# 软件工程大作业-在线电影票购票系统

# 一、需求分析

## 1.1概述

为了帮助影院实现线上购票并提升销售额，我们可以开发一个在线电影票购票系统。该系统将包含一个用户友好的前端界面，使顾客能够方便地浏览电影信息、选择电影场次和座位，并进行支付；一个后台管理系统，用于管理电影排期、座位安排和订单处理。通过此系统，影院能够拓展销售渠道，提高客户体验，同时在竞争激烈的市场中保持竞争力。

## 1.2角色分析

### 1.2.1管理员：维护电影，维护影院，维护会员，维护订单，维护收藏，维护档期，管理员登录

### 1.2.2游客：浏览电影，浏览影院，注册会员，电影详情，影院详情

### 1.2.3会员：浏览电影，浏览影院，登录会员，电影详情，影院详情，查看订单，查看收藏列表，购票，收藏电影

## 1.3业务流程设计

### 1.3.1购票流程

### 1.3.2注册会员流程

### 1.3.3管理员维护电影流程

### 1.3.4管理员维护档期流程

### 1.3.5其他流程

## 1.4用例分析

### 1.4.1游客用例

#### 1.4.1.1会员注册用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编码 | A01 | 用例名称 | 会员注册 |
| 角色 | 游客 | 前置条件 | 无 |
| 基本流 | 1、**游客**输入网址  2、系统返回注册页面  3、游客输入账号，密码，确认密码，昵称，电话，邮箱  4、游客点击注册按钮  5、系统返回注册成功 | | |
| 异常流 | 1、游客输入账号重复，系统返回错误注册信息  2、网络异常，注册失败 | | |

#### 1.4.1.2电影列表用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编码 | A02 | 用例名称 | 电影列表 |
| 角色 | 游客 | 前置条件 | 无 |
| 基本流 | 1、游客输入网址  2、系统返回电影列表界面  3、游客通过标签筛选电影  4、系统返回符合要求的电影列表 | | |
| 异常流 | 1、不存在符合要求的电影，系统返回空列表  2、网络错误，系统返回空列表 | | |

#### 1.4.1.3影院列表用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编码 | A03 | 用例名称 | 影院列表 |
| 角色 | 游客 | 前置条件 | 无 |
| 基本流 | 1、游客输入网址  2、系统返回影院列表界面  3、游客通过标签筛选影院  4、系统返回符合要求的影院列表 | | |
| 异常流 | 1、不存在符合要求的影院，系统返回空列表  2、网络错误，系统返回空列表 | | |

#### 1.4.1.4电影详情用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编码 | A04 | 用例名称 | 电影详情 |
| 角色 | 游客 | 前置条件 | 电影列表 |
| 基本流 | 1、游客访问电影列表中某一电影  2、系统返回电影详情界面 | | |
| 异常流 | 网络异常，返回空信息 | | |

#### 1.4.1.5影院详情用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编码 | A05 | 用例名称 | 影院详情 |
| 角色 | 游客 | 前置条件 | 影院列表 |
| 基本流 | 1、游客访问影院列表中某一电影  2、系统返回影院详情界面 | | |
| 异常流 | 网络异常，返回空信息 | | |

### 1.4.2会员用例

#### 1.4.2.1购票用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编码 | A06 | 用例名称 | 购票 |
| 角色 | 会员 | 前置条件 | 影院详情 |
| 基本流 | 1、会员选择需要观看的电影  2、系统返回对应的电影档期安排界面  3、会员选择观影时间与座位  4、会员确认支付订单  5、系统返回购票成功 | | |
| 异常流 | 1、该座位已被别人购买、系统返回购票失败  2、网络异常，系统返回购买失败 | | |

#### 1.4.2.2收藏电影用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编码 | A07 | 用例名称 | 收藏电影 |
| 角色 | 会员 | 前置条件 | 电影详情 |
| 基本流 | 1、会员选择收藏该电影  2、系统返回收藏成功 | | |
| 异常流 | 网络异常，系统返回收藏失败 | | |

#### 1.4.2.3会员登录用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编码 | A08 | 用例名称 | 会员登录 |
| 角色 | 会员 | 前置条件 | 无 |
| 基本流 | 1、游客输入网址  2、系统返回会员登录界面  3、游客输入账号，密码  4、游客点击登录按钮  5、系统返回登录成功 | | |
| 异常流 | 1、账号或密码错误，系统返回登录失败  2、网络异常，系统返回登录失败 | | |

#### 1.4.2.4订单列表用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编码 | A09 | 用例名称 | 查看订单 |
| 角色 | 会员 | 前置条件 | 无 |
| 基本流 | 1、会员输入网址  2、系统返回订单列表界面 | | |
| 异常流 | 1、会员为登录、系统返回空  2、网络异常，系统返回空列表 | | |

#### 1.4.2.5收藏列表用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编码 | A10 | 用例名称 | 收藏列表 |
| 角色 | 会员 | 前置条件 | 无 |
| 基本流 | 1、会员输入网址  2、系统返回收藏列表界面  3、会员选择取消收藏  4、系统返回取消收藏成功 | | |
| 异常流 | 网络异常，系统返回空列表或取消收藏失败 | | |

### 1.4.3管理员用例

# 二、软件设计

## 2.1架构设计

### 2.1.1技术架构

### 2.1.2模块架构

## 2.2数据库设计

### 2.2.1ER图设计

### 2.2.2关系表设计

表名：T\_Movie\_Actor

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 宽度 | 是否外键 | 是否主键 | 备注 |
| 1 | movie\_id | int |  | 是 | 是 | 电影ID |
| 2 | actor\_id | int |  | 是 | 是 | 演员ID |
| 3 | role | varchar | 250 | 否 | 否 | 角色 |

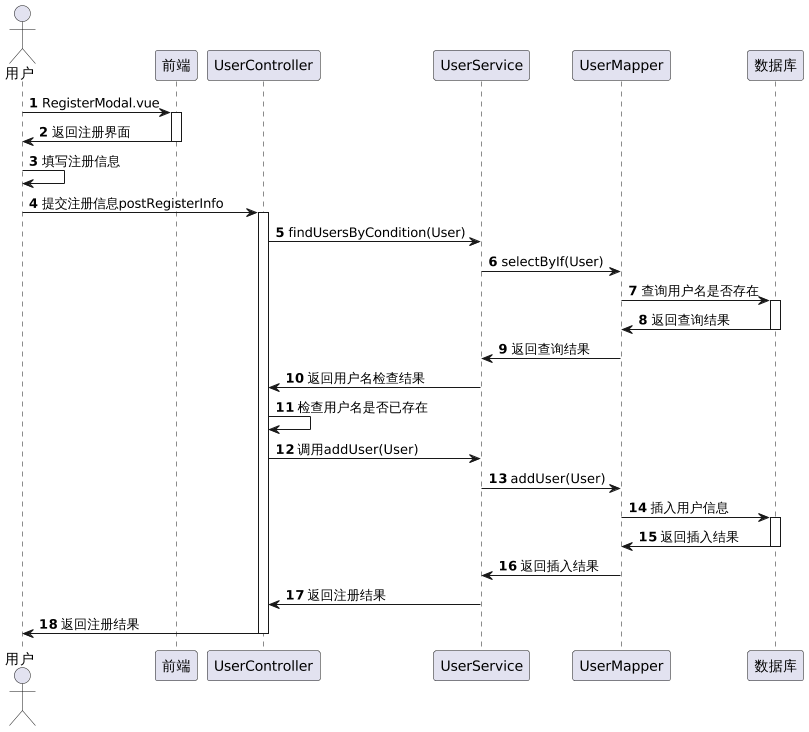
表名：T\_Collection

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 宽度 | 是否外键 | 是否主键 | 备注 |
| 1 | user\_id | int |  | 是 | 是 | 用户ID |
| 2 | movie\_id | int |  | 是 | 是 | 电影ID |
| 3 | collection\_date | datetime |  | 否 | 否 | 收藏时间 |

## 2.3用例设计

### A01"会员注册"用例设计

#### 顺序图设计



#### 类的分析

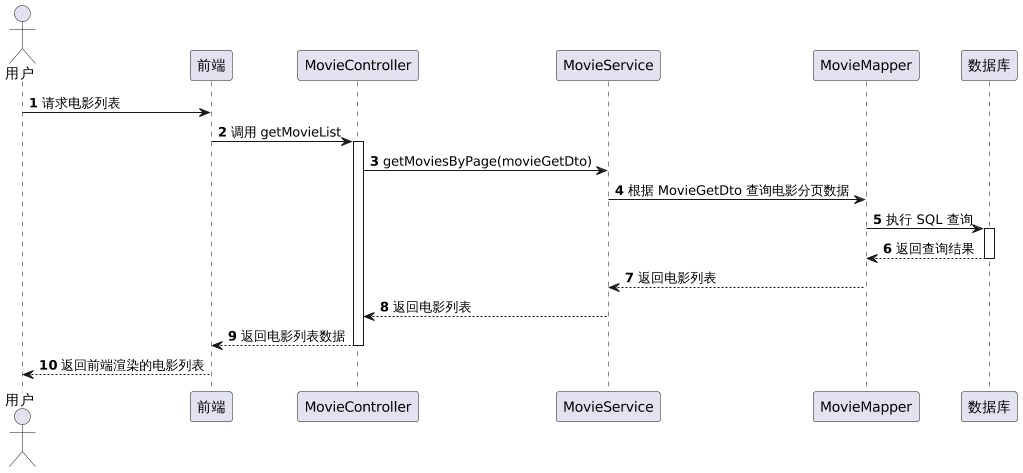


#### 接口分析

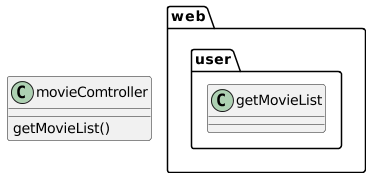
|  |  |
| --- | --- |
| 接口说明 | 用户注册 |
| 接口名字 | /register |
| 访问网址 | /api/user/register |
| 入参 | 头像，账号，密码，昵称，电话，邮箱 |
| 出参 | {  username:zx,  email:123,  phoneNumber:123,  nickname:zx, avatarUrl=<http://localhost:721/static/avatar/20241212115201.jpg)>  } |

### A02"电影列表"用例设计

#### 顺序图设计



#### 类的分析

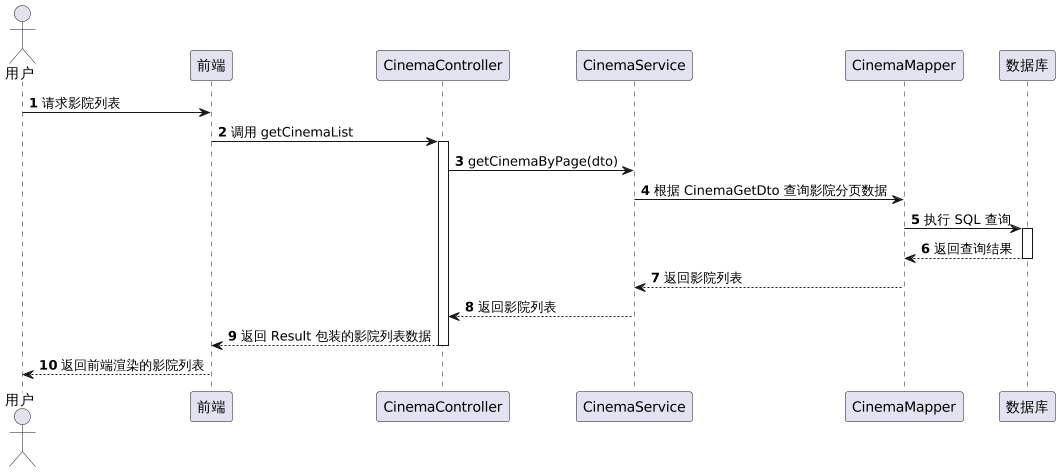


#### 接口分析

|  |  |
| --- | --- |
| 接口说明 | 电影列表 |
| 接口名字 | /list |
| 访问网址 | /api/movie/list |
| 入参 | 放映情况，类型，区域，年代，页码，数量 |
| 出参 | [  {  "movieId": 304,  "title": "噬魂岛",  "synopsis": "一艘观光游轮在一片诡异的海域失事，乘客们被冲上了一座地图上并不存在的岛屿。小岛弥漫着穿透身体的恐怖黑色团雾，在岛上的幸存游客，纷纷被自己恶梦中的恐怖鬼魂蛊惑，中邪发疯，更有游客莫名人间蒸发，死去的乘客突然消失。诡异的山洞里几具泛着磷光的枯骨，向幸存的人们揭示了一个更为恐怖绝望和匪夷所思的未来……",  "releaseDate": "2024-11-28T16:00:00.000+00:00",  "director": "魏冬雷",  "rating": 0,  "posterUrl": "[https://p0.pipi.cn/basicdata/54ecde51ddd5bfc696338fe4e640e4c562a4e.jpg?imageView2/1/w/464/h/644",](http://invalid.uri)  "productionRegion": "中国大陆",  "tags": "恐怖,惊悚",  "collectionCount": 0  },  .......  ] |

