 首页展示

进入首页，将所有的数据做成列表展示，制成分页查询，通过点击文章名称



图 2.1 首页

可以进入文章，浏览文章的具体内容，可以在页面首部对关键字进行查询，对文章标题和内容进行关键字的模糊匹配如图 2.1 所示。

2.2 内容分类

系统中游客视角中内容分类功能是可以根据博主管理员所发布的博客文章根据内容的类别进行划分。分类页面如图 2.2 所示：

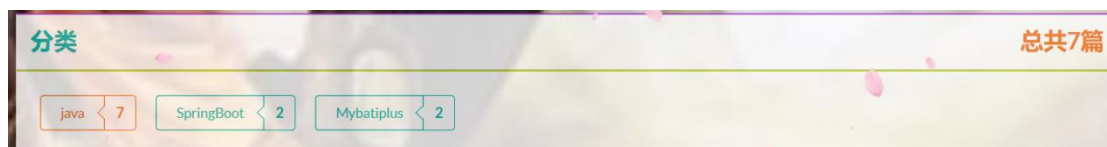


图 2.2 分类标签

通过点击类别，出现对应的文章内容，在文章内容中显示对应的分类标签，如图 2.3 分类内容展示

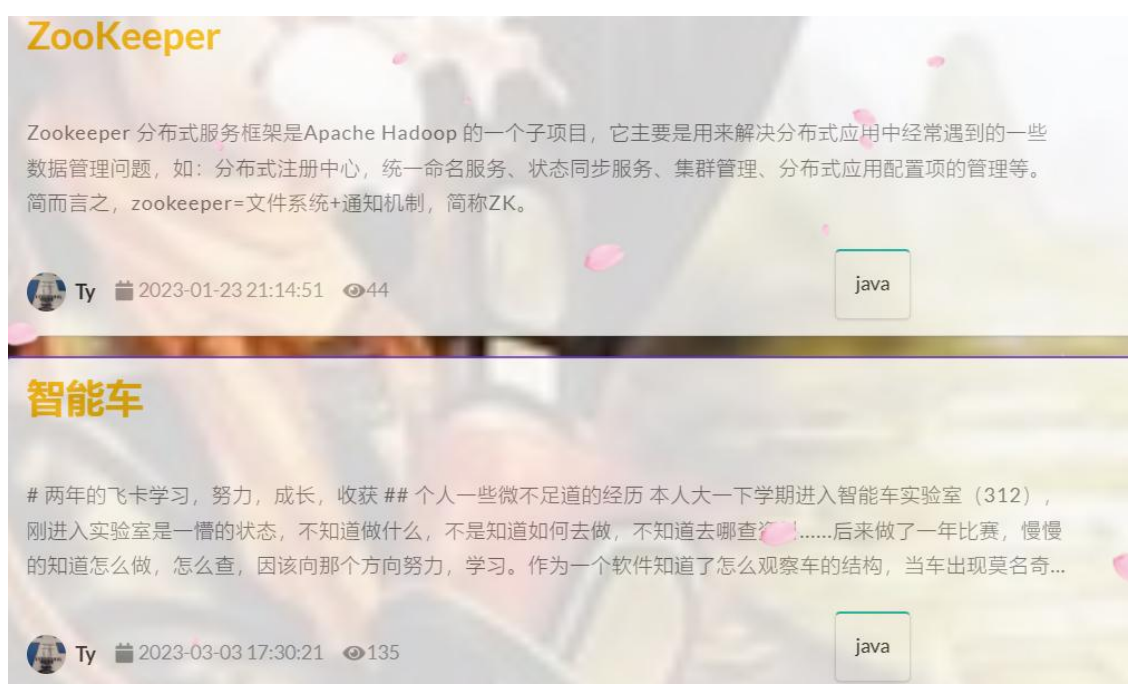


图 2.3 分类内容

2.3 标签分类

系统中游客视角中标签分类功能是可以根据博主管理员所发布的博客文章根据内容的标签进行划分。分类页面如图 2.4 所示：

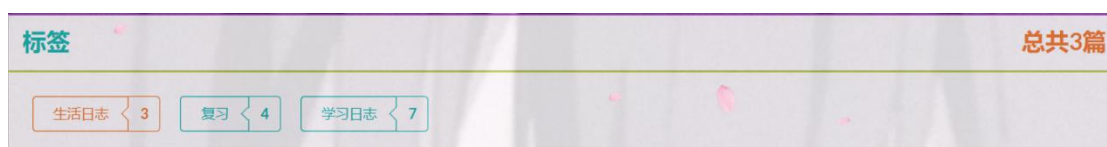


图 2.4 标签内容

点击想查看的标签类别文章，列表渲染出现对应的所有文章，并显示该标签一共

有多少篇文章，页面如图 2.5 所示

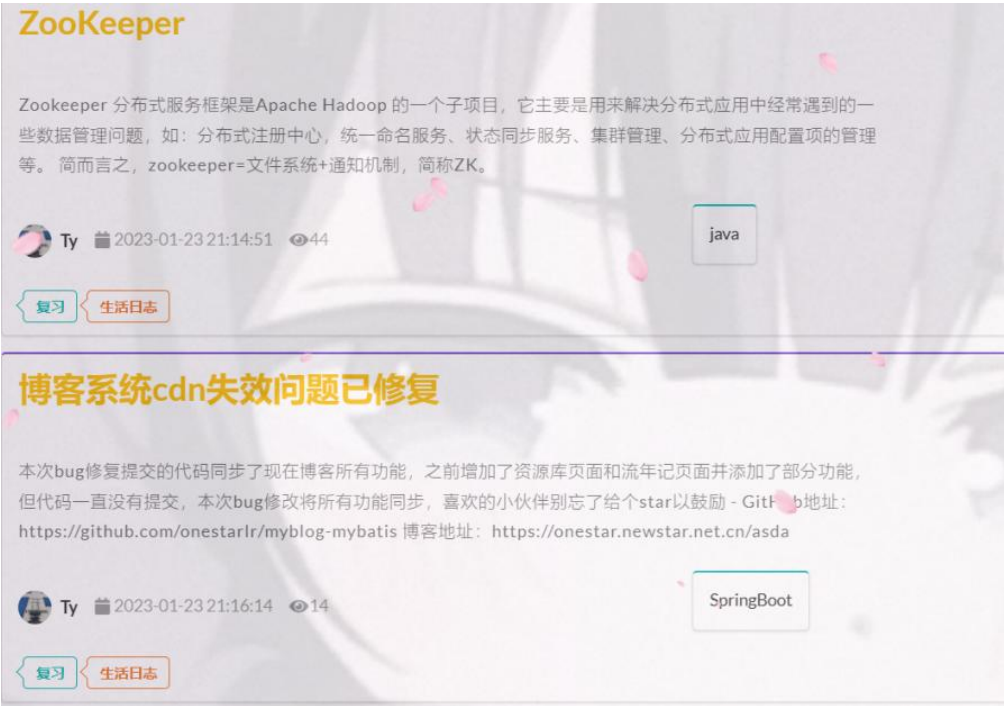


图 2.5 具体内容

2.4 作者介绍

在个人博客系统中，关于我这个模块是为了向大家描述有关于博主作者的一些基本情况，向游客视角用户做一些相关的简单介绍，可以更好的联系到作者，关于查看作者的页面如图 2.6 所示：



图 2.6 关于我

并对博主最近发布的文章根据时间进行分类绘制了折线图，对所有发布的文章根据类别和标签分别绘制了对应的饼状图和柱状图，如图 2.7 所示。

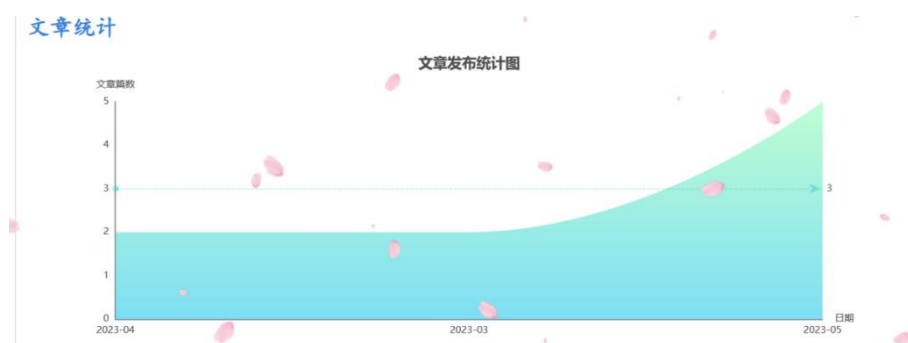


图 2.7 时间统计图

2.5 资源分享

在个人博客系统中，资源分享这个模块是为了记录博主的学习网站，方便查找学习资料，同时也是向游客展示一些学习网站，提供大家学习网站，如图 2.8 所示。



图 2.8 时间统计图

2.5 登入界面

系统中登录功能时管理员进入管理功能的一个入口，该用户需要时数据库中存在的用户才能登录此系统，在登录过程中我们通过账号密码的方式来确认用户是否存在是否可以登录系统，如图 2.9 所示：

Log-in to your account

用户名

password

提交

重置

图 2.9 登入

