## 计划书

[BookManager\_NowcoderGit1\_牛客网](https://git.nowcoder.com/11000160/BookManager) 牛客项目源代码和readme指导

### 1 基本计划

* 设计逻辑分析和思考以及代码自己实操检验
* 与简历适配小要求：将freemaker换成
* 代码规范（优雅代码什么样子你知道吗）
* 项目完整运行起来
* 有效痛点与解决（中间包含利用搜索引擎、文档、Demo、Github解决问题的能力）
* 能做什么检验（最基础的增删改查，最基础的项目完全通透并且能达到自己满意和老师满意的掌握程度那么市场一般也就满意了，其他进阶或晋级的看情况）

### 2 莽代码

* 代码规范（优雅代码什么样子你知道吗）
* 设计逻辑
* 自己实操检验

--知道套路逻辑框架和掌握基本的逻辑思路就好了，需要就自己写写，知道逻辑就很容易写出来的，或者看demo就直接能移过来修改

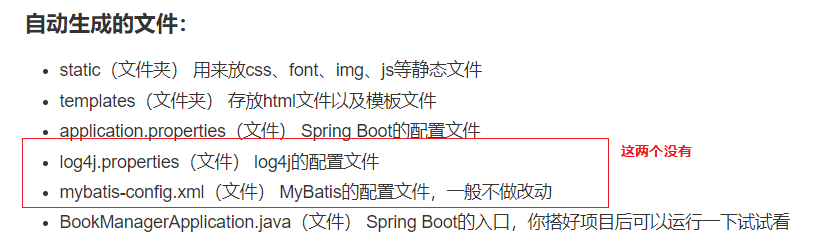
* 有效痛点与解决（中间包含利用搜索引擎、文档、Demo、Github解决问题的能力）
* 能做什么检验（最基础的增删改查，最基础的项目完全通透并且能达到自己满意和老师满意的掌握程度那么市场一般也就满意了，其他进阶或晋级的看情况）

### 3 对于源代码和readme指导的疑问

#### 3.1 Aop模块

在摘录那里

#### 3.2 目录

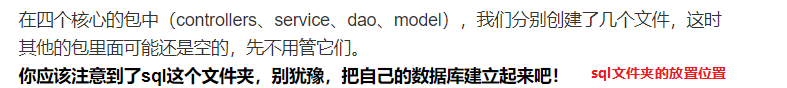


log4j的日志配置文件是没问题的，之前学习Spring boot的时候也有学过日志的配置文件（见笔记的日志-用户配置文件）

并且这里用的是log4j，而我们推荐和学习的都是**logback，日志配置文件可以下载网上的.xml/.properties之后进行一定的修改，也可以创建.xml/.properties自己写**



#### 3.3 sql文件夹



### 4 对于源代码和readme指导的摘录

#### 4.1. 本项目中使用了Freemaker、Web、MyBatis、Aspect四个 模块，填写好GroupId和ArtifactId（ArtifactId使用BookManager）之后，下载到本地，用 [你自己的IDE打开pom.xml文件](大大大)，待pom.xml文件解析下载完毕后，我们的项目框架就搭建好了。

Aspect模块的依赖是：

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-aop</artifactId>

</dependency>

[Springboot中Aspect实现切面（以记录日志为例）\_朱哲的博客-CSDN博客](https://blog.csdn.net/zhuzhezhuzhe1/article/details/80565067)

[Spring Boot2(六)：使用Spring Boot整合AOP面向切面编程 - 云+社区 - 腾讯云](https://cloud.tencent.com/developer/article/1457465)

[Spring Aop(aspect)的详细使用 - 简书](https://www.jianshu.com/p/830e799e099b)

