

2023 威软后端 大作业

1.写在前面:

首先,祝大家新年快乐!

经过了不平凡的2022年,我们终于迎来了充满希望的2023年,希望在新的一年里我们可以与大家共同成长,充满希望,勇往直前.

相信同学们已经经过了前两次JAVA作业和一次SpringBoot作业的考验,掌握了Java语言的基本语法和一特性,了解了SpringBoot项目的运行方式。下面我们即将迎来最后一次大作业.

2.作业介绍:

2.1 引入:

回首刚刚度过的高中时光,同学们是否有过这样的经历?

笔者记得自己在高中的时候,在枯燥的学习生活之余,为数不多的期待就是在放学吃饭和一周一度的活动课的时候,学校里会想起久违而难得的流行音乐,丰富了我们日以继夜刷题的日子.

但是有一个问题就是每个人的口味各不相同,但是每个人对那音乐的渴望却是无二的,那要怎么满足同学们的听歌需求呢?我们设想了这样一种解决方案:线上投票.

由学生们放假回家休息的时候,选取自己喜欢的歌进行投票,假期结束后由学校的工作人员获取歌曲的投票名单,进行审核后.按照顺序在学校内播放.

2.2 功能介绍:

2.2.1 学生端:

2.2.1.1: 注册:

需要提供姓名, 学号, 班级, 密码等必要信息.

学号不可重复, 如果重复则注册失败, 需要重新注册.

2.2.1.2: 登陆:

我们本次项目只需要提供一套运行在服务器的后端系统, 采用MVC的前后端分离模式进行开发, 并不需要前端页面, 但是还需要提供一个登陆接口, 用于验证用户身份并返回一些必要的信息.

加分项1: shiro和jwt:

shiro是一个java的安全框架,用于身份验证,授权,密码,会话管理等功能.能储存用户的登录凭证:session等,同时使用过滤器(filter).对每个http请求进行过滤(根据接口和权限等信息限制访问).

jwt是json web token的缩写,是目前最流行的跨域身份认证解决方案.本质上是一个特殊的字符串.

感兴趣的同学们可以了解一下这两个技术, 也可以只使用其中的一个来解决问题.

这个问题可能对初学者来说有亿点点复杂, 在这里推荐几篇文章供大家参考:

[\(161条消息\) 一篇适合小白的Shiro教程-潮汐先生的博客-CSDN博客shiro](#)

(161条消息).JWT详解baobao555#的博客-CSDN博客jwt

也在这里推荐一下笔者的博客关于这方面的内容（偷偷的）：[Springboot笔记 | 抒情小熊 \(bearslyricattack.github.io\)](#)，这里面也有很多其他的关于SpringBoot的内容供大家参考~

本次加分项只需要使用这两个技术实现权限的验证即可。

2.2.1.3: 增添歌曲:

需要填写的有歌曲的名称，歌手，播放平台（防止学校的工作人员可能找不到这首歌0.0），备注等。

2.2.1.4: 查看可投票歌曲:

可投票歌曲指的是被增添进数据库,并且没有播放过的歌曲（假设一个歌曲被采用后播放一遍后不会再播放第二遍）。

这个功能是获取这些歌曲的列表,方便后续进行歌曲的投票。

2.2.1.5: 投票:

对歌曲列表中的歌曲进行投票。

加分项2: 投票限制:

现实中我们经历过投票的都知道，很多投票都是有限制的，比如每个人每天只能投一票或者说只能在某个时间段内投票或者在某个时间点后无法进行投票等。

本次加分项要求以任意的方式对同学们的投票进行限制即可。

亿点点提示：次数的限制可以在数据库中增添一个或几个字段来解决，也可以使用中间表来解决。而时间问题可以使用Java中的Date类，Calendar类和DateFormat类辅助解决。

2.2.1.6: 查看投票结果:

查看所有可投票歌曲的投票结果，让同学们可以先听几遍即将播放的歌，然后回学校听到时可以一展歌喉（笑）。

加分项3: 投票排序:

按照投票数的多少对歌曲进行投票并展示。可以使用SQL语句实现，同学们可以学习一下SQL的相关知识。

加分项要求按照任意顺序对投票结果继续投票并展示即可。

加分项4: 展示分页:

当查询的结果非常多的时候，一次性把大量的数据传给前端会造成前端压力过大。这种时候就需要使用分页功能，把数据分成很多页的，一页一页的传送到前端。

可以使用JPA Pageable实现歌曲投票列表的分页。

2.2.2 教师端:

2.2.2.1: 注册:

教师注册的时候需要填写自己的工号而不是学号。工号与学号一样不可重复。

2.2.2.2: 登陆:

2.2.2.3: 查看投票结果:

这两个的接口与学生端的要求基本相同。

2.2.2.4: 审批歌曲:

对每首歌曲单独进行审批, 只有审批通过的歌曲才可以播出。审批失败的歌曲无法播放。

2.2.2.5: 修改歌曲的播出状态:

在歌曲播出后, 修改该歌曲的状态为“已经播放”, 在学生端-查看已投票歌曲中展示时就不再展示。

2.2.2.6: 查询歌曲:

输入某首歌曲的名称, 查询某首歌曲的票数。

加分项5: 模糊查询:

大家平常在搜索信息的时候, 通常不需要输入完整的关键词, 就能查询到相应的信息。

本加分项要求以任意的方式实现歌曲模糊查询的功能。

2.2.2.7: 删除歌曲:

删除,顾名思义,就是去掉一些东西。

但是在我们真正的业务场景中,我们一般使用的是假删除。

首先讲一下什么叫真删除和假删除:

真删除:

指的就是彻底地删除, 从数据库表内将数据 进行移除 delete 。就是删除了数据库某一张表中的某一条记录,不但前端访问数据库的时候得不到这些数据,后台访问数据库的时候也看不到这条记录了。

假删除:

指的就是逻辑上的删除, 数据库表内, 数据会包含一个标识flag字段, 例如: status (删除标识, 0代表未删除, 1代表删除, 默认为0), 执行假删时, 只是将数据的删除标识 status从 0 改 1, update。本质上是数据的更新.后台访问数据库的时候仍然可以访问这写条数据,只不过前端访问接口的时候得不到这些数据了,相当于这些数据在用户的“眼”中被删除了.但后台访问数据库的时候仍然能看到这条数据。本次要求使用假删除的方式删除歌曲。

2.2.3: 其他加分项:

2.2.3.1: 个人收藏歌单:

学生和教师可以建立自己的个人收藏歌单, 里面包含了多首歌曲, 可以实现对个人收藏歌单的各种操作, 比如建立, 增加, 删除等。

维护歌曲与个人收藏歌单之间的关系需要可能用到数据库设计中的一对一, 一对多, 多对多, 可以使用jpa中的相对应注解实现, 也可以手写中间表实现。

2.2.3.2: 反射与AOP:

反射是Java的一种高级特性，是框架设计的灵魂。

JAVA反射机制是在运行状态中，对于任意一个类，都能够知道这个类的所有属性和方法；对于任意一个对象，都能够调用它的任意一个方法和属性。

AOP为Aspect Oriented Programming的缩写，意为：面向切面编程，通过预编译方式和运行期动态代理实现程序功能的统一维护的一种技术。

反射是AOP的基础，利于反射和AOP可以做到很多事情，比如统一监控接口的运行情况，在控制台打印程序运行日志等等。

尝试在项目中使用反射和AOP实现任意的功能,手写或者使用注解均可，有时候可以使你的程序更简洁，并在出现bug时更加容易排错。

2.2.3.3: 批量操作:

批量操作可以减少重复性劳动,大大提高我们的工作效率.

要求实现批量的修改歌曲的播放状态,批量的删除数据库中的歌曲等.

批量删除也跟普通的删除一样,要求使用假删除.即通过改变数据库中的表征删除状态的字段来删除数据.

2.2.3.4: 单元测试:

单元测试指的是对某一部分功能进行单独的测试。以防止在程序出现错误时不知道从何查起的问题。

一个合格的程序员必须熟练的掌握单元测试！可以在程序中对特定的组件和逻辑使用单元测试。

2.2.3.5: 前端页面（可能需要较多时间）：

使用前端页面可以更好的完成整个项目，更好的理解前后端交互的整个过程，需要学习一部分HTML,CSS,JavaScript的知识，可以参考前端的几次作业。

如果你认为这个项目还有哪些不足或者是你认为它有哪些欠缺的功能，也可以发挥你的创造力尽情的增添。

3.写在最后:

3.1 歌曲审批状态流程:

歌曲的审批状态应该是

- 1.增添歌曲,此时歌曲状态为未审批
- 2.进行歌曲审批,审批有审批通过和审批未通过两种情况,只有审批通过的歌曲能够播放
- 3.已经播放

再次提示一下,这种一个对象的多种不同的状态可以通过在数据库中增添一个专用的,表征状态的字段来实现.

当然了也可以使用合理的任意其他方式实现.

3.2 总结:

作业乍一看可能有点复杂，不要害怕！，一点一点来，每天做一点可能不知不觉之间就完成了（0.0）。

作业自发布那一天开始,尽量推荐在一个月之内提交,当然了如果没有完成后续提交也是可以的.

作业推荐使用JPA实现,其他任意合理的实现方式也可以.

SpringBoot的学习再次推荐一下b站黑马程序员的视频,本次作业和里面的加分项所用到的相关技术则可以去CSDN等平台上搜索相关文章进行学习。

大家如果在技术上或者这个项目的业务逻辑上有任何问题,都可以找群里的学长学姐讨论,祝大家都能有所收获,有所成长.