对于新用户我们可是让新用户进行注册之后在进行登入，如图 2.10 所示

Log-in to your account



用户注册/登录界面。包含两个输入框：'用户名' (username) 和 'password'。下方有两个按钮：'注册' (register) 和 '登入' (login)。

图 2.10 注册

2.6 音乐盒

系统中音乐盒进入界面之后，游客和博主都可以进行听音乐，音乐盒子可以设置循环播放，如图 2.11 所示



图 2.11 音乐盒

通过后台加入自己喜欢的音乐，提供音乐列表，如图 2.12 所示

ID	名称	作者	日期	链接	状态	操作
1	在你的身边	胜哲	2023-03-05 16:30:52	http://music.163.com/song/media/outer/url?id=475479888.mp3	发布	<button>编辑</button> <button>删除</button>
2	错位时空	艾辰	2023-03-05 16:30:59	http://music.163.com/song/media/outer/url?id=1808492017.mp3	发布	<button>编辑</button> <button>删除</button>
3	我走后	小咪	2023-03-05 16:31:15	http://music.163.com/song/media/outer/url?id=2017812455.mp3	发布	<button>编辑</button> <button>删除</button>
4	堕	Zyboy 忠宇	2023-03-08 16:18:25	http://music.163.com/song/media/outer/url?id=2029461010.mp3	发布	<button>编辑</button> <button>删除</button>

图 2.12 音乐内容

提供添加和修改操作，包括音乐名称、作者、音乐链接、创建时间等选项，在进行编辑时采用 Markdown 格式将数据从数据库中调出来进行展示，可供博主进行修改。

2.7 博客详情

展示博客详情属性模块，如图 2.13 所示

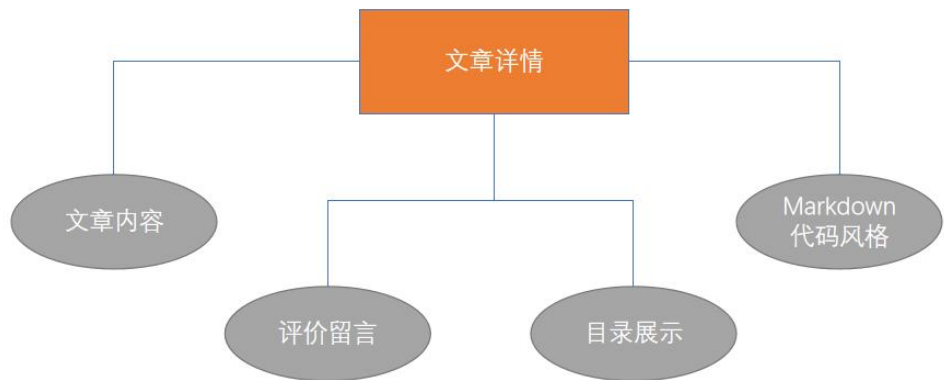


图 2.13 文章属性

博客详情整体格式采用的时 Markdown 格式对内容进行编写，代码使用高亮进行展示，提供目录查看，通过点击目录可以直接跳转到指定的文章内容进行查看，为了提供游客进行评价，提供评价模块，如图 2.14 所示。