**虽然暂时不知道Aspect是怎么在Spring Initializr 添加进去这个模块的，但是知道了这个项目是有用到Aop的，就又多了个挖掘点**

#### 4.2. Web

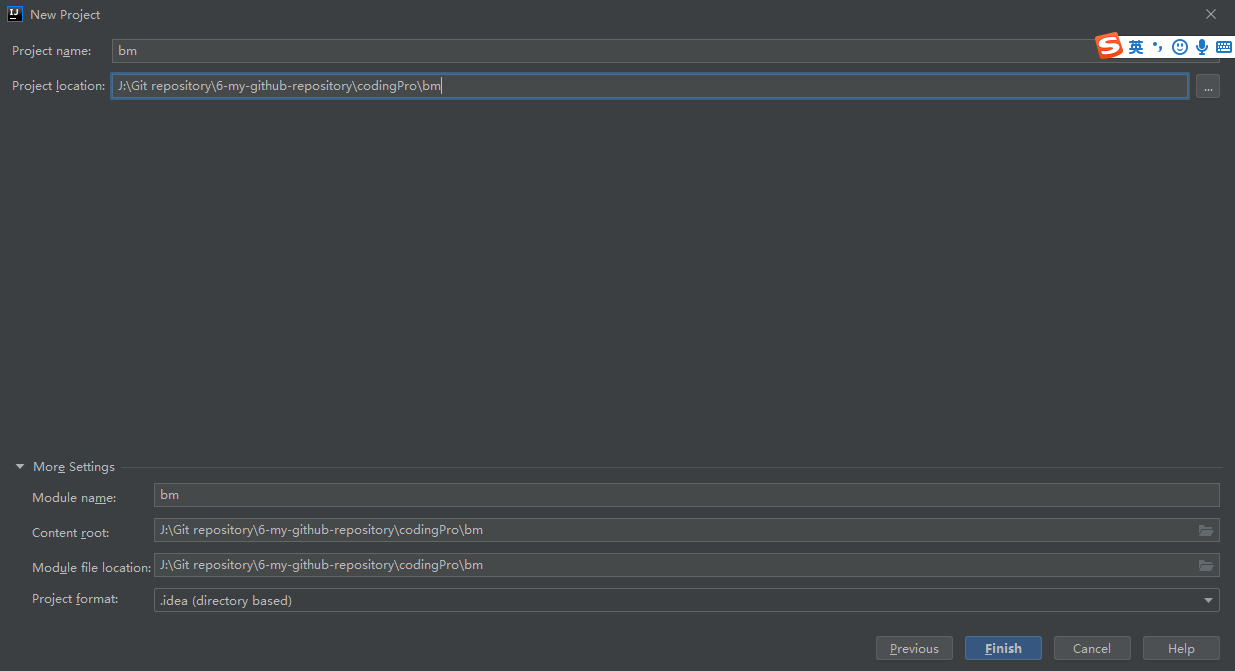
Web组件本质上就是SpringMVC，用来控制整个MVC主体流程，部分同学将SpringMVC理解为Web  
端的入口也是可以的。

#### 4.3. 如果同学们对任何地方有疑问一定要去百度、Google

#### 4.4 关于项目创建的一个目录问题的点

关于项目创建的一个目录问题的点，一直搞不太清楚，而且容易忘，鼓励按照默认去，适当截图贴图辅助记忆。这里也记记，多记几次就不难了。就是要多建一层同名目录，一般都是IDEA自动建，要手动建也可以