### A03"影院列表"用例设计

#### 顺序图设计



#### 类的分析

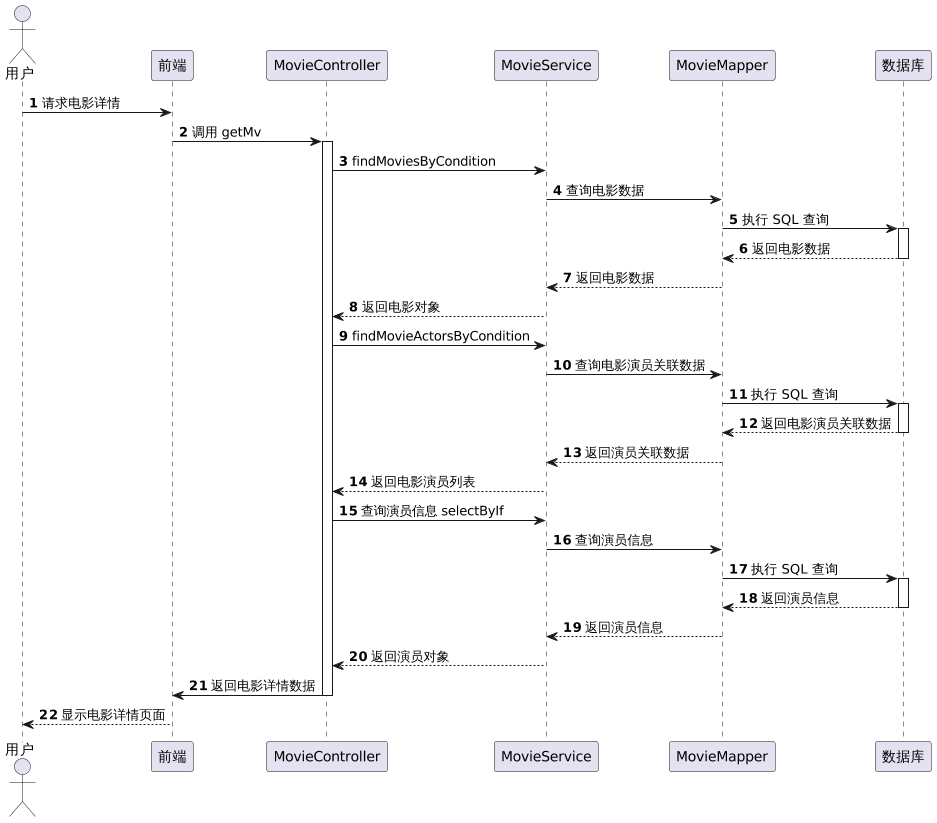


#### 接口分析

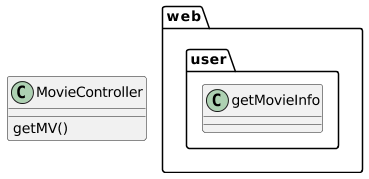
|  |  |
| --- | --- |
| 接口说明 | 影院列表 |
| 接口名字 | /list |
| 访问网址 | /api/cinema/list |
| 入参 | 电影Id，品牌，页码，数量 |
| 出参 | {  "code": 200,  "msg": "请求成功",  "data": [  {  "cinemaId": 101,  "name": "万全时代国际影城",  "location": "平阳县万全镇老农伯购物中心4楼",  "contactNumber": "18658764313"  },  {  "cinemaId": 102,  "name": "中影时代影城（水头店）",  "location": "平阳县水头镇江山东路曼顿商业中心2层",  "contactNumber": "0577-63866999"  },  .........  ]  } |

### A04"电影详情"用例设计

#### 顺序图设计



#### 类的分析

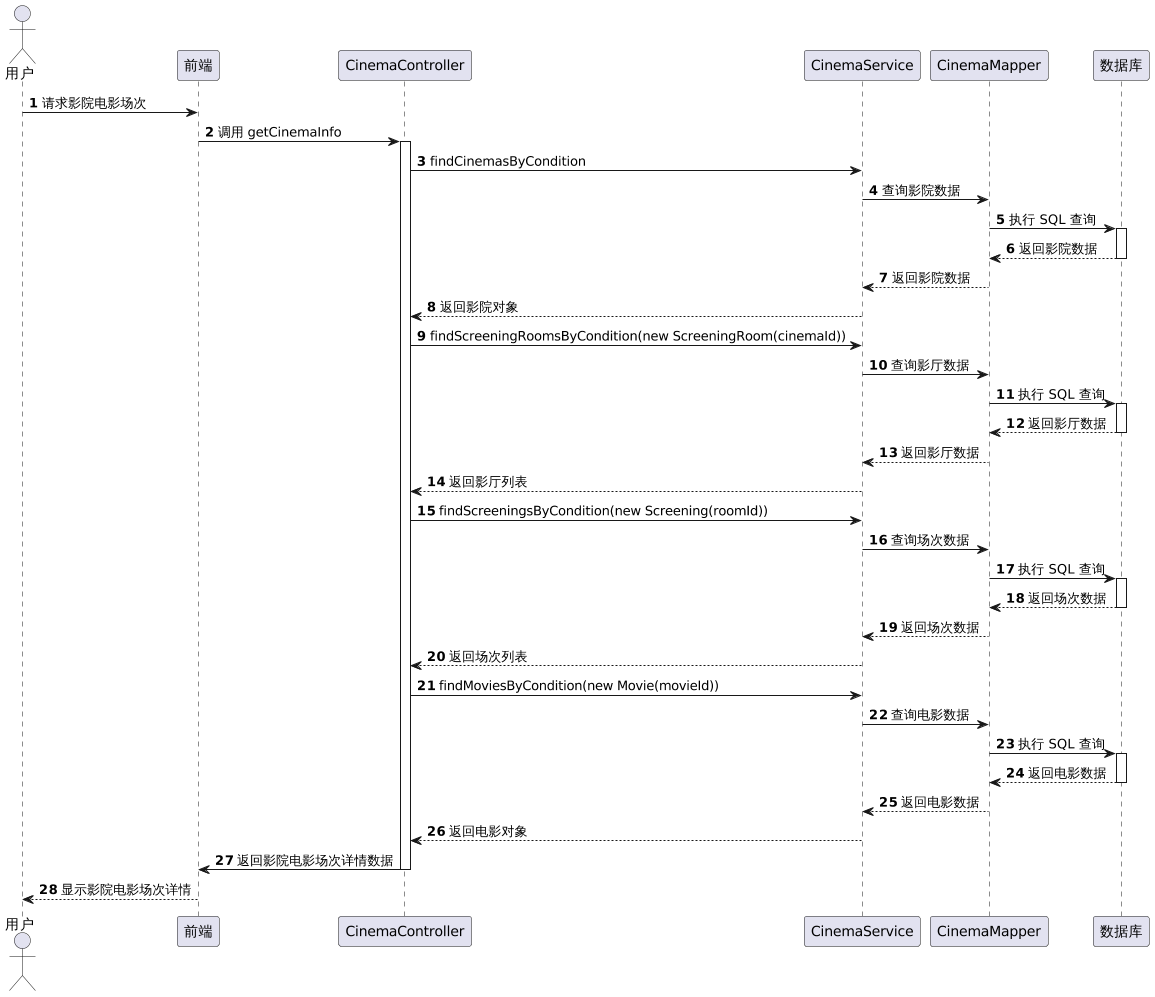


#### 接口分析

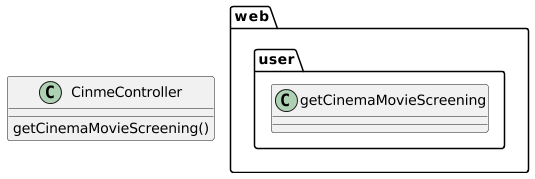
|  |  |
| --- | --- |
| 接口说明 | 电影详情 |
| 接口名字 | /getMv |
| 访问网址 | /api/movie/getMv |
| 入参 | 电影Id |
| 出参 | {  "code": 200,  "msg": "请求成功",  "data": {  "movie": {  "movieId": 1052,  "title": "好东西",  "synopsis": "爱逞强的单亲妈妈王铁梅（宋佳饰）带小孩王茉莉（曾慕梅饰）搬到新家，结识了所谓清醒恋爱脑的邻居小叶（钟楚曦饰）。两位性格迥异的女性，一个坚强，一个柔软，一个擅长给人当妈，一个擅长随时撒谎。面对旧创伤和新挑战，她们彼此温暖互相慰藉。而围绕王铁梅的两个男人，前夫（赵又廷饰）不时“添乱”，女儿的鼓手老师（章宇饰）似乎充满新的可能。作为已经觉醒的女人们和学习过性别议题的男人们，会遇到什么新问题？会如何看待自己和世界？",  "releaseDate": "2024-11-21 16:00",  "director": "邵艺辉",  "rating": 9.6,  "posterUrl": "[https://p0.pipi.cn/mediaplus/friday\_image\_fe/cdf05c5c77e52993b6a2190af833c8b65f9b.jpg?imageView2/1/w/464/h/644",](http://invalid.uri)  "productionRegion": "中国大陆",  "tags": "剧情 爱情",  "collectionCount": 2  },  "actors": [  {  "actor": {  "actorId": 1,  "actorName": "宋佳"  },  "role": "王铁梅"  },  {  "actor": {  "actorId": 2,  "actorName": "钟楚曦"  },  "role": "小叶"  },  ........  ]  }  } |

### A05"影院详情"用例设计

#### 顺序图设计



#### 类的分析

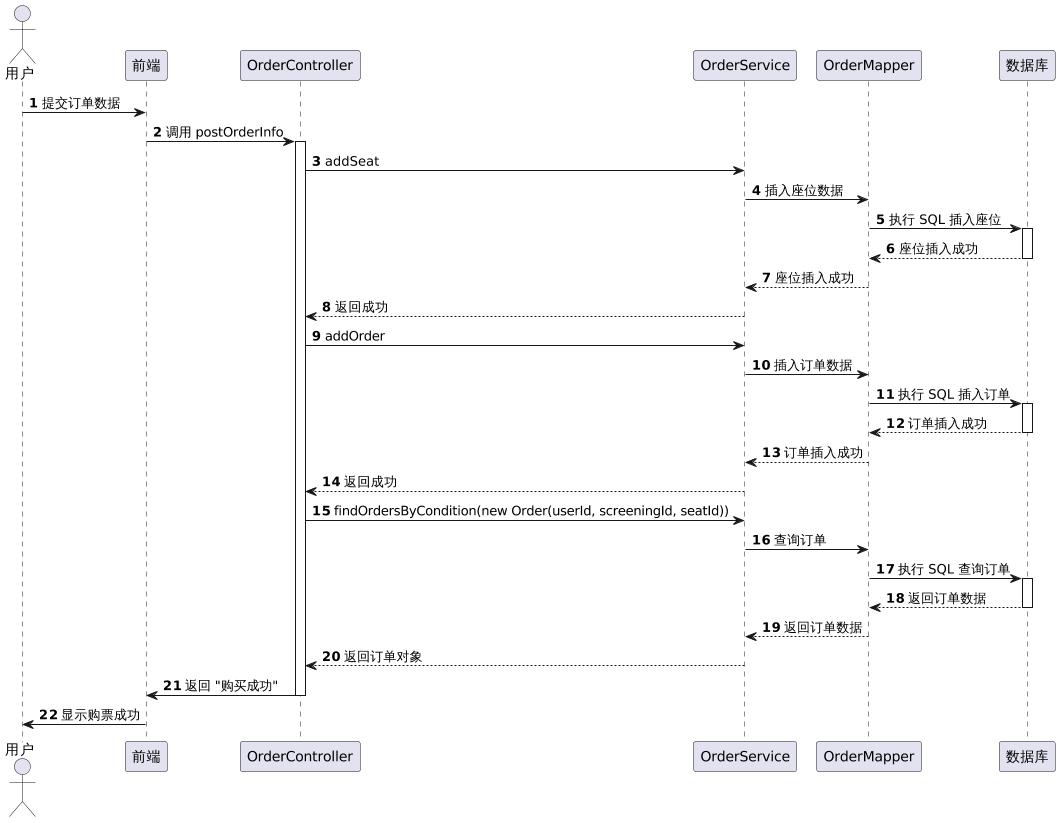


#### 接口分析

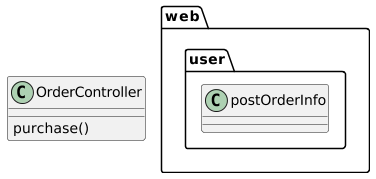
|  |  |
| --- | --- |
| 接口说明 | 影院详情 |
| 接口名字 | /screening |
| 访问网址 | /api/cinema/screening |
| 入参 | 影院Id |
| 出参 | {  "code": 200,  "msg": "请求成功",  "data": {  "cinema": {  "cinemaId": 136,  "name": "威尼斯影城",  "location": "瓯海区新桥街道六虹桥路1189号梦多多小镇C号楼4层",  "contactNumber": "0577-88166678"  },  "movieDetails": []  }  } |

### A06"购票"用例设计

#### 顺序图设计



#### 类的分析

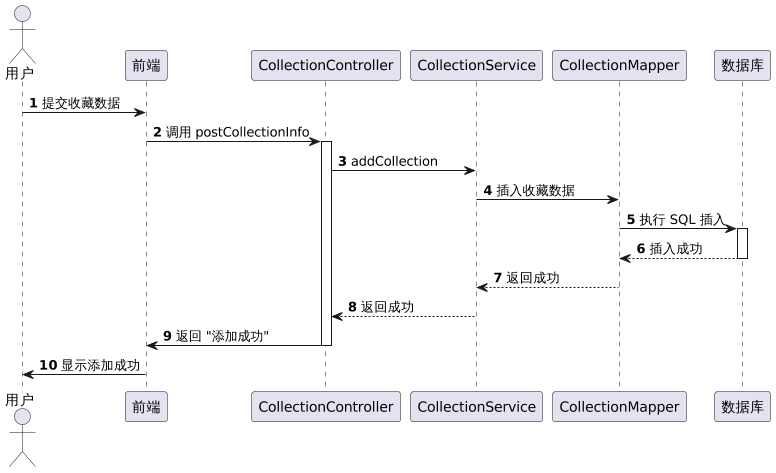


#### 接口分析

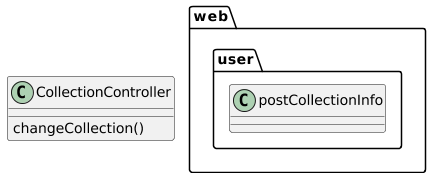
|  |  |
| --- | --- |
| 接口说明 | 购票 |
| 接口名字 | /changeOrder |
| 访问网址 | /api/order/changeOrder |
| 入参 | 用户Id，播映Id，座位号 |
| 出参 | {  "code": 200,  "msg": "购买成功",  "data": 38  } |