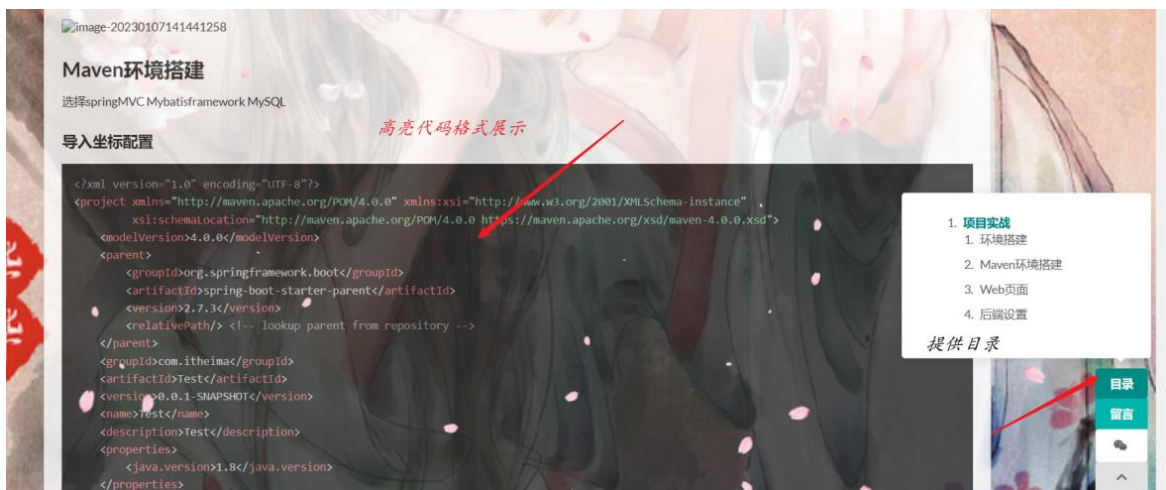


图 2.14 文章代码风格显示

游客通过进行评价可以和博主进行交流；每一篇博客详情使用模板制作将对应篇幅的内容从数据库中提取，将内容展示在页面上给用户观看。

2.7.1 游客评价功能

游客在浏览博客时，可能游客存在疑问或者想和博主进行进一步的交流，此时评价回复功能成为了一个直接渠道，在评价功能中，游客可以提出合理的疑问，当博主或者其他游客浏览时发现，可以加入探讨，因此评论功能成为一个必不可少的功能。通过填写自己评价内容和姓名以及邮箱进行提交。如图 2.15 所示。



图 2.15 文章代码风格显示

2.7.2 评价获取

在从数据库中将评价获取并传送给前台进行显示，首先我们因该是需要对评价进行判断当前这条评价功能时一级评价还是二级或者更多，同意将二级及以上的评价放置于一级评价之后。

具体代码实现如下所示：

//查询出父节点

```
List<Comment> comments =
commentMapper.findByParentIdNull(Long.parseLong("0"));
for(Comment comment : comments){
    Long id = comment.getId();
    String parentNickname1 = comment.getNickname();
    List<Comment> childComments = commentMapper.findParentIdNotNull(id);
    //查询出子评论
    combineChildren(childComments, parentNickname1);
    comment.setReplyComments(tempReplies);
    tempReplies = new ArrayList<>();
}
```

@Override

```
public void combineChildren(List<Comment> childComments, String
parentNickname1) {
    //判断是否有一级子回复
    if(childComments.size() > 0){
        //循环找出子评论的 id
        for(Comment childComment : childComments){
            String parentNickname = childComment.getNickname();
            childComment.setParent_nickname(parentNickname1);
            tempReplies.add(childComment);
            Long childId = childComment.getId();
```

```

        //查询二级以及所有子集回复
        recursively(childId, parentNickname);
    }
}

@Override
public void recursively(Long childId, String parentNickname1) {
    //根据子一级评论的 id 找到子二级评论
    List<Comment> replayComments = commentMapper.findByReplayId(childId);

    if(replayComments.size() > 0){
        for(Comment replayComment : replayComments){
            String parentNickname = replayComment.getNickname();
            replayComment.setParent_nickname(parentNickname1);
            Long replayId = replayComment.getId();
            tempReplies.add(replayComment);
            //循环迭代找出子集回复
            recursively(replayId, parentNickname);
        }
    }
}

```

3 后端接口设计与实现

后端采用的是基于 Spring、MVC、Mybatis 框架进行开发[4]，SpringMVC



图 3.1 后端框架

替代了原来的 Servlet 技术，通过注解方式简化了配置,mybatis 技术在原来的简化了 SQL 语句的编写，最后使用 boot 技术三者进行整合，加速了开发，如图 3.1 所示。

3.1 工具类

在整体项目实现过程中需要使用大量的工具类对各个模块进行优化，比如对于用户注册的密码存入数据库之前做加密处理、不同用户的权限不同，因此对越权范围的处理等等。

3.1.1 MD5 加密

对于用户的个人密码的保护极其重要，我们不能直接将明码直接存入数据库中，防止数据库数据泄露的时候，密码被人获取，因此对密码加密操作是十分重要的，在此项目中对密码的加密使用的是 MD5 加密处理：

```
try {  
    MessageDigest md = MessageDigest.getInstance("MD5");  
    return  
Hex.encodeHexString(md.digest(data.getBytes(StandardCharsets.UTF_8)));  
} catch (NoSuchAlgorithmException e) {  
    e.printStackTrace();  
}
```

对于任何一个密码进行加密之后存入数据库中，想进行破解几乎是不可能的，因此提高了数据安全。

3.1.2 Token&&JWT

在不同用户使用过程中，权限是不相同，于是为了放置越权访问，采用 Token 和 JWT 技术对每一个用户的权限进行限制，如图 3.2 所示[5]