（如果我没有再建一层bm的话，那么项目是不是要附着在codingPro上了）



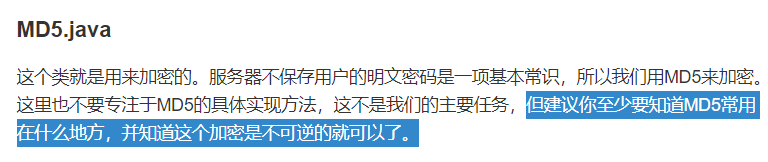


#### 4.5 代码一定一定要自己敲才有用

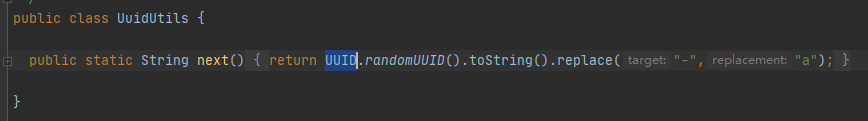
#### 4.6 这里的service只做了最基本的包装，相信你能写的更漂亮！

#### 4.7 好了！你一定要自己敲代码！不然你的收获将非常有限！

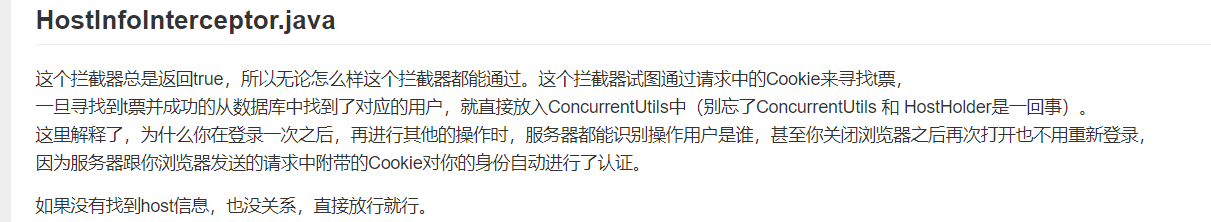
#### 4.8 工具类







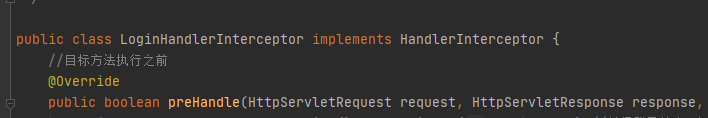
#### 4.9 思考一下，我们在登录的时候会写入host信息到HostHoder里面去，但是我们在操作图书而不是做登录操作的时候，怎么将host信息写到HostHoder中呢？



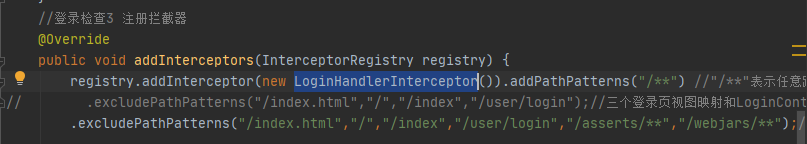


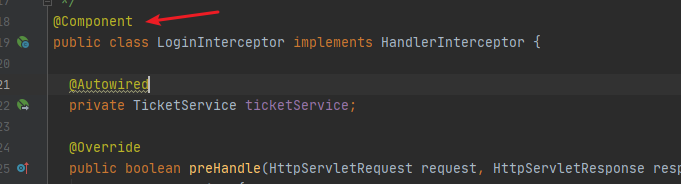
### 5 项目点的记录

#### 5.1 直接new创建对象和IOc控制反转注入创建对象



CRUD项目中





图书管理系统中



#### 5.2 对于如果要Thymeleaf改写

* 1. 空对象怎么处理？（因为之前的Demo只有对空字符串的判断，所以查一下网络）

那个if host可以类似**下面的写法**，观察图书控制器的代码知道host其实是从host工具中取出来的user对象，所以就去user对象的userName看看是不是空值就好了，然后应该是可以写进table标签里面的，根据我对Thymeleaf的认知

<div class="col-md-6" th:if="${not #strings.isEmpty(user.userName)}">

**哦哦哦我看到了，其实CRUD中也有对对象（emp）进行一个判空判断，就是直接可以th:if="${host != null}"--而且这个基本上是万能的，无论是对象、字符串、数值。。。都可以用**

还看到这个，应该在之后对于一些判断会有用，就随便记录一下

<span th:each="i:${course?.teachers}"

                  th:text="${i}+'<br>'">

            </span>

给需要判断的对象加一个问号就可以！

使用 ${xxx?.xxx?} 可以判断对象是否为空

? 号 前的对象或者属性为空那么就不再进行渲染，而且不会导致出现报错的情况 [链接](http://www.bubuko.com/infodetail-3073152.html) [链接](https://bbs.csdn.net/topics/392364154?list=63187901)

尤其这个链接里面大量使用了这里的判断对象空值：[强烈推荐可以看或者后面作为引子](https://blog.csdn.net/zhouzhiwengang/article/details/90232406)

--到此改写基本上成功，对于Thymeleaf的应用也基本上掌握了，不过后面还是要继续巩固。然后对于以后遇到新的情况就网上搜搜看看demo代码或者解决方法基本上就都解决了。也可以查看技术手册，技术手册就是比较考验技术，而且没有那么强的针对性，但是它是最全和最准确的，一旦你能用技术手册解决问题了那么基本上对你没什么问题了