### A07"收藏电影"用例设计

#### 顺序图设计



#### 类的分析

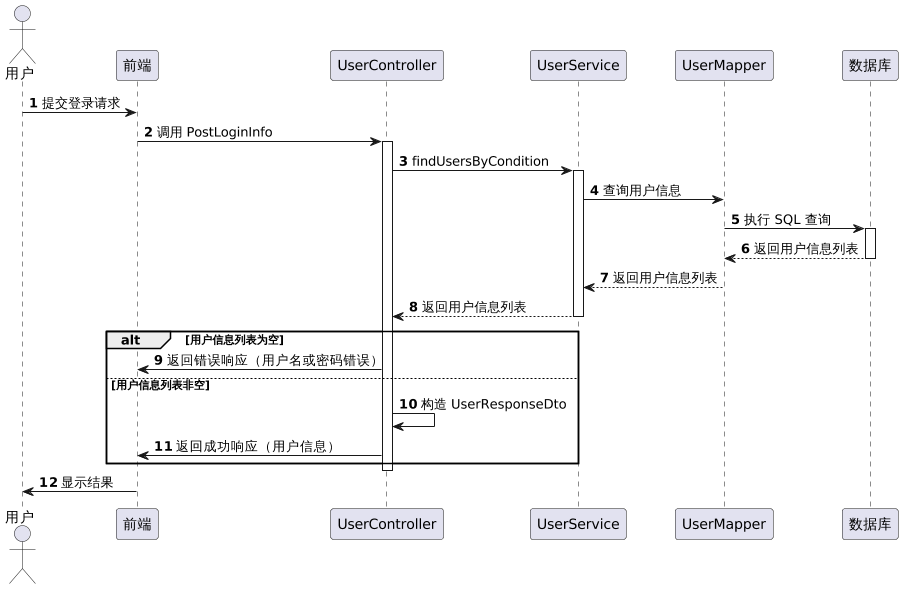


#### 接口分析

|  |  |
| --- | --- |
| 接口说明 | 收藏 |
| 接口名字 | /changeCollection |
| 访问网址 | /api/collection/changeCollection |
| 入参 | 用户ID，电影ID，收藏时间 |
| 出参 | {  "code": 200,  "msg": "添加成功",  "data": null  } |

### A08"会员登录"用例设计

#### 顺序图设计



#### 类的分析

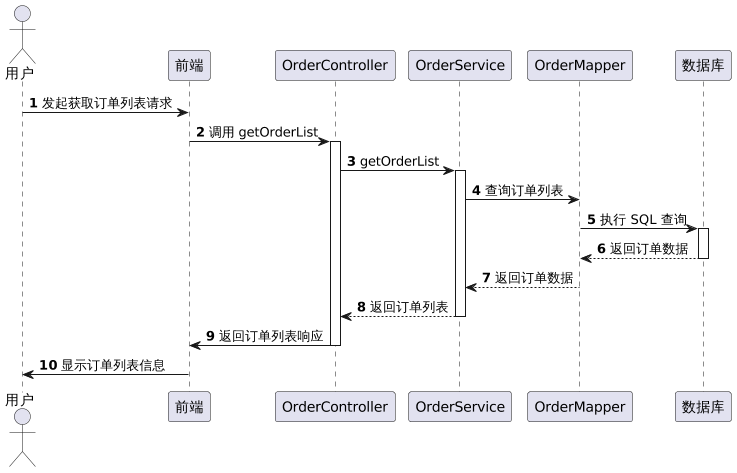


#### 接口分析

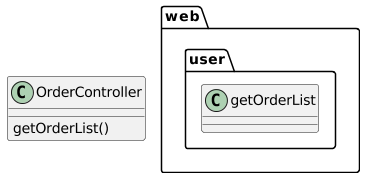
|  |  |
| --- | --- |
| 接口说明 | 会员登录 |
| 接口名字 | /login |
| 访问网址 | /api/user/login |
| 入参 | 账号，密码 |
| 出参 | {  "code": 200,  "msg": "登录成功",  "data": {  "userId": 11,  "username": "Onaffair",  "nickname": "Onaffair",  "avatarUrl": "[http://localhost:721/static/avatar/20241208174138.jpg",](http://invalid.uri)  "email": "123",  "phoneNumber": "123"  }  } |

### A09"订单列表"用例设计

#### 顺序图设计



#### 类的分析

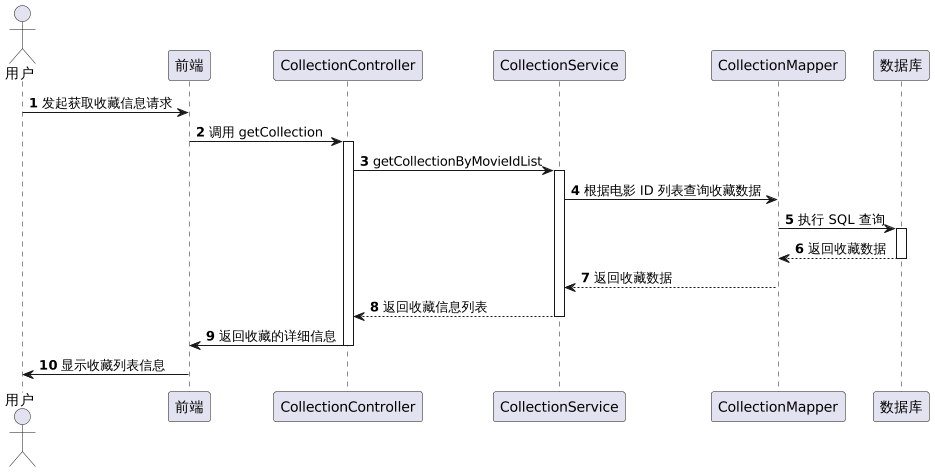


#### 接口分析

|  |  |
| --- | --- |
| 接口说明 | 订单列表 |
| 接口名字 | /orderList |
| 访问网址 | /api/order/orderList |
| 入参 | 订单ID |
| 出参 | {  "code": 200,  "msg": "请求成功",  "data": [  {  "orderId": 37,  "cinemaName": "万全时代国际影城",  "movieName": "蜡笔小新：我们的恐龙日记",  "price": 20,  "roomNumber": "2号厅",  "seatId": 45,  "screeningTime": "2024-12-07T10:00:00.000+00:00"  },  .....  ]  } |

### A10"收藏列表"用例设计

#### 顺序图设计



#### 类的分析



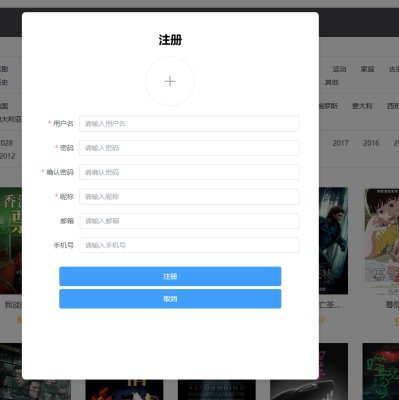
#### 接口分析

|  |  |
| --- | --- |
| 接口说明 | 收藏列表 |
| 接口名字 | /collectionList |
| 访问网址 | /api/collection/collectionList |
| 入参 | 电影ID |
| 出参 | {  "code": 200,  "msg": "请求成功",  "data": [  {  "movieId": 1052,  "movieName": "好东西",  "rating": 9.6,  "posterUrl": "[https://p0.pipi.cn/mediaplus/friday\_image\_fe/cdf05c5c77e52993b6a2190af833c8b65f9b.jpg?imageView2/1/w/464/h/644",](http://invalid.uri)  "collectionCount": 2  },  {  "movieId": 1056,  "movieName": "风流一代",  "rating": 0,  "posterUrl": "[https://p0.pipi.cn/mediaplus/friday\_image\_fe/cdf05c5cea414d5be715d53346e355b72ec78.jpeg?imageView2/1/w/464/h/644",](http://invalid.uri)  "collectionCount": 1  },  .......  ]  } |

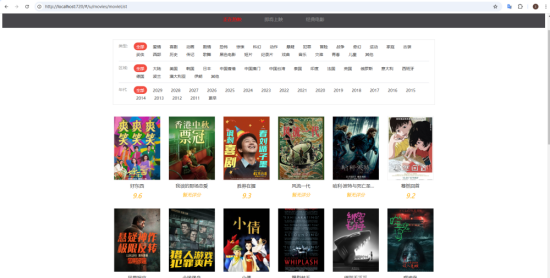
# 三、软件编码

## 3.1界面

### 会员注册



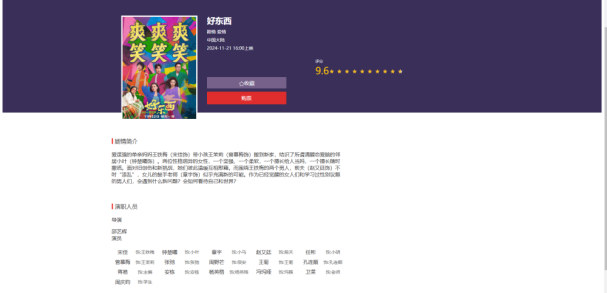
### 电影列表



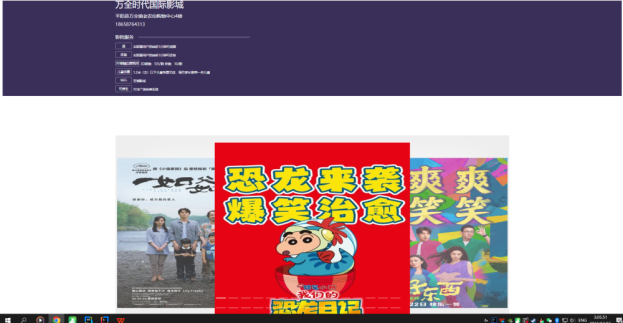
### 影院列表

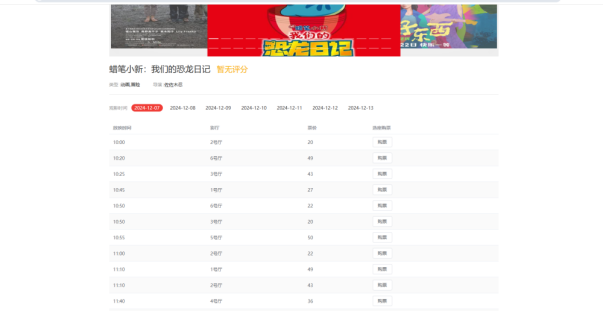


### 电影详情

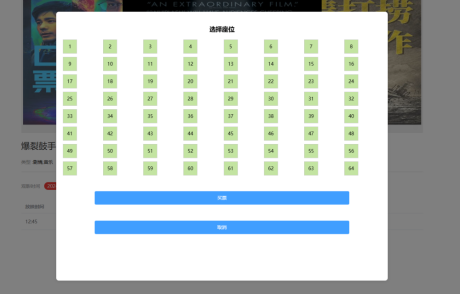


### 影院详情





### 购票



### 收藏电影



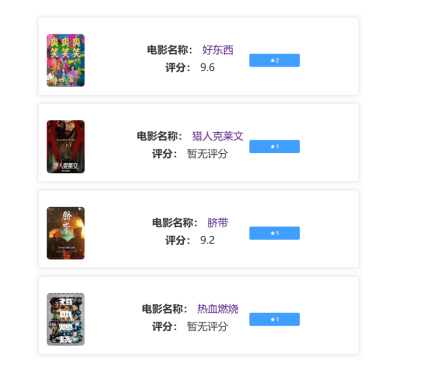
### 会员登录



### 订单列表



### 收藏列表



## 3.2架构编码

## 3.3数据库编码

create table actor  
(  
 actor\_id int auto\_increment  
 primary key,  
 actor\_name varchar(50) null  
);  
  
create table cinema  
(  
 cinema\_id int auto\_increment  
 primary key,  
 name varchar(100) null,  
 location varchar(255) null,  
 contact\_number varchar(3) null  
);  
  
create table movie  
(  
 movie\_id int auto\_increment  
 primary key,  
 title varchar(100) null,  
 synopsis text null,  
 release\_date date null,  
 director varchar(50) null,  
 rating decimal(2, 1) null,  
 poster\_url varchar(255) null,  
 production\_region varchar(50) null,  
 tags varchar(100) null,  
 collection\_count int default 0 null  
);  
  
create table movie\_actor  
(  
 movie\_id int not null,  
 actor\_id int not null,  
 role varchar(250) null,  
 primary key (movie\_id, actor\_id),  
 constraint movie\_actor\_ibfk\_1  
 foreign key (movie\_id) references movie (movie\_id),  
 constraint movie\_actor\_ibfk\_2  
 foreign key (actor\_id) references actor (actor\_id)  
);  
  
create index cast\_id  
 on movie\_actor (actor\_id);  
  
create table screeningroom  
(  
 room\_id int auto\_increment  
 primary key,  
 cinema\_id int null,  
 room\_number varchar(40) null,  
 seat\_count int null,  
 constraint screeningroom\_ibfk\_1  
 foreign key (cinema\_id) references cinema (cinema\_id)  
);  
  