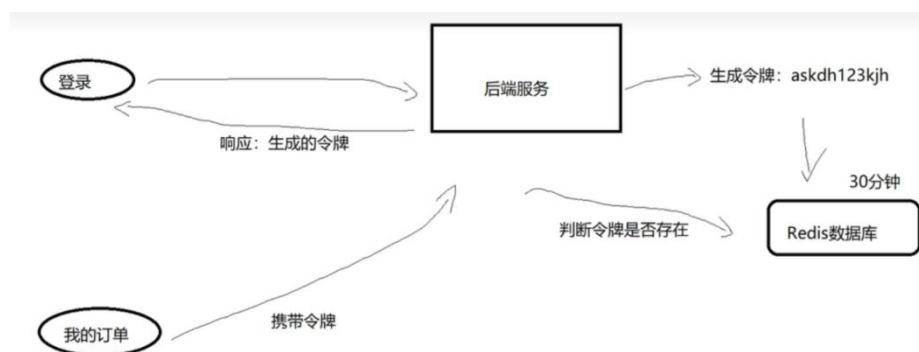


图 3.2 Token 和 JWT 的流程图

Token 和 JWT 的工作过程大致为

- 用户登入，向后端发送请求，后端产生 Token 令牌
- 后端给前端用户响应，将生成的令牌发送给用户
- 后端将生成的令牌存入到 Redis 数据库中
- 用户将 Token 存入到 localStore 中进行保存
- 当其他用户需求发送请求时，会携带一个令牌（Token）
- 后端向 Redis 中取出令牌和用户携带过来的令牌进行对比，如果一致就成功显示，否则返回登入界面

生成 Token 的部分代码

```
public String getToken(String userId, String userRole){  
    System.out.println(System.currentTimeMillis());
```

```

String token = JWT.create()
    .withClaim("userId",userId)
    .withClaim("userRole",userRole)
    .withClaim("timeStamp",System.currentTimeMillis())
    .sign(Algorithm.HMAC256(privateKey));
return token;
}

```

3.1.3 拦截器

拦截器会对所用的 API 进行拦截检查，将其中一些检查没有问题数据进行放行处理，将需要做特殊处理的数据进行做拦截，其中 Token 的检查就是就是在拦截器中检查，在登入之前先通过拦截器进行检查 Token 是否合理，如果不合理则进行拦截，反之就放行。

@Component

```

public class ProjectInterceptor implements HandlerInterceptor {
    @Autowired
    TokenUtils tokenUtils;
    @Value("${token.privateKey}")
    private String privateKey;
    @Value("${token.yangToken}")
    private Long yangToken;
    @Value("${token.oldToken}")
    private Long oldToken;
    @Override
    public boolean preHandle(HttpServletRequest request, HttpServletResponse
response, Object handler) throws Exception {
        System.out.println("preHandle");
        // 获取 header
        String token = request.getHeader("token");
        if(token == null) {
            System.out.println("token 为空");
            return false;
        }
        Map<String,String> map = tokenUtils.parseToken(token);
        String userId = map.get("userId");
        String userRole = map.get("userRole");
        long timeStamp = 0;
        if(!map.isEmpty()){
            timeStamp = System.currentTimeMillis() -
Long.parseLong(map.get("timeStamp"));
        }
        // 判断 Token 是否过期
        if(timeStamp < yangToken){
            System.out.println("年轻 token");
        }
    }
}

```

```

// 刷新 token
else if(timeStamp > yangToken && timeStamp < oldToken){
response.setHeader("token",tokenUtils.getToken(userId,userRole));
}
else {
    System.out.println("过期");
}
if("user".equals(userRole)){
    System.out.println(userRole + "账号");
    return true;
}
if("admin".equals(userRole)){
    System.out.println(userRole + "账号");
    return true;
}
return false;
}
}

```

3.2 DAO 接口于 Service 实现

数据存储 in 数据库当中，当需要时从数据库中调取数据，使用 SQL 语句[6]对数据进行增删改查等操作，需要对文件进行配置，需要对查询语句指定对象和对象所在的包

```

<mapper namespace="com.itheima.Mapper.UserMapper">
    <select id="GetById" resultType="com.itheima.domain.User">
        select * from tb_user where id = 1;
    </select>
</mapper>

```

图 3.3 查询 user 用户

由于需要对数据操作次数过多，于是似乎用 Mybatis 技术，对 SQL 查询进行优化处理，创建 dao 类，创建的每个对象都继承 BaseMapper 类，很大程度上简化了开发，如图 3.3 所示。

3.2.1 Mapper 对象类

为所有的实体类对象创建自己的 Mapper 类，每个 Mapper 类对象继承 BaseMapper 类，在 Mapper 类中就可以使用 Mybatis 类提供方法，也可以自定义使用 SQL 进行编写，灵活性非常高。

@Mapper

```
public interface TypeMapper extends BaseMapper<Type> {
```

```
    @Select("select * from t_type where name = #{name}")
```



```

public Type SelectByName(String name);

@update("update t_type set name = #{newname} where name =
#{lastname}")

public Boolean UpdateByName(String lastname, String newname);

}

```

3.2.2 Service 创建和实现

对各个对象的属性进行操作需要提供独自の API，在 Service 类中创建接口，在 ServiceImpl 类继承 Service 接口，使用自动装配将 Mapper 中的实体类注入到 ServiceImpl 中使用，实现 CRUD 功能，如图 3.4 所示。[8]

```

@Service
public class AplayerServiceImpl implements AplayerService {

    @Autowired
    private AplayerMapper aplayerMapper;

    @Override
    public Boolean save(Aplayer aplayer) {

        int insert = aplayerMapper.insert(aplayer);

        return insert == 1 ? true : false;
    }
}

```

图 3.4 保存用户功能

3.3 Web 层服务

使用 B/S 的设计模式对项目进行展示，在服务器中运行，web 技术必然是不可或缺的，对比 Servlet，由于 springMVC 是方法级别的拦截，springMVC 的方法之间基本上独立的，独享 request response 数据。它还可以和很多框架进行无缝集成，开发效率和性能高。现在很多公司都采用 springMVC 来进行开发，咱们公司也是采用 springMVC 来进行。