* 1. end

#### 5.3 关于Dao文件也就是Mybatis取值的#{}，没学过但是我想起了之前精进那里的，通了通了

**mybatis防止SQL注入（其实很简单 用# 不用$）**

**【结论】在编写MyBatis的映射语句时，尽量采用“#{xxx}”这样的格式。若不得不使用“${xxx}”这样的参数，要手工地做好过滤工作，来防止SQL注入攻击。**

**#{}：相当于JDBC中的PreparedStatement**

**${}：是输出变量的值**

**简单说，#{}是经过预编译的，是安全的；${}是未经过预编译的，仅仅是取变量的值，是非安全的，存在SQL注入。**

[MyBatis 中#{}和${}区别\_w3cschool](https://www.w3cschool.cn/mybatis/mybatis-yta93bpj.html)

#### 5.4 有没有发现Controller的方法很多参数都是model，用来传递数据过去前端

也就是如果我们需要往前端传什么东西我们就在方法参数加一个model，把一些视图传递过去，model.addAttribute()用法类似map(String, Object)，可以添加对组键值对，例如bookList就添加了两组

#### 5.5 重定向的语句还是蛮重要的，要明确一下，其实很简单

就是return “redirect: + 新请求~~/指定url~~”

return "redirect: /index "

转发就是把redirect换成forward，写法一样

return "forward:/error"

注：后面想了下不太确定指定url可不可以，按原理说其实是可以的，但是一般好像对应到指定的html文件都是直接return “\*\*\*/\*\*\*”，不过重定向和转发请求是一定没问题的

一般对应到指定的html文件都是直接return “\*\*\*/\*\*\*”：

return "book/books";

return "book/addbook";

return "404";

#### 5.6 个人不算规律的小发现--Ioc、Bean和new

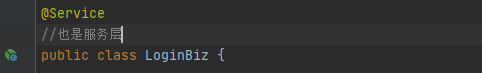


**也就是Controller层、Dao层、Service层和不知哪一层的Component层**

所以就是不属于以上四类基本上就不能用Ioc，例如这里的model，不过用new是一样的

（当然你要在model层的类加@**Component注释成为Bean也是可以用IOC方式的，拦截器就是加**@**Component的；而spring04里最开始的Dao层是@ Repository的，现在@Mapper不知道怎么说**）





#### 5.6-1 对5.6的补充

[Spring 与 Mybatis 中的 @Repository 与 @Mapper\_Xu\_JL1997的博客-CSDN博客](https://blog.csdn.net/Xu_JL1997/article/details/90934359)

**@Mapper 是 Mybatis 的注解，和 Spring 没有关系，@Repository 是 Spring 的注解，用于声明一个 Bean。**（重要）

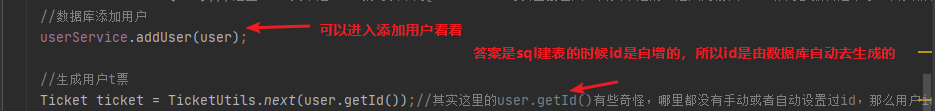
在 Spring 程序中，Mybatis 需要找到对应的 mapper，**在编译的时候动态生成代理类，实现数据库查询功能**，所以我们需要在接口上添加 @Mapper 注解

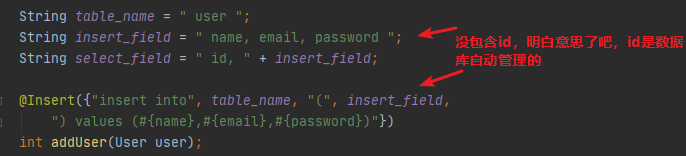
**总结：**

1. @Mapper 一定要有，否则 Mybatis 找不到 mapper。
2. @Repository 可有可无，可以消去依赖注入的报错信息。
3. @MapperScan 可以替代 @Mapper。

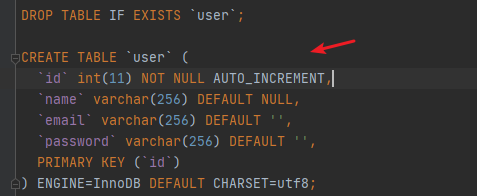
[springboot中@Mapper和@Repository的区别 - &天涯海角& - 博客园](https://www.cnblogs.com/sitian2050/p/11872261.html)