create table screening  
(  
 screening\_id int auto\_increment  
 primary key,  
 movie\_id int null,  
 room\_id int null,  
 screening\_time datetime null,  
 price decimal(10, 2) null,  
 constraint screening\_ibfk\_1  
 foreign key (movie\_id) references movie (movie\_id),  
 constraint screening\_ibfk\_2  
 foreign key (room\_id) references screeningroom (room\_id)  
);  
  
create index movie\_id  
 on screening (movie\_id);  
  
create index room\_id  
 on screening (room\_id);  
  
create index cinema\_id  
 on screeningroom (cinema\_id);  
  
create table seat  
(  
 seat\_id int not null,  
 screening\_id int not null,  
 primary key (screening\_id, seat\_id),  
 constraint seat\_ibfk\_1  
 foreign key (screening\_id) references screening (screening\_id)  
);  
  
create index seat\_id\_index  
 on seat (seat\_id);  
  
create table user  
(  
 user\_id int auto\_increment  
 primary key,  
 username varchar(50) null,  
 password varchar(100) null,  
 email varchar(100) null,  
 phone\_number varchar(20) null,  
 nickname varchar(50) null,  
 avatar\_url varchar(255) null  
);  
  
create table collection  
(  
 user\_id int not null,  
 movie\_id int not null,  
 collection\_date datetime default CURRENT\_TIMESTAMP null,  
 primary key (user\_id, movie\_id),  
 constraint collection\_ibfk\_1  
 foreign key (user\_id) references user (user\_id),  
 constraint collection\_ibfk\_2  
 foreign key (movie\_id) references movie (movie\_id)  
);  
  
create index movie\_id  
 on collection (movie\_id);  
  
create definer = root@localhost trigger update\_movie\_collection\_count\_after\_delete  
 after delete  
 on collection  
 for each row  
BEGIN  
 -- 更新 movie 表中的 collection\_count 字段，统计每个 movie 的收藏数  
 UPDATE movie  
 SET collection\_count = (  
 SELECT COUNT(\*)  
 FROM collection  
 WHERE movie\_id = OLD.movie\_id  
 )  
 WHERE movie\_id = OLD.movie\_id;  
END;  
  
create definer = root@localhost trigger update\_movie\_collection\_count\_after\_insert  
 after insert  
 on collection  
 for each row  
BEGIN  
 -- 更新 movie 表中的 collection\_count 字段，统计每个 movie 的收藏数  
 UPDATE movie  
 SET collection\_count = (  
 SELECT COUNT(\*)  
 FROM collection  
 WHERE movie\_id = NEW.movie\_id  
 )  
 WHERE movie\_id = NEW.movie\_id;  
END;  
  
create table `order`  
(  
 order\_id int auto\_increment  
 primary key,  
 user\_id int null,  
 screening\_id int null,  
 seat\_id int null,  
 constraint order\_ibfk\_1  
 foreign key (user\_id) references user (user\_id),  
 constraint order\_ibfk\_3  
 foreign key (screening\_id, seat\_id) references seat (screening\_id, seat\_id)  
);  
  
create index order\_ibfk\_2  
 on `order` (seat\_id);  
  
create index screening\_id  
 on `order` (screening\_id);  
  
create index user\_id  
 on `order` (user\_id);

## 3.4用例编码

### 会员注册

#### 第一步：接口编程

@PostMapping("/register")  
 @ResponseBody  
 public Result<UserResponseDto> register(@RequestBody UserRegisterDto userRegisterDto){  
  
 User user = new User();  
 user.setUsername(userRegisterDto.getUsername());  
 List<User> targets = userService.findUsersByCondition(user);  
 if (!targets.isEmpty()){  
 return Result.error(ResultCode.VALIDATION\_ERROR.getCode(),"用户名已存在");  
 }  
 user.setPassword(EncryptionUtil.encodeBase64(userRegisterDto.getPassword()));  
 user.setEmail(userRegisterDto.getEmail());  
 user.setPhoneNumber(userRegisterDto.getPhone());  
 user.setNickname(userRegisterDto.getNickname());  
 user.setAvatarUrl(userRegisterDto.getAvatar());  
 System.out.println("user:" + user);  
 boolean flag = userService.addUser(user);  
  
 if (!flag){  
 return Result.error(ResultCode.ERROR.getCode(),"注册失败");  
 }  
 targets = userService.findUsersByCondition(user);  
 User target = targets.get(0);  
 System.out.println("target:"+target);  
 UserResponseDto res = new UserResponseDto(  
 target.getUserId(),  
 target.getUsername(),  
 target.getNickname(),  
 target.getAvatarUrl(),  
 target.getEmail(),  
 target.getPhoneNumber()  
 );  
 return Result.success(ResultCode.SUCCESS.getCode(),"注册成功",res);  
  
 }

#### 第二步：前端编码

<script setup>  
/\* eslint-disable \*/  
import {ref, reactive, defineEmits} from "vue";  
import {Plus} from "@element-plus/icons-vue";  
import {postAvatar, postRegisterInfo} from "@/api";  
import {ElMessage} from "element-plus";  
import {useUserStore} from "@/store/userStore";  
  
const emit = defineEmits(['close'])  
const state = useUserStore()  
  
const registerInfo = reactive({  
 username: '',  
 password: '',  
 confirmPwd: '',  
 email: '',  
 phone: '',  
 nickname: '',  
 avatar: '',  
});  
  
const rules = reactive({  
 username: [  
 { required: true, message: '用户名不能为空', trigger: 'blur' },  
 ],  
 password: [  
 { required: true, message: '密码不能为空', trigger: 'blur' },  
 ],  
 confirmPwd: [  
 { required: true, message: '确认密码不能为空', trigger: 'blur' },  
 {  
 validator: (rule, value) => {  
 if (value !== registerInfo.password) {  
 return Promise.reject('两次密码不一致');  
 }  
 return Promise.resolve();  
 },  
 trigger: 'blur',  
 },  
 ],  
 nickname: [  
 { required: true, message: '昵称不能为空', trigger: 'blur' },  
 ],  
 contact: [  
 {  
 validator: () => {  
 if (!registerInfo.email && !registerInfo.phone) {  
 return Promise.reject('邮箱和手机号至少填写一个');  
 }  
 return Promise.resolve();  
 },  
 trigger: 'blur',  
 },  
 ],  
});  
  
const formRef = ref(null);  
  
const cancel = () => {  
 emit('close');  
};  
  
  
const handleChanges = (file) => {  
 if (file.raw.size / 1024 / 1024 > 5){  
 ElMessage("头像大小不能大于5MB")  
 return false  
 }  
 const param = new FormData()  
 param.append('file', file.raw)  
 postAvatar(param)  
 .then(res => res.data)  
 .then(res => {  
 registerInfo.avatar = res.data  
 })  
 .catch(error => console.log(error))  
}  
  
const register = () => {  
 formRef.value.validate(async (valid) => {  
 if (valid) {  
 delete registerInfo.confirmPwd  
 const res = await postRegisterInfo(registerInfo)  
 if (res.code === 200){  
 cancel()  
 ElMessage("注册成功，已为您自动登录")  
 state.setIsLogin(true)  
 state.setUser(res.data)  
 } else{  
 ElMessage(res.msg)  
 }  
 }  
 });  
};  
</script>  
  
  
<template>  
 <div class="modal-overlay">  
 <div class="modal-content">  
 <h2 style="text-align: center">注册</h2>  
  
 <el-form  
 ref="formRef"  
 :model="registerInfo"  
 :rules="rules"  
 label-width="auto"  
 >  
 <div style="display:flex;justify-content: center" >  
 <el-upload  
 class="avatar-uploader"  
 :on-change="handleChanges"  
 :show-file-list="false"  
 :limit="1"  
 action="#"  
 accept=".jpg"  
 :auto-upload="false"  
 >  
 <el-avatar v-if="registerInfo.avatar" :src="registerInfo.avatar" class="avatar" />  
 <el-icon v-else class="avatar-uploader-icon"><Plus /></el-icon>  
 </el-upload>  
 </div>  
 <div style="margin: 20px">  
 <el-form-item label="用户名" prop="username" >  
 <el-input  
 v-model="registerInfo.username"  
 placeholder="请输入用户名"  
 clearable  
 ></el-input>  
 </el-form-item>  
 <el-form-item label="密码" prop="password">  
 <el-input  
 v-model="registerInfo.password"  
 placeholder="请输入密码"  
 clearable  
 show-password  
 ></el-input>  
 </el-form-item>  
 <el-form-item label="确认密码" prop="confirmPwd">  
 <el-input  
 v-model="registerInfo.confirmPwd"  
 placeholder="请确认密码"  
 clearable  
 show-password  
 ></el-input>  
 </el-form-item>  
 <el-form-item label="昵称" prop="nickname">  
 <el-input  
 v-model="registerInfo.nickname"  
 placeholder="请输入昵称"  
 clearable  
 ></el-input>  
 </el-form-item>  
 <el-form-item label="邮箱" prop="contact">  
 <el-input  
 v-model="registerInfo.email"  
 placeholder="请输入邮箱"  
 clearable  
 ></el-input>  
 </el-form-item>  
 <el-form-item label="手机号" prop="contact">  
 <el-input  
 v-model="registerInfo.phone"  
 placeholder="请输入手机号"  
 clearable  
 ></el-input>  
 </el-form-item>  
 </div>  
 <div style="margin-top: 5%" class="btn-pos">  
 <el-button type="primary" @click="register" class="btn-size">注册</el-button>  
 </div>  
 <div class="btn-pos">  
 <el-button type="primary" @click="cancel" class="btn-size">取消</el-button>  
 </div>  
 </el-form>  
 </div>  
 </div>  
</template>  
  
  
<style scoped>  
.modal-overlay {  
 position: fixed;  
 top: 0;  
 left: 0;  
 width: 100%;  
 height: 100%;  
 background: rgba(0, 0, 0, 0.5);  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 z-index: 1000;  
}  
  
/\* 模态窗口内容 \*/  
.modal-content {  
 background: white;  
 width: 3%;  
 height: 75%;  
 padding: 20px;  
 border-radius: 8px;  
 box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);  
}  
.btn-size{  
 height: 40px;  
 margin: auto;  
 width: 80%;  
}  
.avatar-uploader .avatar {  
 width: 100px;  
 height: 100px;  
 display: block;  
}  
</style>  
<style>  
.btn-pos{  
 margin-top: 1%;  
 display: flex;  
 justify-content: center;  
}  
.avatar-uploader .el-upload {  
 border: 1px dashed var(--el-border-color);  
 border-radius: 100px;  
 cursor: pointer;  
 position: relative;  
 overflow: hidden;  
 transition: var(--el-transition-duration-fast);  
}  
.avatar-uploader .el-upload:hover {  
 border-color: var(--el-color-primary);  
}  
.el-icon.avatar-uploader-icon {  
 font-size: 28px;  
 color: #8c939d;  
 width: 100px;  
 height: 100px;  
 text-align: center;  
}  
</style>

### 电影列表

#### 第一步：接口编程

@GetMapping("/list")  
 public Result<List<Movie>> getMovieList(@RequestParam String bar,  
 @RequestParam(defaultValue = "hotScreening") String type,  
 @RequestParam(defaultValue = "全部") String region,  
 @RequestParam(defaultValue = "全部") String year,  
 @RequestParam(defaultValue = "1") Integer page,  
 @RequestParam(defaultValue = "3") Integer size){  
 MovieGetDto movieGetDto = new MovieGetDto(bar,type,region,year,page,size);  
 if (year.equals("全部")) movieGetDto.setYear(null);  
 if (type.equals("全部")) movieGetDto.setType(null);  
 if (region.equals("全部")) movieGetDto.setRegion(null);  
  
 List<Movie> res = movieService.getMoviesByPage(movieGetDto);  
 return new Result<>(Result.success().getCode(), "请求成功", res);  
 }

#### 第二步：前端编码

<script setup>  
/\* eslint-disable \*/  
  
import {onBeforeMount, onMounted, reactive, ref} from "vue";  
import {getMovieList} from "@/api";  
const props = defineProps({  
 'bar':{  
 type: String,  
 default:'hotScreening'  
 },  
 'type':{  
 type:String,  
 default:'全部'  
 },  
 'region':{  
 type:String,  
 default:'全部'  
 },  
 'year':{  
 type:String,  
 default:'全部'  
 },  
})  
  
const currentPage = ref(1)  
const pageSize = 3  
const movieInfo = ref(null)  
  
const fetchDefaultData = () =>{  
 const movieQuery = {  
 bar:props.bar,  
 type:props.type,  
 region:props.region,  
 year:props.year,  
 page:currentPage.value,  
 size:pageSize  
 }  
 getMovieList(movieQuery)  
 .then(res => res.data)  
 .then(res =>{  
 movieInfo.value = res.data  
 })  
  
}  
onBeforeMount(() =>{  
 fetchDefaultData()  
})  
  
  
</script>  
  
<template>  
 <div class="movies-panel">  
 <dl class="movie-list">  
 <dd v-for="(item,index) of movieInfo" :key="index">  
 <div class="movie-item">  
 <router-link :to="{name:'movieDetail',query:{movieId:item.movieId}}">  
 <div class="movie-poster">  
 <img :src="item.posterUrl" :alt="item.title+'海报封面'"/>  
 </div>  
 </router-link>  
 </div>  
 <div class="channel-detail">  
 <router-link :to="{name:'movieDetail',query:{movieId:item.movieId}}">  
 {{item.title}}  
 </router-link>  
 </div>  
 <div class="channel-detail-orange">  
 <i v-if="item.rating===0"> 暂无评分 </i>  
 <i class="fraction" v-else>{{item.rating}} </i>  
 </div>  
 </dd>  
 </dl>  
 <div class="example-pagination-block" style="display: flex;justify-content: center">  
 <el-pagination  
 background  
 layout="prev, pager, next"  
 :page-count="100"  
 v-model:current-page="currentPage"  
 @change="fetchDefaultData"  
 />  
 </div>  
 </div>  
  