3.4 属性配置

在对数据库操作之前需要，连接 IDEA 和 SQL 进行连接，如图 3.5 所示

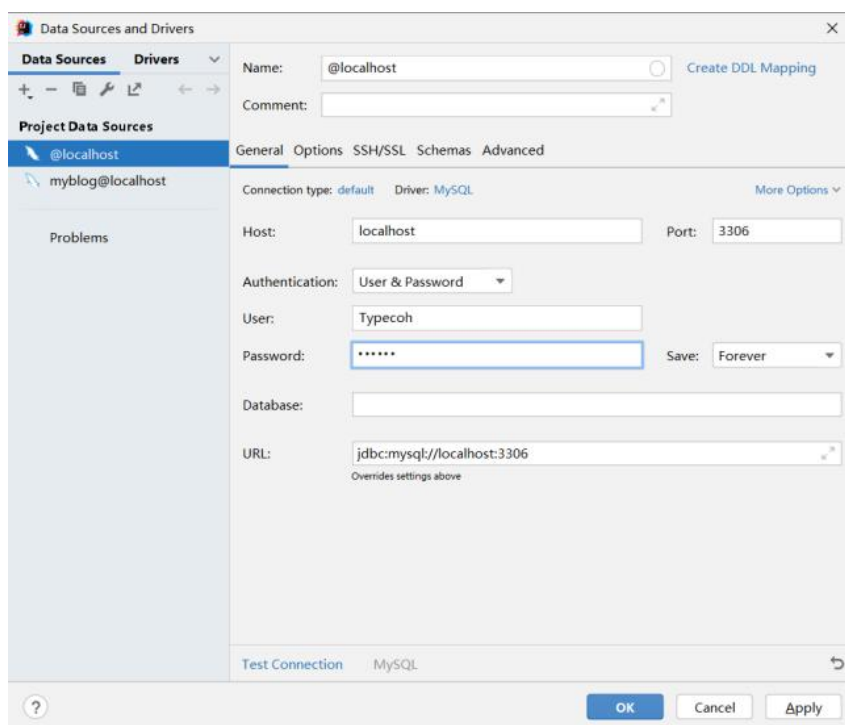


图 3.5 连接数据库

连接成功之后将需要使用的数据库的信息提供给 java，将 java 于对应的数据库进行绑定

datasource:

driver-class-name: com.mysql.jdbc.Driver

url:

`jdbc:mysql://localhost:3306/myblog?useUnicode=true&characterEncoding=utf-8`

username: Typecoh

password: 123456

3.4 日志记录

SpringBoot 使用 Commons Logging 记录所有内部日志（默认使用，因为底层直接内嵌，引入其他日志框架，要引入 jar 的麻烦，就直接是默认使用），但开放日志的底层实现，现在主流使用 Logback 默认情况下，Spring Boot 会用 Logback 来记录日志，并用 INFO 级别输出到控制台，如图 3.6 所示

日志的相关代码展示

// 记录 登录信息

@Before("log()")

```
public void doBefore(JoinPoint joinPoint){
```

```
    ServletRequestAttributes attributes = (ServletRequestAttributes)
```

```
    RequestContextHolder.getRequestAttributes();
```

```
    HttpServletRequest request = attributes.getRequest();
```

```
    String requestURL = request.getRequestURL().toString();
```

```
    String localAddr = request.getLocalAddr();
```

```

String classMethod = joinPoint.getSignature().getDeclaringTypeName() + "." +
joinPoint.getSignature().getName();
Object [] args = joinPoint.getArgs();
requestLog.setUrl(requestURL);
requestLog.setIp(localAddr);
requestLog.setClassMethod(classMethod);
requestLog.setAgrs(args);
logger.info("-----记 录 日 志
-----");
logger.info("RequestLog: {}" + requestLog);
}

```

```

2023-03-31 09:08:18.278 INFO 11944 --- [main] o.hibernate.annotations.common.Version : HCANN000000: Hibernate Commons Annotations {5.1.2.Final}
2023-03-31 09:08:18.374 INFO 11944 --- [main] org.hibernate.dialect.Dialect : HH0000400: Using dialect: org.hibernate.dialect.MySQL8Dialect
2023-03-31 09:08:18.504 INFO 11944 --- [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HH0000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
2023-03-31 09:08:18.513 INFO 11944 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'

-----
3.4.2
2023-03-31 09:08:19.421 WARN 11944 --- [main] c.b.n.core.metadata.TableInfoHelper : Can not find table primary key in Class: "com.itheima.pojo.APlayer".
2023-03-31 09:08:19.679 WARN 11944 --- [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering.
2023-03-31 09:08:19.830 INFO 11944 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8082 (http) with context path ''
2023-03-31 09:08:19.840 INFO 11944 --- [main] com.itheima.BlogApplication : Started BlogApplication in 4.661 seconds (JVM running for 6.008)

```

图 3. 6 日志打印结果

4 数据库存储设计与实现

4. 1 数据库 E-R 图设计

E-R 图也称实体-联系图 (Entity Relationship Diagram), 提供了表示实体类型、属性和联系的方法, 用来描述现实世界的概念模型。它是描述现实世界概念结构模型的有效方法。是表示概念模型的一种方式, 用矩形表示实体型, 矩形框内写明实体名; 用椭圆表示实体的属性, 并用无向边将其与相应的实体型连接起来; 用菱形表示实体型之间的联系, 在菱形框内写明联系名, 并用无向边分别与有关实体型连接起来, 同时, 在无向边旁标上联系类型 (1:1, 1:n 或 m:n)。连线: 实体与属性之间; 实体与联系之间; 联系与属性之间用直线相连, (对于一对一联系, 要在两个实体连线方向各写 1; 对于一对多联系, 要在一的一方写 1, 多的一方写 N; 对于多对多关系, 则要在两个实体连线方向各写 N,M。)。