#### 5.7 关于之前疑问的id问题





数据库：



初值大概是00000000001，然后自增

同理book、ticket也是同样的实现

#### 5.8 Controller和Interceptor的重定向和转发的区别

Controller就很容易通过return和关键词就能去实现，在5.5有具体实现实例可以去看

在拦截器中就是基于它的固定参数response来实现的（会问为什么是response来实现？那你觉得重定向和转发不是响应发起的难道是request请求发起的？当然是中间的响应去sendRedirect的啦）

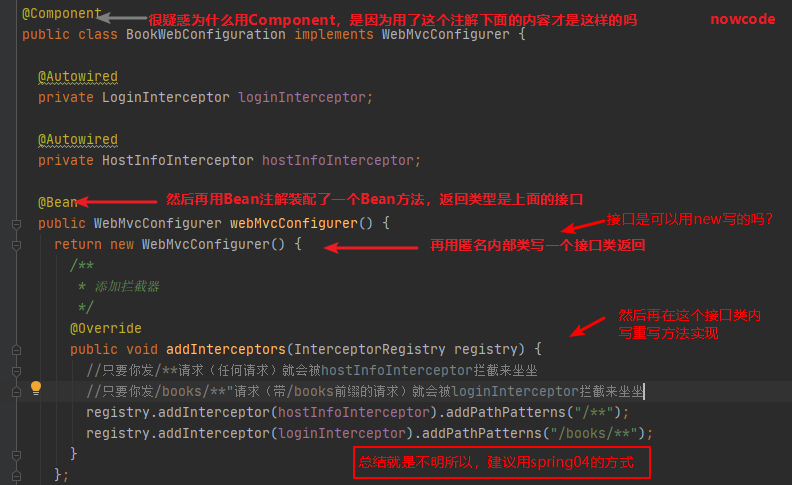


就下面的模板语句：

response.sendRedirect("/users/login");

#### 5.9 关于configurer的实现nowcode和spring04好像又有不同

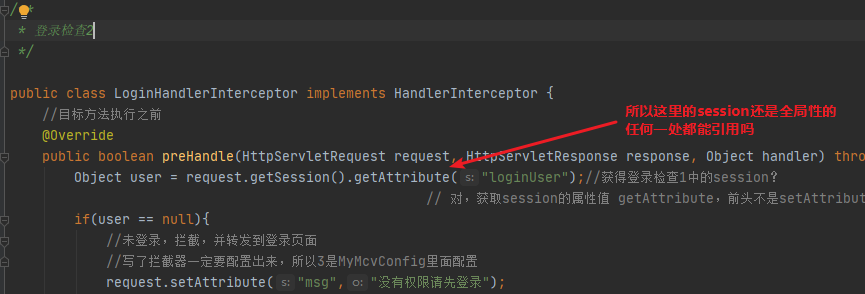


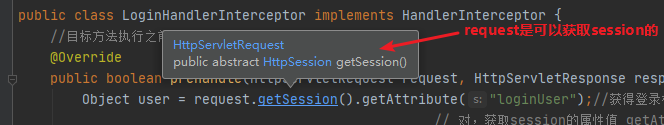


### 6 对于项目中自己发现的疑问

#### 6.1 还是没怎么懂其中的session，就是里面的机制没怎么懂，







就大致的理解就是response和request其实就是每一次的http操作的一个基本单位，http不就是请求和回应嘛，然后他们两其实日常最常见的操作就是**setAttribute()和getAttribute()** 以及与session和cookie相关的操作

对于这一个part的内容你还可以再继续之前关于廖哥的servlet和web精进那个part

这里做一个简单的讲解就是：就大致的理解就是response和request其实就是每一次的http操作的一个基本单位，不就是请求和回应嘛，在spring是用HttpServlet开头HttpServletResponse、HttpServletRequest的类来封装它们的。然后他们两其实日常最常见的操作就是**setAttribute()和getAttribute()** 以及与session和cookie相关的操作，操作就是规定性的应用，很简单的，会用就好了。是全局性的，你个http请求还搞局部的？

session也是用HttpSession封装了类。然后三个类绝大多数都是直接作为参数进去用，然后调用相应Api即可

CRUD中：

1. 用登录控制器 session.setAttribute("loginUser",username); 来添加session，廖哥也是说放个user进去【用HttpSession session存session】

