音乐网站

第一节

讲了这个音乐网站可以做什么

第二节

系统配置

系统开发 JDK + Windows + Maven

开发语言 JavaEE + vue

后端框架Springboot

前端Vue

数据库和工具Mysql + Navicat

开发工具 Idea VSCode

游览器Microsoft Edge

系统搭建步骤

1创建后台springboot并启动测试

2导入并编译前端代码vue：网站和后台管理

编译：npm install

试运行：npm run dev

3设计数据库

4正式开发

第三节

1创建Springboot项目

2配置阿里云镜像

3pom文件中依赖导入

4创建数据库

5application.properties中配置：端口号，连接mysql

第四节

music-manage搭建

music-client搭建

第五节 第六节

新建数据库中的表 完善表结构

管理员 admin

前端用户表 consumer

歌曲 song

歌手 singer

歌单列表 song\_list

歌单包含的歌曲列表 list\_song

收藏表 collect

评论表 comment up评论点赞数

评价表 rank\_list

第七节

在VScode中添加add folder to workspace 两个前端（music-manage music-client）

music-manage中：

package.json中可以运行哪些

assets资源文件夹

Login.vue 登录 更改登录背景

第八节

music-manage中：

登录窗创建

v-model 双向绑定

placeholder 没有内容时会提示的信息

type="password" 隐藏密码

rules 规则

submitForm 提交

css的配置

第九节

在后端

解决跨域问题config中WebMvcConfig中addCorsMappings

第十节

和数据库的对应关系domain

数据库的接口dao

建一些xml文件写sql mapper

实现service

controller

1先写domain中的类 传输还需要实现序列化

2然后写对应的dao 中的@Repository是让service很容易访问

3然后再去mapper中写一个xml实现它

4然后再去写service（管理员接口）再写一个类实现接口

5然后在controller中新建一个类

@Autowired用于自动注入 AdminServiceImpl 类的实例，这样就可以在控制器中使用该服务类的方法。

@RequestMapping返回的地址 提交方式method

@RequestMapping(value = "/admin/login/status", method = RequestMethod.POST)

6. constant工具类 ==unilt（它）

第十一课

music-manage中：

1访问后台

api中request.ts（工具类）—超时时间最多5秒 运行跨域 响应头 …

url在vue.config.js中NODE\_HOST

2响应拦截器

封装get post … 方法

3调用axios帮我们处理请求

4Promise异步操作

5再创建一个index.ts（实现类）

=>执行

第十二节 实现登录跳转

1在后端中

2[YinMusicApplication中 @MapperScan("com.example.yin.dao")](mailto:YinMusicApplication中@MapperScan(%22com.example.yin.dao%22)) 扫描这个包

3application.properties中配mybatis和 时间

1music-manage中：

2mixins是工具类

3然后登录后能跳转到首页

4在views中创建InfoPage.vue（首页）-- router（路由）

5中Login.vue中如果登录成功跳转到首页

if (result.success) routerManager(RouterName.Info, { path: RouterName.Info });

第十三节

1music-manage中：

2在views的home中引入YinHeade、YinAudio、YinAside 头，左侧菜单，中间部分

/components/layouts/YinHeader.vue

/components/layouts/YinAside.vue

/components/layouts/YinAudio.vue

第十四节 左侧菜单

1/components/layouts/YinAside.vue中 先写左侧菜单部分

2<div class="sidebar"> 左侧 位置设置

3<style scoped> scoped只用于当前 样式设置

4index="consumer"> 路由指向

5系统首页info 用户管理consumer 歌手管理singer 歌单管理songList

然后在对应views中创建SingerPage.vue 歌手页面等对应的Page

6在template中写了class，如<div class="sidebar">，就要在style中写对应的sidebar样式

7. :collapse="collapse" 下面写了对应的collapse 意思是菜单是否收缩—不收缩

第十五节 头 缩放菜单

/components/layouts/YinHeader.vue

1. class="header"对应在下面写header样式
2. 折叠图片class="collapse-btn"也是对应在下面写样式
3. 头与菜单栏的通信用