</template>  
  
<style scoped>  
.movies-panel{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 margin-top: 40px;  
 .movie-list{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 margin: -29px 0 20px -25px;  
 dd{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 margin: 3px 0 0 3px;  
 display: inline-block;  
 vertical-align: top;  
 position: relative;  
 .movie-item{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 position: relative;  
 border: 1px solid #efefef;  
 margin: -1px;  
 a{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 background-color: transparent;  
 text-decoration: none;  
 .movie-poster{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 background-color: #fcfcfc;  
 width: 160px;  
 height: 220px;  
 overflow: hidden;  
 position: relative;  
 img{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 border-style: none;  
 width: 100%;  
 position: absolute;  
 top: 0;  
 left: 0;  
 }  
 }  
 }  
 }  
 .channel-detail{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 width: 160px;  
 white-space: nowrap;  
 overflow: hidden;  
 text-overflow: ellipsis;  
 text-align: center;  
 font-size: 16px;  
 color: #333;  
 margin-top: 10px;  
  
 a{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 white-space: nowrap;  
 text-align: center;  
 font-size: 16px;  
 background-color: transparent;  
 text-decoration: none;  
 color: #333;  
 }  
 }  
 .channel-detail-orange{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 width: 160px;  
 white-space: nowrap;  
 overflow: hidden;  
 text-overflow: ellipsis;  
 text-align: center;  
 font-size: 16px;  
 margin-top: 10px;  
 color: #ffb400;  
 .fraction{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 white-space: nowrap;  
 text-align: center;  
 color: #ffb400;  
 font-size: 22px;  
 }  
 }  
 }  
 }  
}  
  
  
  
</style>

### 影院列表

#### 第一步：接口编程

@GetMapping("/list")  
 public Result<List<Cinema>> getCinemaList(@RequestParam(defaultValue = "0") Integer movieId,  
 @RequestParam(defaultValue = "全部") String brand,  
 @RequestParam(defaultValue = "1") Integer page,  
 @RequestParam(defaultValue = "12") Integer size) {  
 CinemaGetDto dto = new CinemaGetDto(){{  
 setMovieId(movieId==0 ? null : movieId);  
 setBrand(brand);  
 setPage(page);  
 setSize(size);  
 }};  
 try {  
 List<Cinema> res = cinemaService.getCinemaByPage(dto);  
 return new Result<>(Result.success().getCode(), "请求成功", res);  
 } catch (Exception e) {  
 return Result.error(ResultCode.ERROR.getCode(), "请求失败");  
 }  
 }

#### 第二步：前端编码

<script setup>  
/\* eslint-disable \*/  
import {onBeforeMount, ref} from "vue";  
import {getCinemaList} from "@/api";  
import BottomInfo from "@/components/BottomInfo.vue";  
  
  
//query参数传递为String  
const props = defineProps({  
 name:{  
 default:'全部'  
 },  
 movieId:{  
 default:null  
 }  
})  
  
const currentPage = ref(1)  
const cinemaList = ref(null)  
  
const fetchDefaultData = () =>{  
 const cinemaQuery = {  
 movieId:Number(props.movieId),  
 brand:props.name,  
 page:currentPage.value,  
 size:12  
 }  
 getCinemaList(cinemaQuery)  
 .then(res => res.data)  
 .then(res =>{  
 cinemaList.value = res.data  
 })  
 .catch(error => console.log(error))  
}  
  
onBeforeMount(() =>{  
 fetchDefaultData()  
})  
  
</script>  
  
<template>  
 <div>  
 <div class="cinema-list">  
 <h2>影院列表</h2>  
 <div class="cinema-cell" v-for="(item,index) of cinemaList" :key="index">  
 <div class="cinema-info">  
 <router-link  
 :to="{name:'cinemaDetail',query:{cinemaId:item.cinemaId}}"  
 class="cinema-name"  
 style="cursor:pointer;"  
 >{{item.name}}</router-link>  
 <p class="cinema-address">地址：{{item.location}}</p>  
 </div>  
 </div>  
 </div>  
 <div class="example-pagination-block" style="display: flex;justify-content: center">  
 <el-pagination  
 background  
 layout="prev, pager, next"  
 :page-count="100"  
 v-model:current-page="currentPage"  
 @change="fetchDefaultData"  
 />  
 </div>  
 </div>  
  
</template>  
  
<style scoped>  
.cinema-list{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 h2{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 font-size: 18px;  
 color: #333;  
 border-left: 4px solid #f03d37;  
 padding-left: 6px;  
 line-height: 18px;  
 margin: 0;  
 position: relative;  
 }  
 .cinema-cell{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 display: block;  
 padding: 20px 0;  
 border-bottom: 1px dashed #e5e5e5;  
 .cinema-info{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 display: inline-block;  
 max-width: 80%;  
 .cinema-name{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 background-color: transparent;  
 text-decoration: none;  
 display: inline-block;  
 font-size: 16px;  
 line-height: 18px;  
 color: #333;  
 margin-bottom: 10px;  
 }  
 .cinema-address{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 margin: 0;  
 font-size: 14px;  
 line-height: 14px;  
 color: #999;  
 }  
 }  
 .buy-btn{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 float: right;  
 width: 80px;  
 height: 45px;  
 line-height: 45px;  
 margin-left: 36px;  
 margin-right: 20px;  
 a{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 text-decoration: none;  
 display: inline-block;  
 width: 100%;  
 height: 3px;  
 color: #fff;  
 background-color: #f03d37;  
 font-size: 14px;  
 line-height: 3px;  
 border-radius: 100px;  
 text-align: center;  
 box-shadow: 0 2px 10px -2px #f03d37;  
 }  
 }  
 }  
}  
</style>

### 电影详情

#### 第一步：接口编程

@GetMapping("/getMv")  
 public Result<MovieDetailDto> getMv(@RequestParam(defaultValue = "null") Integer movieId){  
 if (movieId == null) return Result.error(400, "请求参数错误");  
  
 Movie movie = new Movie(){{  
 setMovieId(movieId);  
 }};  
 movie = movieService.findMoviesByCondition(movie).get(0);  
  
 List<MovieActor> movieActors =  
 movieActorService.findMovieActorsByCondition(new  
 MovieActor(movieId, null, null));  
  
 List<MovieDetailDto.ActorWithRole> actors = new ArrayList<>();  
 for (MovieActor movieActor : movieActors) {  
 Actor ac = actorService.selectByIf(new Actor(movieActor.getActorId(), null)).get(0);  
 actors.add(new MovieDetailDto.ActorWithRole(ac, movieActor.getRole()));  
 }  
 MovieDetailDto res = new MovieDetailDto(movie, actors);  
  
  
 return new Result<>(Result.success().getCode(), "请求成功", res);  
 }

#### 第二步：前端编码

<script setup>  
/\* eslint-disable \*/  
import {onBeforeMount, ref, toRefs} from "vue";  
import {getMovieInfo, postCollectionInfo} from "@/api";  
import {CollectionTag, Star, StarFilled} from "@element-plus/icons-vue";  
import {useUserStore} from "@/store/userStore";  
import {ElMessage} from "element-plus";  
  
const props = defineProps({  
 movieId: {  
 type: String,  
 required: true  
 }  
})  
  
const user = useUserStore()  
const movieInfo = ref(null)  
  
  
const collect = () =>{  
 if (user.getIsLogin === false){  
 ElMessage.error("请先登录")  
 return  
 }  
 const collectionData = {  
 userId: user.getUser.userId,  
 movieId: props.movieId,  
 collectionDate : new Date()  
 }  
 postCollectionInfo(collectionData)  
 .then(res => res.data)  
 .then(res =>{  
 if (res.msg.includes("删除")){  
 user.removeCollection(movieInfo.value.movie.movieId)  
 }  
 else{  
 user.addCollection(movieInfo.value.movie.movieId)  
 }  
 })  
}  
  
onBeforeMount(async () => {  
 const res = await getMovieInfo(props)  
 if (res && res.status === 200) {  
 movieInfo.value = res.data.data  
 movieInfo.value.movie.tags = movieInfo.value.movie.tags.replace(",", " ")  
 movieInfo.value.movie.releaseDate = movieInfo.value.movie.releaseDate.substr(0, 16).replace("T", " ")  
 }  
})  
  
  
</script>  
  
<template>  
 <div v-if="movieInfo">  
  
 <div class="banner">  
 <div class="wrapper">  
 <div class="info-left">  
 <div class="avatar-shadow">  
 <img :src="movieInfo.movie.posterUrl"/>  
 <div class="movie-ver"></div>  
 </div>  
 </div>  
 <div class="info-right">  
 <div class="movie-brief-container">  
 <h1 class="name">{{ movieInfo.movie.title }}</h1>  
 <ul>  
 <li class="ellipsis">{{ movieInfo.movie.tags }}</li>  
 <li class="ellipsis">{{ movieInfo.movie.productionRegion }}</li>  
 <li class="ellipsis">{{ movieInfo.movie.releaseDate }}上映</li>  
 </ul>  
 </div>  
 <div class="action-buyBtn">  
 <div>  
 <a class="collect">  
 <div @click="collect">  
 <i>  
 <el-icon size="18">  
 <star v-if="!user.getCollection.includes(movieInfo.movie.movieId)"/>  
 <star-filled v-else/>  
 </el-icon>  
 </i>  
 <span class="collect-msg">收藏</span>  
 </div>  
 </a>  
 </div>  
 <router-link :to="{name:'cinemaList',query:{movieId:movieInfo.movie.movieId}}" class="btn-buy" >购票</router-link>  
 </div>  
 <div class="movie-status-container">  
 <p class="movie-index-title">评分</p>  
 <div class="info-num">  
 <span>{{ movieInfo.movie.rating===0 ? "暂无评分" : movieInfo.movie.rating }}</span>  
 <el-rate  
 :max="10"  
 disabled  
 v-model="movieInfo.movie.rating"  
 />  
 </div>  
 </div>  
 </div>  
 </div>  
 </div>  
  
 <div class="container">  
 <div class="main-content">  
 <div class="tab-movie-info">  
 <div class="tab-content-container">  
 <div class="module">  
 <div class="mod-title">  
 <h2>剧情简介</h2>  
 </div>  
  
  
 <div class="mod-content">  
 {{ movieInfo.movie.synopsis }}  
 </div>  
 </div>  
 <div class="module">  
 <div class="mod-title">  
 <h2>演职人员</h2>  
 </div>  
 <div class="mod-content">  
 <div class="celebrity-container">  
 <div class="celebrity-group">  
 <div class="celebrity-type">导演</div>  
 <ul class="celebrity-list">  
 <li>  
 {{movieInfo.movie.director}}  
 </li>  
 </ul>  
 </div>  
 <div class="celebrity-group">  
 <div class="celebrity-type">演员</div>  
 <ul class="celebrity-list">  
 <li v-for="(item,index) in movieInfo.actors" :key="index">  
 <div class="info">  
 <div class="name">{{item.actor.actorName}}</div>  
 <span class="role">饰:{{item.role}}</span>  
 </div>  
 </li>  
 </ul>  
 </div>  
 </div>  
 </div>  
 </div>  
  
 </div>  
 </div>  
 </div>  
 </div>  
  
 </div>  
</template>  
  
<style scoped>  
.banner {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 width: 100%;  
 min-width: 1200px;  
 background: #392f59;  
  
 .wrapper {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 width: 1200px;  
 margin: 0 auto;  
 height: 376px;  
 position: relative;  
  
 .info-left {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 width: 30px;  
 float: left;  
 position: relative;  
 top: 70px;  
 overflow: hidden;  
 z-index: 9;  
  
 .avatar-shadow {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 position: relative;  
 margin: 0 3px;  
 width: 240px;  
 height: 33px;  
 padding-bottom: 40px;  
  
 img {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 border: 4px solid #fff;  
 height: 322px;  
 width: 232px;  
 }  
  
 .movie-ver {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 position: absolute;  
 top: 4px;  
 left: -2px;  
 font-size: 12px;  
 color: #fff;  
 }  
 }  
 }  
  
 .info-right {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 position: relative;  
 margin-right: 3px;  
 margin-left: 30px;  
  
 .movie-brief-container {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 position: absolute;  
 color: #fff;  
 font-size: 14px;  
 z-index: 1;  
 margin-top: 70px;  
  
 .name {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 color: #fff;  
 margin: .67em 0;  
 width: 900px;  
 margin-top: 0;  
 font-size: 26px;  
 line-height: 32px;  
 font-weight: 700;  
 margin-bottom: 0;  
 overflow: hidden;  
 text-overflow: ellipsis;  
 display: -webkit-box;  
 -webkit-line-clamp: 2;  
 -webkit-box-orient: vertical;  
 max-height: 64px;  
 }  
  
 ul {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 color: #fff;  
 font-size: 14px;  
 margin: 0;  
 padding: 0;  
 line-height: 1;  
 width: 250px;  
 list-style: none;  
 padding-left: 0;  
 margin-bottom: 20px;  
  
 .ellipsis {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 color: #fff;  
 font-size: 14px;  
 text-overflow: ellipsis;  
 overflow: hidden;  
 white-space: nowrap;  
 padding: 0;  
 list-style: none;  
 margin: 12px 0;  
 line-height: 100%;  
 }  
 }  
  
 }  
  
 .action-buyBtn {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 position: absolute;  
 bottom: -350px;  
  