1. 在登录拦截器中用Object user = request.getSession().getAttribute("loginUser"); 取出user，用request来操作session【用HttpServletRequest request取session】

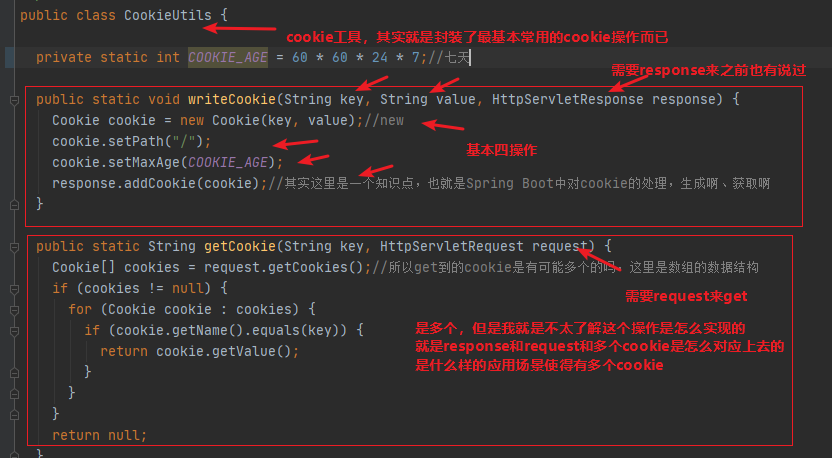
还有request.setAttribute("msg","没有权限请先登录"); 来带一个属性映射请求发送出去，给html用

还有 request.getRequestDispatcher("/index.html").forward(request,response);来实现一个转发

--同样的其实是可以通过request来存session，实质都是request.getSession()得到session之后再进行set和get操作。

req.getSession().setAttribute("user", name);

同样图书管理系统里面的cookie也是跟廖哥教程讲的同样的操作（session也是廖哥session里面的操作，基本上还一模一样）



廖哥教程：





**这里可以让你理解一下多个cookie是怎么回事**

Cookie 实际上是存储在客户端上的文本信息，并保留了各种跟踪的信息。

**Cookie 工作步骤：**

客户端**请求**服务器，如果**服务器**需要记录该用户的状态，就是**用 response** **向客户端浏览器**颁发一个 Cookie。并且客户端浏览器会把 Cookie 保存下来。（应该是有浏览器端的代码，你知道会保存即可，封装思想）

当浏览器**再请求**该网站时，浏览器把**该请求的网址连同 Cookie** 一同提交给服务器。服务器检查该 Cookie，以此来辨认用户状态。

（所以cookie是针对于特定网站的，是多个的，看你请求和你行为的需求。）

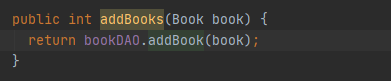
#### 6.2 Dao层的类



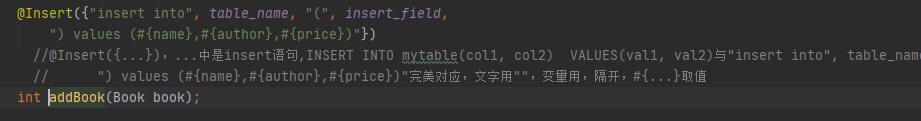
所以这里也是mybatis对我们的封装吗？就是它会内部直接利用这些SQL语句去和数据库打交道，至于怎么打的，你就不用管，你也不用管他是class还是interface

所以Dao层（Mybatis）就是提供操作数据库的Api的，你目前指导这个就好了

但是你会不会疑惑



这个int到底返回什么



进入Dao层是这样的代码，那么这个interface到底返回了什么？很疑惑

#### 6.3 发现了这个项目里面的很多内容是需要自己去补充的

例如它就没写查，但是在readme中有给到根据id、书名等查找书籍的方法，其实实现不难，就是一个SECLECT \* FROM table\_name WHERE id = /name= 等等

## 功能模块设计与实现

## 困难点和解决方案

## 改进点和改进思路

## 畅想和展望