第十六节 全屏

1<div class="logo">{{ nusicName }}</div>

也是一样有class对应写logo的样式，然后nusicName在enums中music-name.ts

2尝试添加全屏，但未成功，因为写法不一样

第十七节 下拉框 退出登录

头像未找到怎么改

第十八节

调整内容位置

跟着左移

第十九节

歌手管理 同第十节

1写了domain

第二十节

2在dao中增改删查

注意查完—与下面返回的列表相对应

第二十一节

3mapper\SingerMapper.xml

写对应dao中的的增改删查 mysql语句

第二十二节

4写service中的接口

5写service中impl中的实现类

第二十三节

6在controller层写歌手控制类

写对应dao中的的增改删查 java语句

第二十四节

在前端music-manage中实现

7views中SingerPage.vue

添加歌手

第二十五节

歌手信息：歌手名、性别、故乡、歌手简介

添加 确定 取消

8在src\api\index.ts中添加歌手

views中SingerPage.vue中写

第二十六节 歌手图片

9在src\api\index.ts中添加查询歌手

views中SingerPage.vue中写

10在src\store\index.ts新建一个存储，

如userPic: "/img/avatorImages/user.jpg",

第二十七节 能够在前端获取到后端的图片

在后端中的config中写一个重定向地址的WebCharacterEncodingFilter

第二十八节 新增列

views中SingerPage.vue中写

el-table-column --列

新增歌手这一列

新增性别这一列

新增地区这一列

新增简介这一列

overflow: scroll 如果超出了就出现一个下拉

第二十九节 更新歌手图片

1前端提交和显示部分

2后台真正的往数据库里更新这条数据，还有把图片插入到目录

在后台

\controller\SingerController.java 添加一个更新歌手头像

@RequestMapping 上传文件

第三十节 实现了头像上传功能

在前端 在歌手图片下面加一个上传的功能 <el-upload

:action="uploadUrl(scope.row.id)"执行这个函数上传

:on-success="handleImgSuccess" 成功之后执行这个函数

:before-upload="beforeImgUpload" 上传之前执行这个函数

第三十一节 加一搜索框（输入框的内容，搜索得到的内容）

实现筛选歌手

v-model="searchWord"

第三十二节 实现翻页功能

1每页几个 当前是第几页（默认1） 计算当前表格中的数据如第一页 页-1+ pageSize

const pageSize = ref(5); // 页数

const currentPage = ref(1); // 当前页

2实现翻页

@current-change="handleCurrentChange" 写一个函数实现翻页

第三十三节 修改歌手信息 编辑功能

1添加操作这一列

2编辑 删除

@click="editRow(scope.row)">编辑

@click="deleteRow(scope.row.id)">删除

第三十四节

单个删除 批量删除

单个删除

<el-button type="danger" @click="deleteRow(scope.row.id)">删除</el-button>

第三十五节

批量删除

1添加一个多选框

<el-table-column type="selection" width="40" align="center"></el-table-column>

2 @selection-change="handleSelectionChange"

定义handleSelectionChange函数 已经选择了哪些

3在前面加一个批量删除

<el-button @click="deleteAll">批量删除</el-button>

1. 写对应的deleteAll函数

<el-table height="665px" border size="small" :data="data" @selection-change="handleSelectionChange">