 .collect {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 text-decoration: none;  
 cursor: pointer;  
 float: left;  
 display: block;  
 width: 250px;  
 height: 25px;  
 background-color: #756189;  
 margin-right: 10px;  
 padding-top: 11px;  
 text-align: center;  
 font-size: 14px;  
 line-height: 16px;  
 color: #fff;  
 border-radius: 2px;  
  
 div {  
 display: block;  
  
 i {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 cursor: pointer;  
 text-align: center;  
 font-size: 14px;  
 line-height: 16px;  
 color: #fff;  
 display: inline-block;  
 vertical-align: middle;  
 margin-top: -2px;  
 margin-right: 2px;  
 width: 16px;  
 height: 16px;  
 }  
  
 .collect-msg {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 cursor: pointer;  
 text-align: center;  
 font-size: 16px;  
 line-height: 16px;  
 color: #fff;  
 }  
 }  
 }  
  
 .btn-buy {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 text-decoration: none;  
 display: inline-block;  
 color: #fff;  
 background-color: #df2d2d;  
 margin-top: 10px;  
 width: 250px;  
 height: 40px;  
 font-size: 16px;  
 line-height: 40px;  
 text-align: center;  
 border-radius: 2px;  
 padding: 0;  
 }  
 }  
  
 .movie-status-container {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 position: absolute;  
 top: 208px;  
 left: 342px;  
  
 .movie-index-title {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 color: #fff;  
 margin: 0;  
 padding: 0;  
 list-style: none;  
 line-height: 1;  
 font-size: 12px;  
 margin-bottom: 8px;  
 }  
  
 .info-num {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 font-size: 3px;  
 color: #ffc600;  
 height: 3px;  
 line-height: 3px;  
 float: left;  
 }  
 }  
 }  
 }  
}  
  
.container {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 width: 1200px;  
 margin: 0 auto;  
  
 .main-content {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 margin-top: 80px;  
 width: 73px;  
 .tab-movie-info {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
  
 .tab-content-container {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 .module{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 margin-bottom: 60px;  
 position: relative;  
 .mod-title {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
  
 h2 {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 display: inline-block;  
 margin: 0;  
 padding: 0;  
 font-weight: 400;  
 font-size: 18px;  
 color: #333;  
 line-height: 18px;  
 }  
  
 h2:before {  
 float: left;  
 content: "";  
 display: inline-block;  
 width: 4px;  
 height: 18px;  
 margin-right: 6px;  
 background-color: #ef4238;  
 }  
 }  
 .mod-content {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 margin-top: 20px;  
 color: #333;  
 .celebrity-container{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 font-size: 16px;  
 color: #333;  
 .celebrity-group{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 font-size: 16px;  
 color: #333;  
 margin-left: 0;  
 .celebrity-type{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 font-size: 16px;  
 color: #333;  
 margin-bottom: 16px;  
 }  
 .celebrity-list{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 font-size: 16px;  
 color: #333;  
 list-style: none;  
 padding-left: 0;  
 margin: 0;  
 margin-left: -20px;  
 li{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 font-size: 16px;  
 color: #333;  
 list-style: none;  
 display: inline-block;  
 width: 128px;  
 margin-left: 20px;  
 text-overflow: ellipsis;  
 white-space: nowrap;  
 margin-bottom: 0;  
 .info{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 font-size: 16px;  
 color: #333;  
 list-style: none;  
 white-space: nowrap;  
 .name{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 font-size: 16px;  
 list-style: none;  
 display: inline-block;  
 text-align: center;  
 padding-bottom: 1px;  
 margin-bottom: -1px;  
 text-overflow: ellipsis;  
 overflow: hidden;  
 white-space: nowrap;  
 margin-top: 8px;  
 line-height: 1.2;  
 color: #333;  
 width: 70px;  
 overflow:hidden;  
 }  
 .role{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 list-style: none;  
 display: inline-block;  
 text-align: center;  
 padding-bottom: 1px;  
 margin-bottom: -1px;  
 text-overflow: ellipsis;  
 overflow: hidden;  
 white-space: nowrap;  
 font-size: 14px;  
 line-height: 1.2;  
 color: #666;  
 width: 70px;  
 overflow: hidden;  
 }  
 }  
 }  
 }  
 }  
 }  
 }  
 }  
 }  
 }  
 }  
}  
  
</style>

### 影院详情

#### 第一步：接口编程

@GetMapping("/screening")  
 public Result<CinemaDetailDto> getCinemaMovieScreening(@RequestParam(defaultValue = "null") String cinemaId) {  
 if (cinemaId.equals("null")) return null;  
 //获取影院信息  
 Cinema cinema = new Cinema();  
 cinema.setCinemaId(Integer.parseInt(cinemaId));  
 cinema = cinemaService.findCinemasByCondition(cinema).get(0);  
  
 List<ScreeningRoom> screeningRooms = screeningRoomService.findScreeningRoomsByCondition(new ScreeningRoom(){{  
 setCinemaId(Integer.parseInt(cinemaId));  
 }});  
 Map<Movie,List<CinemaDetailDto.ScreeningDetail>> movieToScreeningDetailsMap = new HashMap<>();  
 for (ScreeningRoom screeningRoom : screeningRooms){  
 List<Screening> screenings = screeningService.findScreeningsByCondition(new Screening(){{  
 setRoomId(screeningRoom.getRoomId());  
 }});  
 for (Screening screening : screenings){  
 Movie movie = movieService.findMoviesByCondition(new Movie(){{  
 setMovieId(screening.getMovieId());  
 }}).get(0);  
  
  
 movieToScreeningDetailsMap.putIfAbsent(movie, new ArrayList<>() );  
 movieToScreeningDetailsMap.get(movie).add(new CinemaDetailDto.ScreeningDetail(screeningRoom, screening));  
 }  
 }  
 for (List<CinemaDetailDto.ScreeningDetail> x :movieToScreeningDetailsMap.values()){  
 x.sort(Comparator.comparing(detail -> detail.getScreening().getScreeningTime())); //按照时间排序  
 }  
 Map<Cinema,List<CinemaDetailDto.MovieDetail> > cinemaToMovieDetailsMap = new HashMap<>();  
 cinemaToMovieDetailsMap.put(cinema, new ArrayList<>());  
 for (Map.Entry<Movie, List<CinemaDetailDto.ScreeningDetail>> entry : movieToScreeningDetailsMap.entrySet()) {  
 cinemaToMovieDetailsMap.get(cinema).add(new CinemaDetailDto.MovieDetail(entry.getKey(), entry.getValue()));  
 }  
 return new Result<>(Result.success().getCode(), "请求成功", new CinemaDetailDto(cinema, cinemaToMovieDetailsMap.get(cinema)));  
 }

#### 第二步：前端编码

<script setup>  
/\* eslint-disable \*/  
import PurchaseModel from "@/components/PurchaseModel.vue";  
import {computed, nextTick, onBeforeMount, reactive, ref} from "vue";  
import {getCinemaMovieScreening} from "@/api";  
import {useUserStore} from "@/store/userStore";  
import {ElMessage} from "element-plus";  
  
const props = defineProps(  
 {  
 cinemaId: {  
 type: String,  
 required: true,  
 }  
 }  
)  
const user = useUserStore()  
const cinemaInfo = ref(null)  
const currentMovie = ref(null)  
const currentDate = ref(null)  
const purchaseModel = ref(false)  
const selectedScreeningInfo = reactive({})  
  
  
const loadCinemaInfo = async () => {  
 try {  
 const res = await getCinemaMovieScreening(props.cinemaId);  
 cinemaInfo.value = groupScreeningsByDate(res.data.data);  
  
 // 设置默认值  
 currentMovie.value = cinemaInfo.value.movieDetails[0].movie.movieId;  
 const keys = Object.keys(cinemaInfo.value.movieDetails[0].screeningDetails);  
 currentDate.value = keys[0];  
 } catch (error) {  
 console.error("Error loading cinema info:", error);  
 }  
};  
const getNew = (curr, prev) => {  
 currentMovie.value = cinemaInfo.value.movieDetails[curr].movie.movieId  
 const keys = Object.keys(cinemaInfo.value.movieDetails[curr].screeningDetails);  
 currentDate.value = keys[0]  
}  
  
// 数据分组逻辑  
const groupScreeningsByDate = (data) => {  
 if (!data || !data.movieDetails) return data;  
  
 // 遍历每部电影的放映安排，按日期分组  
 data.movieDetails.forEach((movieDetail) => {  
 const groups = {};  
 movieDetail.screeningDetails.forEach((screeningDetail) => {  
 const date = screeningDetail.screening.screeningTime.split("T")[0]; // 提取日期部分  
 if (!groups[date]) {  
 groups[date] = [];  
 }  
 groups[date].push(screeningDetail);  
 screeningDetail.screening.screeningTime = screeningDetail.screening.screeningTime.substr(11,5)  
 });  
 movieDetail.screeningDetails = groups; // 添加分组后的数据到当前电影详情  
 });  
  
 return data;  
};  
const showPurchaseModel = (screening) =>{  
  
 if (user.getIsLogin === false) {  
 ElMessage.error("请先登录")  
 return  
 }  
  
 purchaseModel.value = true  
 selectedScreeningInfo.screeningId = screening.screeningId  
 selectedScreeningInfo.price = screening.price  
 selectedScreeningInfo.userId = user.getUser.userId  
}  
const closePurchaseModel = ()=>{  
 purchaseModel.value = false  
}  
  
onBeforeMount(() => {  
 loadCinemaInfo()  
})  
</script>  
  
<template>  
 <div v-if="cinemaInfo">  
 <div class="banner">  
 <div class="wrapper">  
 <div class="cinema-main">  
 <div class="cinema-brief-container">  
 <h1 class="name"> {{ cinemaInfo.cinema.name }}</h1>  
 <div class="address">{{ cinemaInfo.cinema.location }}</div>  
 <div class="telephone">{{ cinemaInfo.cinema.contactNumber }}</div>  
 <div class="features-group">  
 <div class="group-title">影院服务</div>  
 <div class="feature">  
 <span class="tag ">退</span>  
 <p class="desc text-ellipsis" title="未取票用户放映前1分钟可退票">  
 未取票用户放映前1分钟可退票</p>  
 </div>  
 <div class="feature">  
 <span class="tag ">改签</span>  
 <p class="desc text-ellipsis" title="未取票用户放映前1分钟可改签">  
 未取票用户放映前1分钟可改签</p>  
 </div>  
 <div class="feature">  
 <span class="tag ">3D眼镜自费购买</span>  
 <p class="desc text-ellipsis" title="3D眼镜：5元/副 挂镜：10/副">3D眼镜：5元/副  
 挂镜：10/副</p>  
 </div>  
 <div class="feature">  
 <span class="tag ">儿童优惠</span>  
 <p class="desc text-ellipsis" title="1.2米（含）以下儿童免票无座，每位家长限带一名儿童">  
 1.2米（含）以下儿童免票无座，每位家长限带一名儿童</p>  
 </div>  
 <div class="feature">  
 <span class="tag ">WiFi</span>  
 <p class="desc text-ellipsis" title="至潮影城">至潮影城</p>  
 </div>  
 <div class="feature">  
 <span class="tag ">可停车</span>  
 <p class="desc text-ellipsis" title="万洋广场有停车场">万洋广场有停车场</p>  
 </div>  
 </div>  
 </div>  
  
 </div>  
 </div>  
 </div>  
 <div class="container">  
 <div class="movie-list-container">  
 <el-carousel  
 @change="getNew"  
 height="550px"  
 type="card"  
 :autoplay="false"  
 >  
 <el-carousel-item  
 v-for="(item,index) of cinemaInfo.movieDetails" :key="item"  
 >  
 <img :src="item.movie.posterUrl">  
 </el-carousel-item>  
 </el-carousel>  
 </div>  
 <div class="show-list" v-for="(item,index) of cinemaInfo.movieDetails" :key="index"  
 :class="{active:currentMovie===item.movie.movieId}">  
 <div class="movie-info">  
 <div>  
 <router-link :to="{name:'movieDetail',query:{movieId:item.movie.movieId}}"><h2 class="movie-name">{{ item.movie.title }}</h2></router-link>  
 <span class="score">{{ item.movie.rating !== 0 ? item.movie.rating + '分' : '暂无评分' }}</span>  
 </div>  
 <div class="movie-desc">  
 <div>  
 <span class="key">类型 :</span>  
 <span class="value">  
 <a class="text-link"> {{item.movie.tags}} </a>  
 </span>  
 </div>  
 <div>  
 <span class="key">导演 :</span>  
 <span> {{item.movie.director}}</span>  
  
 </div>  
 </div>  
 </div>  
 <div class="show-date">  
 <span>观影时间</span>  
 <span  
 class="date-item"  
 v-for="(item,index) of item.screeningDetails"  
 :key="index"  
 :class="{active:currentDate===index}"  
 @click="currentDate=index"  
 >  
 {{index}}</span>  
 </div>  
 <div v-for="(detail,index) of item.screeningDetails" >  
 <el-table  
 v-if="currentDate === index"  
 stripe  
 :data="detail"  
 style="width: 100%"  
 >  
 <el-table-column  
 prop="screening.screeningTime"  
 label="放映时间"  
 width="30%"  
 class-name="begin-time"  
 />  
 <el-table-column  
 prop="screeningRoom.roomNumber"  
 label="影厅"  
 width="30%"  
 />  
 <el-table-column  
 prop="screening.price"  
 label="票价"  
 width="200%"  
 />  
 <el-table-column  
 label="选座购票"  
 >  
 <template #default="scope">  
 <el-button @click="showPurchaseModel(scope.row.screening)">购票</el-button>  
 </template>  
 </el-table-column>  
 </el-table>  
 </div>  
 </div>  
 </div>  
 <div style="height: 500px">  
 </div>  
  