4. 1. 1 数据关系的介绍

在个人博客中存储的数据主要包括有音乐播放器对象信息、博客内容信息、博客分类信息、音乐信息、博客标签信息、登入用户信息。

4.1.2 数据属性

每一个对象都有自己的属性, 对于博客而言存在标题、创建时间、修改时间、分类、标签、观看次数、是否发表等。如图 3.7 所示

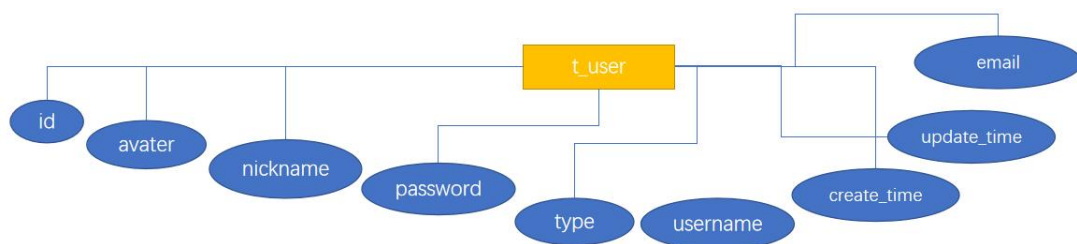


图 3.7 用户表的属性

4.1.3 E-R 图的联系

数据对象之间存在一定的关系，对于一篇博客来说，是有对应的标签和分类属性，因此对于标签和分类而言，标签和博客之间的对应关系是一对多，分类和

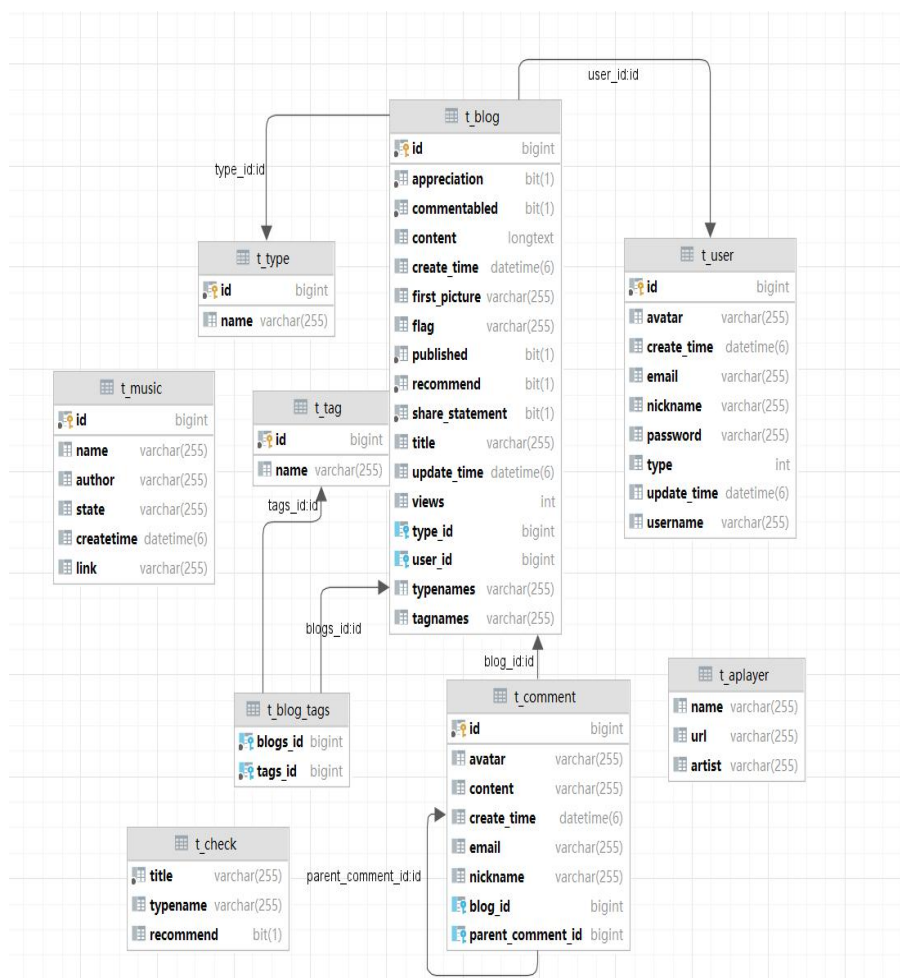


图 3.8 E-R 图

博客之间也是一对多的关系，而博客对于标签和分类是多对一的关系。对象与对象之间的关系，我们使用 E-R 图进行表示出来，通过 E-R 图的表示，进行化简之后可以明确的看出，每一张数据表中的主键和外键，对数据表的设计提供了方便如图 3.8 所示。

4.2 数据库表设计

根据上面的需求，个人博客网站的数据库中的各个表的设计结构如图所示，每个表格表示在数据库中的一个表，如表 4-1 所示

表 4-1 歌曲

字段名称	数据类型	主键	描述
Id	Int	是	歌曲 Id
link	varchar	否	歌曲链接
author	varchar	否	歌曲作者
name	varchar	否	歌曲名称
state	varchar	否	歌曲发表状态
createtime	varchar	否	歌曲创建时间

博客详情如表 4-2 所示

表 4-2 博客

字段名称	数据类型	主键	描述
Id	Int	是	博客 id
content	varchar	否	博客内容
create_time	varchar	否	博客创建时间
first_picture	varchar	否	博客的配图
recommend	varchar	否	博客推荐
title	varchar	否	博客标题
update_time	varchar	否	博客更新时间
views	varchar	否	博客参观次数
type_id	varchar	否	博客类别 id
user_id	varchar	否	博客作者 id
author	varchar	否	博客作者

博客评论详情如表 4-3 所示

表 4-3 评价

字段名称	数据类型	主键	描述
Id	Int	是	评论 id
avatar	varchar	否	评论照片
content	varchar	否	评论内容
create_time	varchar	否	评论时间
email	varchar	否	评论人的邮件
nickname	varchar	否	博主姓名
blog_id	Int	否	博主的 Id
parent_comment_id	varchar	否	评论的父评论

标签如表 4-4 所示表

4-4 标签

字段名称	数据类型	主键	描述
Id	Int	是	标签 id
name	varchar	否	标签名称

类别如表 4-5 所示表

4-5 标签

字段名称	数据类型	主键	描述
Id	Int	是	类别 id
name	varchar	否	类别名称

博客用户详情如表 4-6 所示

表 4-6 用户

字段名称	数据类型	主键	描述
Id	Int	是	用户 id
avatar	varchar	否	用户照片
content	varchar	否	评论内容
create_time	varchar	否	用户创建时间
email	varchar	否	用户邮件
nickname	varchar	否	用户姓名
password	varchar	否	用户密码
type	varchar	否	用户身份