将这个改成了665 解决了它下拉框到底还不能完全查看歌手信息

第三十六节

添加一列歌手管理 超链接 跳转到 歌曲管理

1. 写了SongPage.vue
2. 在路由index.ts中添加
3. 返回SingerPage.vue跳转到歌曲管理

第三十七节 歌曲管理 song

同前面歌手管理一样（第十六节）从后端到前端

1写了domain

2在dao中增改删查

注意查完—与下面返回的列表相对应

3mapper\SingerMapper.xml

写对应dao中的的增改删查 mysql语句

第三十八节

4写service中的接口

5写service中impl中的实现类

6在controller层写歌手控制类

写对应dao中的的增改删查 java语句

第四十节 添加歌曲

前后端连接可以从前端歌曲管理中添加歌曲，在后端的song中可以查看到

第四十一节 查询歌曲

1在后端controller中添加一个查询

2在前端index中添加查询歌曲

第四十二节 编辑 删除

同41节 需要编辑功能的先在后端增加 然后才能在前端实现

第四十三节 歌词显示

让歌词一行一行的能在滚动条，滚动

第四十四节 更新歌曲图片

同41节 需要编辑功能的先在后端增加 然后才能在前端实现

第四十四节 更新歌曲

同41节 需要编辑功能的先在后端增加 然后才能在前端实现

第四十五节

第四十六节

第四十七节

播放一手歌曲MP3 实现

第四十八节

1写了domain

2在dao中增改删查

注意查完—与下面返回的列表相对应

3mapper\SingerMapper.xml

写对应dao中的的增改删查 mysql语句

第四十九节

4写service中的接口

5写service中impl中的实现类

第五十节

6在controller层写歌手控制类

写对应dao中的的增改删查 java语句

第五十一节 前端增删改查

7. SongListPage.vue

8. \api\index.ts

第五十二节 后端

同上流程 listsong

第五十三节 前端

第五十四节

解决删除

第五十五节 用户管理

1写了domain

第五十六节

2在dao中增改删查

注意查完—与下面返回的列表相对应

第五十七节

3mapper\SingerMapper.xml

写对应dao中的的增改删查 mysql语句

第五十八节

4写service中的接口

5写service中impl中的实现类

第五十九节

6在controller层写歌手控制类

写对应dao中的的增改删查 java语句

第六十节

没有编辑和添加功能，我们可以添加

第六十一节

添加校验 空的时候不可以添加

六十二节

数据的统计 --图



第六十三节 用户数怎么求



第六十四节



第六十五节

饼图 它绘制的更好看 --改一下

第六十六节 柱状图

第六十七节

饼图 它绘制的更好看 --改一下

第六十八节

柱状图

第六十九节

home中的轮番图

在\src\enums\swiper.ts

:interval="4000" 换图片的时间

遍历swiperList在\src\enums\swiper.ts中

<el-carousel-item v-for="(item, index) in swiperList" :key="index">

<img :src="item.picImg" />

第七十节

前台 头

music logo

在YinLoginLogo.vue中

对应music-name.ts

第七十一节 左侧菜单 跳转到其他页面

1在YinNav.vue中

2对应nav.ts

没有我的页面

解决了YinNav.vue标红的错误

这些错误消息表示TypeScript编译器无法推断出Vue组件中**styleList**数组的类型。它将其视为未知类型，该类型没有名为**'name'**的属性。通过显式指定**styleList**属性的类型，TypeScript编译器将理解数组中的每个项应该具有一个**name**属性，并且错误应该得到解决。

3路由配置

第七十二节 菜单高亮显示

store缓存工具

放在了缓存configure.ts中

第七十三节 home中添加歌曲歌手

第七十四节 home中添加歌曲歌手

把另外那个前端中的api中的内容复制过来

主要改改index.ts

然后在home中写

第七十五节 回到顶部

YinScrollTop.vue

const S = document.body.scrollTop || document.documentElement.scrollTop;

第七十六节 底部

YinFooter.vue

第七十七节 转向搜索

YinHeader.vue 搜索

对应search中的所有内容

第七十八节 搜索 tab切换

歌曲 歌单 相互切换



第七十九节

模糊查询（前后端）

然后api index

然后在mixins中的index中写

第八十节

在SearchSong.vue查看

第八十一节

第八十二节

歌曲搜索显示

ul 列表 li 列



第八十三节 播放歌曲

song.ts

第八十四节

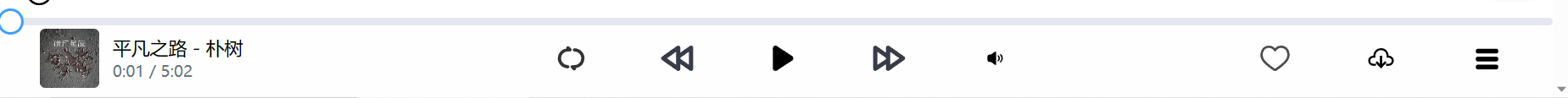
将song.ts放到YinAudio.vue

第八十五节 播放歌曲

第八十六节 播放歌曲

第八十七节 播放歌曲

YinPlayBar.vue



\utils\index.ts 工具 解析时间/歌词。。。

第八十八节 播放歌曲

第八十九节 播放歌曲

 实现这个播放功能