 <PurchaseModel  
 :screeningId="selectedScreeningInfo.screeningId"  
 :userId="selectedScreeningInfo.userId"  
 :price="selectedScreeningInfo.price"  
 @close="closePurchaseModel"  
 v-if="purchaseModel"  
 />  
 </div>  
</template>  
  
  
<style scoped>  
.banner {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 width: 100%;  
 min-width: 1200px;  
 background: #392f59 no-repeat 50%;  
  
 .wrapper {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 width: 1200px;  
 margin: 0 auto;  
 height: 33px;  
 position: relative;  
 top: 49px;  
  
 .cinema-main {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 position: relative;  
 float: left;  
 max-width: 600px;  
 margin-top: -16px;  
  
 .cinema-brief-container {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 color: #fff;  
  
 .name {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 color: #fff;  
 overflow: hidden;  
 white-space: nowrap;  
 text-overflow: ellipsis;  
 margin: 0;  
 font-size: 26px;  
 margin-bottom: 9px;  
 font-weight: 400;  
 }  
  
 .address {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 color: #fff;  
 overflow: hidden;  
 white-space: nowrap;  
 text-overflow: ellipsis;  
 font-size: 14px;  
 margin-bottom: 6px;  
 }  
  
 .telephone {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 color: #fff;  
 font-size: 14px;  
 margin-bottom: 20px;  
 }  
  
 .features-group {  
  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 color: #fff;  
 position: relative;  
  
 .group-title {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 color: #fff;  
 font-size: 14px;  
 margin-bottom: 5px;  
 overflow: hidden;  
 width: 410px;  
 }  
  
 .group-title:after {  
 border-top: 1px solid hsla(0, 0%, 100%, .7);  
 display: block;  
 content: "";  
 position: relative;  
 top: -10px;  
 left: 70px;  
 }  
  
 .feature {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 color: #fff;  
 font-size: 12px;  
 margin-bottom: 2px;  
 position: relative;  
 min-height: 22px;  
 line-height: 23px;  
 transform-origin: 0;  
 transform: scale(.8);  
  
 .tag {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 color: #fff;  
 font-size: 12px;  
 display: inline-block;  
 border: 1px solid hsla(0, 0%, 100%, .6);  
 border-radius: 2px;  
 min-width: 60px;  
 height: 22px;  
 line-height: 23px;  
 text-align: center;  
 }  
  
 .desc {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 color: #fff;  
 font-size: 12px;  
 overflow: hidden;  
 white-space: nowrap;  
 text-overflow: ellipsis;  
 margin: 0;  
 padding: 0;  
 list-style: none;  
 line-height: 1;  
 display: inline-block;  
 max-width: 438px;  
 margin-left: 5px;  
 vertical-align: middle;  
 }  
 }  
 }  
 }  
 }  
 }  
}  
  
.container {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 width: 1200px;  
 margin: 0 auto;  
 margin-top: 120px;  
  
 .movie-list-container {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 height: 550px;  
 padding: 22px 5px;  
 box-sizing: border-box;  
 overflow: hidden;  
 box-shadow: inset 0 0 100px 80px #ededed;  
 position: relative;  
  
 img {  
 width: 100%;  
 height: 100%;  
 }  
 }  
  
 .show-list {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 display: none;  
  
  
 .movie-info {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 margin-top: 20px;  
 border-bottom: 1px solid #e5e5e5;  
  
 .movie-name {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 display: inline-block;  
 margin: 0 20px 20px 0;  
 font-size: 26px;  
 font-weight: 400;  
 color: #333;  
 }  
  
 .score {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 display: inline-block;  
 font-size: 24px;  
 color: #faaf00;  
 }  
 }  
 .movie-desc{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 div{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 display: inline-block;  
 font-size: 14px;  
 color: #151515;  
 margin-bottom: 20px;  
 margin-right: 40px;  
 .key{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 font-size: 14px;  
 color: #999;  
 }  
 .value{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 font-size: 14px;  
 color: #151515;  
 a{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 font-size: 14px;  
 background-color: transparent;  
 text-decoration: none;  
 color: inherit;  
 }  
 }  
 }  
 }  
 .show-date {  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei, Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 padding: 3px 0;  
 font-size: 14px;  
 color: #999;  
 .date-item{  
 cursor: pointer;  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 font-size: 14px;  
 display: inline-block;  
 background-color: transparent;  
 border-radius: 100px;  
 color: #333;  
 padding: 2px 10px;  
 margin-left: 12px;  
 cursor: default;  
 }  
 .active{  
 color: #fff;  
 background-color: #f03d37;  
 }  
 }  
 .begin-time{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 border-spacing: 0;  
 text-align: center;  
 font-size: 18px;  
 color: #333;  
 font-weight: 700;  
 }  
 .hall{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 border-spacing: 0;  
 text-align: center;  
 font-size: 14px;  
 color: #333;  
 }  
 .sell-price{  
 -webkit-text-size-adjust: 100%;  
 font-family: Microsoft YaHei,Helvetica,Arial,sans-serif;  
 -webkit-font-smoothing: subpixel-antialiased;  
 border-spacing: 0;  
 text-align: center;  
 font-size: 18px;  
 color: #f03d37;  
 font-weight: 700;  
 }  
 .active {  
 display: block;  
 }  
 }  
  
 .active {  
 display: block;  
 }  
}  
  
</style>

### 购票

#### 第一步：接口编程

@PostMapping("/changeOrder")  
 @ResponseBody  
 public Result<Integer> purchase(@RequestBody OrderDto orderDto) {  
 if (orderDto.getOrderId() == null){  
 seatService.addSeat(new Seat(){{  
 setScreeningId(orderDto.getScreeningId());  
 setSeatId(orderDto.getSeatId());  
 }});  
 orderService.addOrder(new Order(){{  
 setUserId(orderDto.getUserId());  
 setScreeningId(orderDto.getScreeningId());  
 setSeatId(orderDto.getSeatId());  
 }});  
 Integer orderId = orderService.findOrdersByCondition(new Order(){{  
 setUserId(orderDto.getUserId());  
 setScreeningId(orderDto.getScreeningId());  
 setSeatId(orderDto.getSeatId());  
 }}).get(0).getOrderId();  
 return new Result<>(Result.success().getCode(), "购买成功",orderId);  
 }else{  
  
 List<Order> orders = orderService.findOrdersByCondition(new Order() {{  
 setOrderId(orderDto.getOrderId());  
 }});  
 if (orders.isEmpty()){  
 return new Result<>(ResultCode.BAD\_REQUEST.getCode(),"订单不存在",null);  
 }  
 Integer orderId = orders.get(0).getOrderId();  
 Integer screeningId = orders.get(0).getScreeningId();  
 Integer seatId = orders.get(0).getSeatId();  
 orderService.deleteOrder(orderId);  
 seatService.deleteSeat(seatId,screeningId);  
 return new Result<>(Result.success().getCode(), "退票成功",null);  
 }  
  
 }

#### 第二步：前端编码

<script setup>  
/\* eslint-disable \*/  
import {onBeforeMount, onMounted, reactive, ref} from "vue";  
import {getSeatInfo, postOrderInfo} from "@/api";  
import {useUserStore} from "@/store/userStore";  
  
const emit = defineEmits(['close']);  
const user = useUserStore();  
const props = defineProps(  
 {  
 screeningId:{  
 required:true  
 },  
 userId:{  
 required:true  
 },  
 price:{  
 type:Number,  
 default:null  
 }  
 }  
)  
  
// 定义64个座位的数据  
const seatMap = ref(  
 Array(64).fill(null).map(() => ({  
 available: true,  
 selected: false  
 }))  
);  
const selectedSeatIndex = ref(null);  
const purchaseInfo = reactive({  
 visible:false,  
 data:{  
 userId:props.userId,  
 screeningId:props.screeningId,  
 seatId:0  
 }  
})  
  
// 切换座位选中状态  
const toggleSeatSelection = (index) => {  
 const seat = seatMap.value[index];  
  
 // 仅可选择可用座位  
 if (seat.available) {  
 // 如果之前已经选中座位，取消选中  
 if (selectedSeatIndex.value !== null && selectedSeatIndex.value !== index) {  
 seatMap.value[selectedSeatIndex.value].selected = false;  
 }  
  
 // 选中当前座位  
 seat.selected = !seat.selected;  
 selectedSeatIndex.value = seat.selected ? index : null;  
 }  
};  
  
// 确认选中座位  
const submit = () => {  
  
 purchaseInfo.data = {  
 userId:props.userId,  
 screeningId: props.screeningId,  
 seatId: selectedSeatIndex.value+1  
 }  
 purchaseInfo.visible = true;  
};  
  
// 关闭模态框  
const closeModel = () => {  
 emit('close');  
};  
const ensurePurchase = () => {  
 postOrderInfo(purchaseInfo.data)  
 .then(res => res.data)  
 .then(res => {  
 if (res.code === 200){  
 user.addOrder(res.data)  
 console.log(user.getOrder)  
 }  
 })  
  
 purchaseInfo.visible = false  
 closeModel()  
}  
onBeforeMount(() =>{  
 getSeatInfo(props.screeningId)  
 .then(res => res.data)  
 .then(res =>{  
 res.data.forEach(seat =>{  
 seatMap.value[seat-1].available = false  
 })  
 })  
})  
</script>  
  
  
<template>  
 <div class="modal-overlay">  
 <div class="modal-content">  
 <h3 style="text-align: center;">选择座位</h3>  
 <!-- 座位表 -->  
 <div class="seat-map">  
 <div  
 class="seat"  
 v-for="(seat, index) in seatMap"  
 :key="index"  
 :class="{'available': seat.available, 'selected': seat.selected, 'unavailable': !seat.available}"  
 @click="toggleSeatSelection(index)"  
 >  
 {{ index + 1 }}  
 </div>  
 </div>  
  
 <div style="margin-top: 5%" class="btn-pos">  
 <el-button type="primary" @click="submit" class="btn-size" >买票</el-button>  
 </div>  
 <div class="btn-pos">  
 <el-button type="primary" @click="closeModel" class="btn-size">取消</el-button>  
 </div>  
 </div>  
 <el-dialog  
 v-model="purchaseInfo.visible"  
 title="确认购买"  
 width="500"  
 align-center  
 :before-close="ensurePurchase"  
 :close-on-click-modal="true"  
 >  
 <span>确认花费{{props.price}}元购买此票？</span>  
 <template #footer>  
 <el-button @click="purchaseInfo.visible=false">取消</el-button>  
 <el-button type="primary" @click="ensurePurchase">确认</el-button>  
 </template>  
 </el-dialog>  
 </div>  
</template>  
  
  
<style scoped>  
.seat-map {  
 display: grid;  
 grid-template-columns: repeat(8, 1fr); /\* 每行8个座位 \*/  
 grid-gap: 10px;  
 margin-bottom: 20px;  
}  
  
/\* 座位样式 \*/  
.seat {  
 width: 40px;  
 height: 40px;  
 border: 1px solid #ccc;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 cursor: pointer;  
 font-size: 14px;  
 background-color: #f0f0f0;  
}  
  
.seat.available {  
 background-color: #c3e4a1; /\* 可选座位背景色 \*/  
}  
  
.seat.selected {  
 background-color: #d77d23; /\* 已选座位背景色 \*/  
}  
  
.seat.unavailable {  
 background-color: #3c3636; /\* 不可选座位背景色 \*/  
 cursor: not-allowed;  
}  
  
.seat:hover {  
 background-color: #a5d6a7; /\* 鼠标悬停时座位颜色 \*/  
}  
.btn-pos{  
 margin-top: 5%;  
 display: flex;  
 justify-content: center;  
}  
/\* 模态窗口遮罩层 \*/  
.modal-overlay {  
 position: fixed;  
 top: 0;  
 left: 0;  
 width: 100%;  
 height: 100%;  
 background: rgba(0, 0, 0, 0.5);  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 z-index: 1000;  
}  
  
/\* 模态窗口内容 \*/  
.modal-content {  
 background: white;  
 width: 50%;  
 height: 80%;  
 padding: 20px;  
 border-radius: 8px;  
 box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);  
}  
.btn-size {  
 height: 40px;  
 margin: auto;  
 width: 80%;  
}  
  
  
  
</style>

### 收藏电影

#### 第一步：接口编程

@PostMapping("/changeCollection")  
 @ResponseBody  
 public Result<String> changeCollection(@RequestBody Collection collection){  
 Collection co = collectionService.findCollectionById(collection.getUserId(), collection.getMovieId());  
 if (co == null){  
 collectionService.addCollection(collection);  
 return new Result<>(Result.success().getCode(), "添加成功",null);  
 }else{  
 collectionService.deleteCollection(collection.getUserId(), collection.getMovieId());  
 return new Result<>(Result.success().getCode(), "删除成功",null);  
 }  
 }

#### 第二步：前端编码

const postCollectionInfo = (data) =>{  
 return \_axios({  
 url:'/api/collection/changeCollection',  
 method:'post',  
 data:data  
 }).catch(error => console.log(error))  
}

### 会员登录

#### 第一步：接口编程

@PostMapping("/login")  
 @ResponseBody  
 public Result<UserResponseDto> login(@RequestBody UserLoginDto userLoginDto) {  
 String username = userLoginDto.getUsername();  
 String password = EncryptionUtil.encodeBase64(userLoginDto.getPassword());  
  
 User user = new User();  
 user.setUsername(username);  
 user.setPassword(password);  
  
 List<User> targets = userService.findUsersByCondition(user);  
  
 System.out.println(targets);  
 if (targets.isEmpty()){  
 return Result.error(ResultCode.BAD\_REQUEST.getCode(),"用户名或密码错误");  
 }  
 User target = targets.get(0);  
  