根据设计好的数据库表，使用 SQL 语句创建数据库，生成对应的数据库的表

4.3 事务处理

事务是一种机制、一个操作序列，包含了一组数据库操作命令。事务把所有的命令作为一个整体一起向系统提交或撤销操作请求，即这一组数据库命令要么都执行，要么都不执行，因此事务是一个不可分割的工作逻辑单元。在数据库系统上执行并发操作时，事务是作为最小的控制单元来使用的，特别适用于多用户同时操作的数据库系统。

4.3.1 原子性 (Atom)

事务的一组操作是原子的不可再分割的，这组操作要么同时完成要么同时不完成。

4.3.2 一致性 (Consistency)

事务在执行前后数据的完整性保持不变。数据库在某个状态下符合所有的完整性约束的状态叫做数据库具有完整性。在解散一个部门时应该同时处理员工表中的员工保证这个事务结束后，仍然保证所有的员工能找到对应的部门，满足外键约束。

4.3.3 隔离性 (Isolation)

当多个事务同时操作一个数据库时，可能存在并发问题，此时应保证各个事务要进行隔离，事务之间不能互相干扰。

4.3.4 持久性 (Durability)

持久性是指一个事务一旦被提交，它对数据库中数据的改变就是永久性的，不能再回滚。

4.3.5 解决方案

@Transactional 是 java 中使用的注解形式的事务,也就是说可以直接使用该注解来完成事务操作.该注解是通过 JDBC 的事务 + Spring 的 AOP 动态代理来完成的,如图 4.1 所示.

```
@Transactional
public interface TypeService {

    // 增加一篇文章
    public Boolean saveType(Type type);

    // 删除一篇文章
    public Boolean deleteType(Integer id);

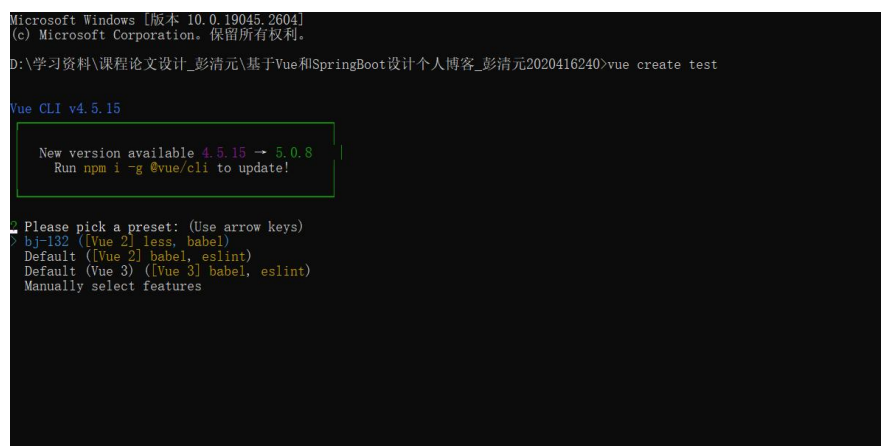
    // 修改一篇文章
    public Boolean updateType(String lastname,String
```

图 4.1 绑定事务

5 项目搭建

5.1 Vue 创建

创建文件夹,使用 cmd 打开控制台,输入 create vue name 进行创建文件,如图 5.1 所示



```
Microsoft Windows [版本 10.0.19045.2604]
(c) Microsoft Corporation. 保留所有权利。

D:\学习资料\课程论文设计_彭清元\基于Vue和SpringBoot设计个人博客_彭清元2020416240>vue create test

Vue CLI v4.5.15

New version available 4.5.15 -> 5.0.8
Run npm i -g @vue/cli to update!

Please pick a preset: (Use arrow keys)
> b1-132 ([Vue 2] less, babel)
Default ([Vue 2] babel, eslint)
Default ([Vue 3] babel, eslint)
Manually select features
```

图 5.1 创建 Vue 项目

选择合适的语法进行创建

5.2 SpringBoot 创建

此项目创建使用的 IDEA, 打开 idea 软件, 常见工程, 如图 5.2 所示

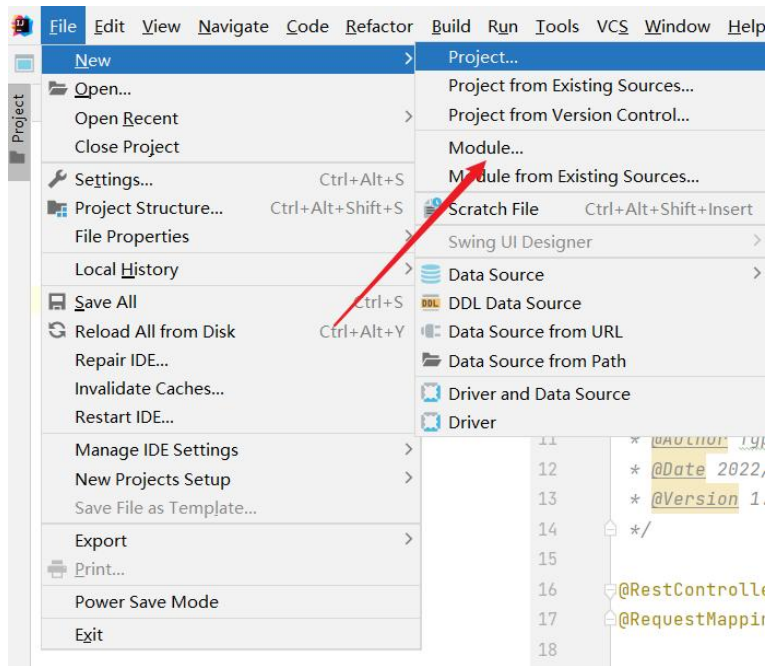


图 5.2 创建 SpringBoot 项目

选择所需要创建项目的类型，如图 5.3 所示

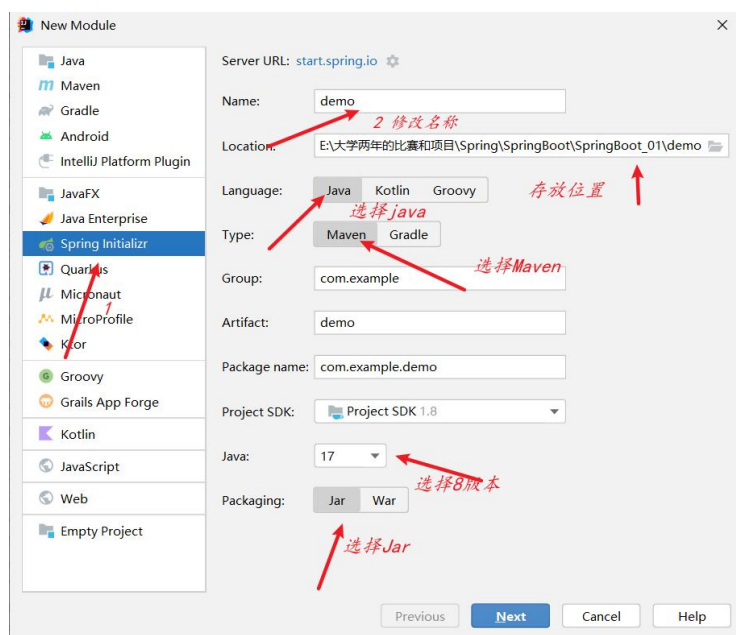


图 5.3 选择版本

选择 SpringWeb 框架进行搭建，使用 SpringBoot 版本是 2.7.3 版本，选择版本不易过高，版本过高会导致，创建的项目无法启动，创建失败，如图 5.4 所示；

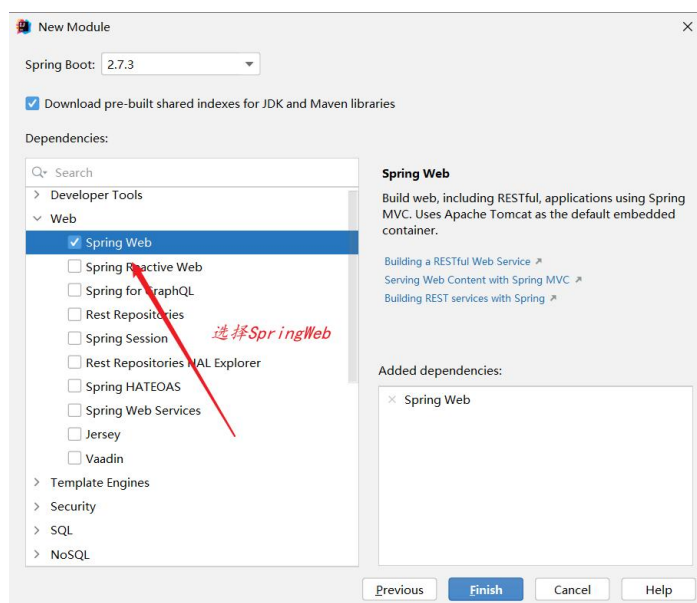


图 5.4 创建 SpringWeb

创建成功之后进行测试，如图 5.5 所示

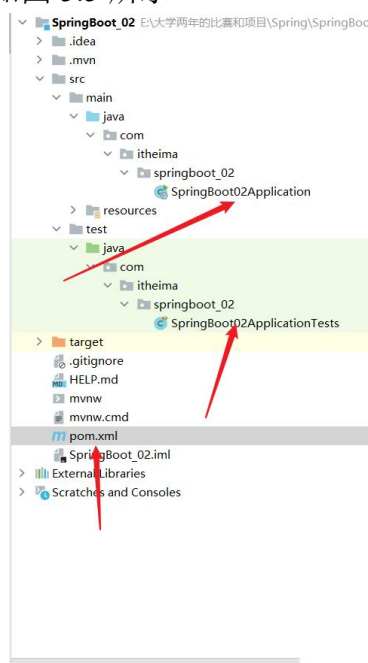


图 5.5 测试 SpringBoot

6 总结

本系统作为一个完整的博客系统，主要包括了前台页面浏览和后态管理功能，其中前台浏览主要是包括首页浏览、文章分类、文章标签分类、作者自我介绍、登入界面、音乐播放器的循环播放；后台设计接口提取数据库中的数据，通过 axios[9]技术传送给前台页面；在后台中进行数据添加、修改、删除，对于文章内容，均是采用的 Markdown 格式，方便用户进行编写，查看。

相比于流行的博客设计，本系统的页面简洁，操作方便，界面美观，引入音乐播放功能，并实现了对应模块的增删改查功能。

完成这个系统,我得到了很大的锻炼,从 0 到 1 完整的将一个项目开发完成,从数据设计,数据分析,到数据库的创建、前端页面的布局、设置、编写,最后到后端提供接口,操作数据库等。

参考文献

- [1] 毛晓东,于童。基于 B/S 结构的煤矿员工绩效考核系统设计[J].电脑编程技巧与维护,2016(06):59-60.
- [2] Menu license links to non Open Source license · Issue #60 · microsoft/vscode · GitHub.
- [3] Visual Studio Code 文档 .
- [4] 李天赐,李璟璐,于姗姗,刘梓艺。基于 Spring MVC 的高校学院学工助理系统的设计与实现[J].智能计算机与应用,2018,8(04):167-169.
- [5] 柯采.基于 J2EE 和 MVC 模式的 Web 应用研究[J].软件,2020,41(03):165-167.
- [6] 蒋睿.MySQL 数据库安全研究[J].电脑知识与技术,2020,16(09):3-4+21.
- [7] 孙卫琴. Tomcat 与 Java Web 开发技术详解(第 2 版)[M]. 北京: 电子工业出版社,2012.
- [8] 乔岚。基于 MyBatis 和 Spring 的 JavaEE 数据持久层的研究与应用[J].信息与电脑(理论版),2017(08):73-76.
- [9] 林萍,朱婵。基于 Ajax 技术和 JAVAEE 的分页查询优化[J].计算机系统应用,2017,26(08):184-189.