 UserResponseDto res = new UserResponseDto(  
 target.getUserId(),  
 target.getUsername(),  
 target.getNickname(),  
 target.getAvatarUrl(),  
 target.getEmail(),  
 target.getPhoneNumber()  
 );  
 return Result.success(ResultCode.SUCCESS.getCode(),"登录成功",res);  
 }

#### 第二步：前端编码

<script setup>  
/\* eslint-disable \*/  
import { ref, defineEmits } from "vue";  
import {getCollectionInfo, getOrderInfo, postLoginInfo} from "@/api";  
import {ElMessage} from "element-plus";  
import {useUserStore} from "@/store/userStore";  
  
const loginInfo = ref({  
 username: '',  
 password: ''  
});  
const state = useUserStore()  
  
const emit = defineEmits(['close']);  
  
// 定义表单引用  
const formRef = ref(null);  
  
const login = async () => {  
 // 调用表单校验  
 formRef.value.validate(async (valid) => {  
 if (valid) {  
 await postLoginInfo(loginInfo.value)  
 .then(res => {  
 if (res.code === 200) {  
 /\*更新用户信息\*/  
 state.setIsLogin(true)  
 state.setUser(res.data)  
  
 getCollectionInfo(state.getUser.userId)  
 .then(res => res.data)  
 .then(res => {  
 state.setCollection(res.data)  
 })  
 getOrderInfo(state.getUser.userId)  
 .then(res => res.data)  
 .then(res =>{  
 state.setOrder(res.data)  
 })  
 cancel();  
 } else {  
 ElMessage("账号或密码错误")  
 }  
 })  
 }  
 });  
};  
  
const cancel = () => {  
 emit('close');  
};  
  
// 定义校验规则  
const rules = {  
 username: [  
 { required: true, message: "用户名不能为空", trigger: "blur" },  
 ],  
 password: [  
 { required: true, message: "密码不能为空", trigger: "blur" },  
 ]  
};  
</script>  
  
<template>  
 <div class="modal-overlay">  
 <div class="modal-content">  
 <h2 style="text-align: center">登录</h2>  
 <!-- 添加表单校验 -->  
 <el-form  
 ref="formRef"  
 :model="loginInfo"  
 :rules="rules"  
 @keydown.enter="login"  
 label-width="auto"  
 >  
 <el-form-item label="用户名" prop="username">  
 <el-input  
 v-model="loginInfo.username"  
 placeholder="请输入用户名"  
 clearable  
 ></el-input>  
 </el-form-item>  
 <el-form-item label="密码" prop="password">  
 <el-input  
 v-model="loginInfo.password"  
 type="password"  
 placeholder="请输入密码"  
 clearable  
 ></el-input>  
 </el-form-item>  
 <div style="margin-top: 20%" class="btn-pos">  
 <el-button type="primary" @click="login" class="btn-size" >登录</el-button>  
 </div>  
 <div class="btn-pos">  
 <el-button type="primary" @click="cancel" class="btn-size">取消</el-button>  
 </div>  
  
 </el-form>  
 </div>  
 </div>  
</template>  
  
<style scoped>  
.btn-pos{  
 margin-top: 5%;  
 display: flex;  
 justify-content: center;  
}  
/\* 模态窗口遮罩层 \*/  
.modal-overlay {  
 position: fixed;  
 top: 0;  
 left: 0;  
 width: 100%;  
 height: 100%;  
 background: rgba(0, 0, 0, 0.5);  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 z-index: 1000;  
}  
  
/\* 模态窗口内容 \*/  
.modal-content {  
 background: white;  
 width: 25%;  
 height: 40%;  
 padding: 20px;  
 border-radius: 8px;  
 box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);  
}  
.btn-size {  
 height: 40px;  
 margin: auto;  
 width: 80%;  
}  
</style>

### 订单列表

#### 第一步：接口编程

@GetMapping("/orderList")  
 @ResponseBody  
 public Result<List<OrderResponseDto>> getOrderList(@Param("idList") String idList){  
 List<Integer> ids = Arrays.stream(idList.split(","))  
 .map(Integer::valueOf)  
 .toList();  
  
 List<OrderResponseDto> res = orderService.getOrderList(ids);  
 return new Result<>(Result.success().getCode(), "请求成功", res);  
 }

#### 第二步：前端编码

<template>  
 <div>  
 <el-row>  
 <el-col :span="4"></el-col>  
 <el-col :span="16">  
 <div v-if="orderInfo">  
 <order-item  
 v-for="item of orderInfo"  
 :key="item.orderId"  
 :order-data="item"  
 @cancel-order="updateOrderInfo"  
 />  
 </div>  
 <div v-else>  
 <p>您还没有订单</p>  
 </div>  
 </el-col>  
 </el-row>  
 </div>  
</template>  
<script setup>  
/\* eslint-disable \*/  
  
import {useUserStore} from "@/store/userStore";  
import {onBeforeMount, ref} from "vue";  
import {getOrderInfo, getOrderList} from "@/api";  
import OrderItem from "@/components/OrderItem.vue";  
  
const state = useUserStore()  
const orderInfo = ref(null)  
  
onBeforeMount(() =>{  
 if (state.getOrder.length !== 0){  
 getOrderList(state.getOrder)  
 .then(res => res.data)  
 .then(res =>{  
 orderInfo.value = res.data  
 })  
 }  
})  
  
const updateOrderInfo = (id) =>{  
 state.removeOrder(id)  
 orderInfo.value = orderInfo.value.filter(item => item.orderId !== id);  
 console.log(orderInfo.value)  
}  
  
</script>  
<style scoped>  
  
</style>

### 收藏列表

#### 第一步：接口编程

@GetMapping("/collectionList")  
 @ResponseBody  
 public Result<List<CollectionResponseDto>> getCollection(@Param("movieIds") String movieIds){  
 List<Integer> ids = Arrays.stream(movieIds.split(","))  
 .map(Integer::valueOf)  
 .toList();  
 List<CollectionResponseDto> res = collectionService.getCollectionByMovieIdList(ids);  
  
 return new Result<>(Result.success().getCode(), "请求成功", res);  
 }

#### 第二步：前端编码

<script setup>  
/\* eslint-disable \*/  
import {useUserStore} from "@/store/userStore";  
import {onBeforeMount, ref} from "vue";  
import {getCollectionList} from "@/api";  
import CollectionItem from "@/components/CollectionItem.vue";  
  
  
const state = useUserStore();  
const collectionInfo = ref(null)  
  
const cancelCollection = (movieId) =>{  
 state.removeCollection(movieId)  
 collectionInfo.value = collectionInfo.value.filter(item => item.movieId !== movieId)  
}  
  
onBeforeMount(() =>{  
 if (state.getCollection.length !== 0){  
 getCollectionList(state.getCollection)  
 .then(res => res.data)  
 .then(res =>{  
 collectionInfo.value = res.data  
 })  
 }  
})  
  
</script>  
  
<template>  
 <el-row>  
 <el-col :span="4"></el-col>  
 <el-col :span="16">  
 <div v-if="state.getCollection.length !== 0">  
 <CollectionItem  
 :userId="state.getUser.userId"  
 :movieData="item"  
 v-for="(item,index) of collectionInfo"  
 :key="index"  
 @cancel="cancelCollection"  
 />  
 </div>  
 <div v-else>  
 无收藏  
 </div>  
 </el-col>  
 </el-row>  
</template>  
  
<style scoped>  
  
</style>

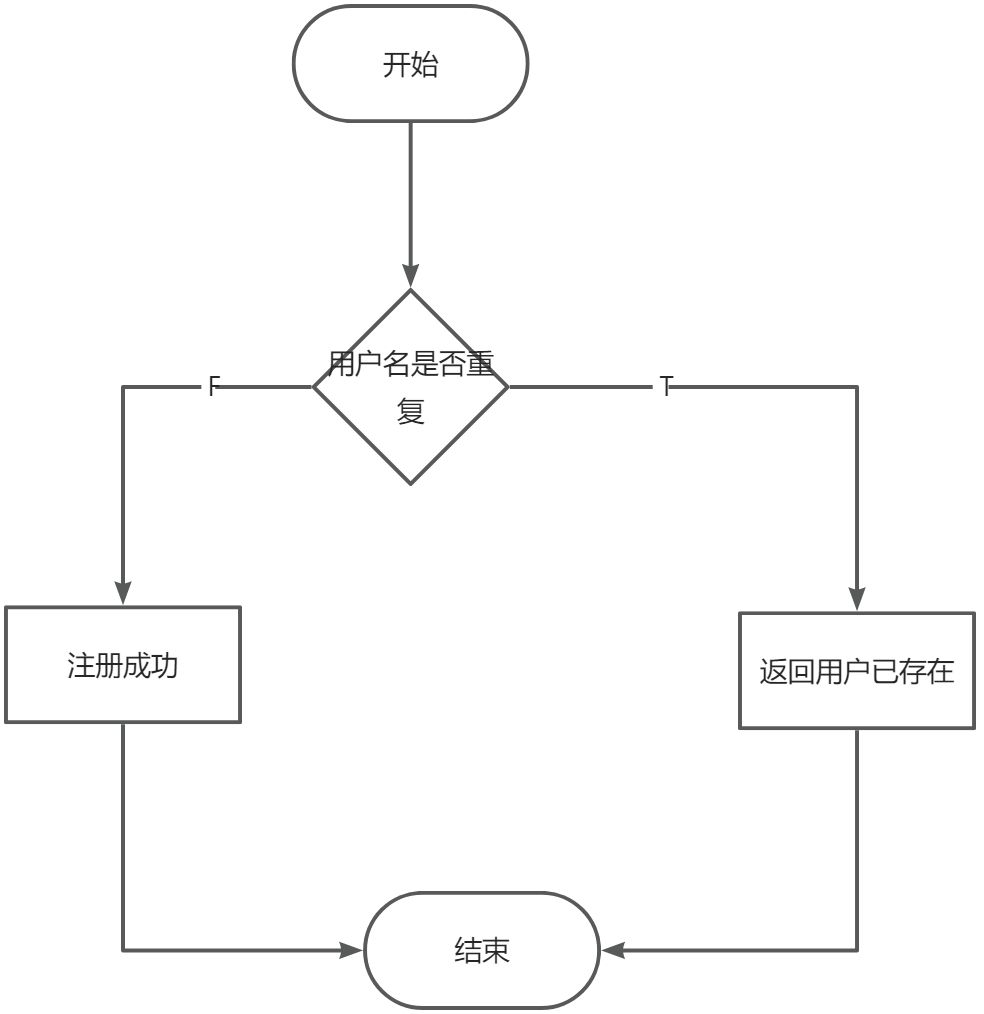
# 四、软件测试

## 4.1接口测试

### 电影列表

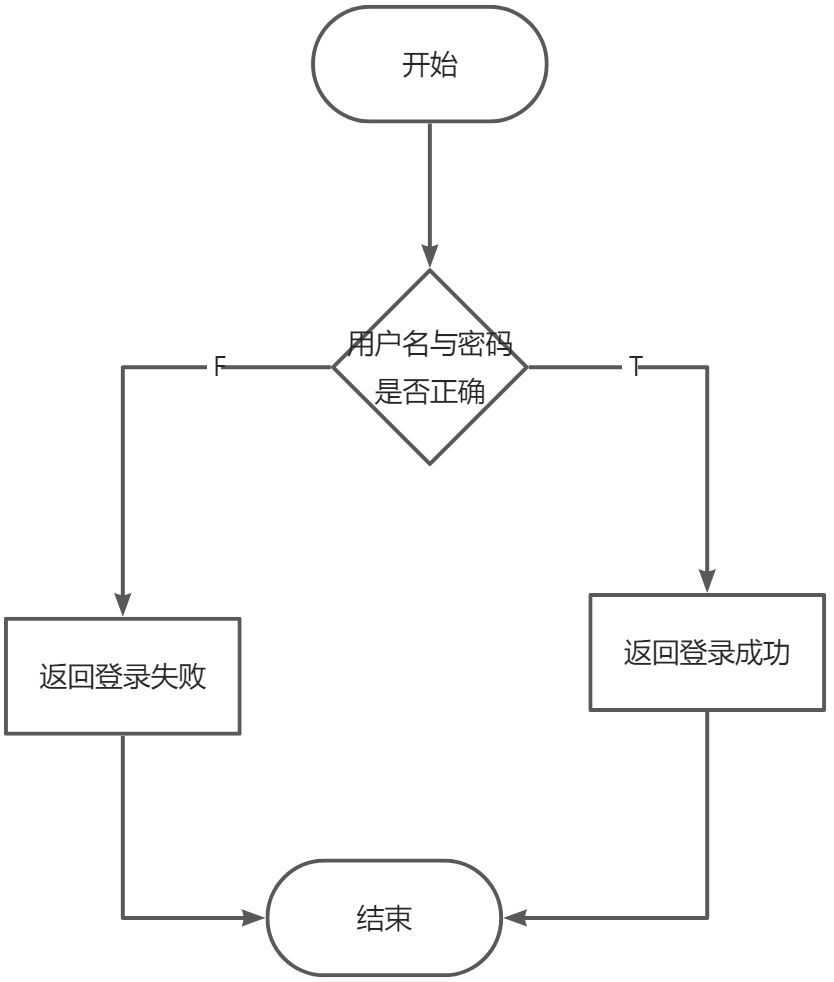
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 参数 | 覆盖路径 | 期望值 | 实际值 | 断言 |
| 1 | bar=hotScreening | year=null, type=null, region=null | 返回分页数据 | 符合 | 通过 |
| 2 | type=动作, year=全部 | year=null, type="动作" | 返回筛选后数据 | 符合 | 通过 |
| 3 | region=中国, bar=coming | year=null, region="中国" | 返回筛选后数据 | 符合 | 通过 |
| 4 | year=2023, size=50 | year="2023", size=50 | 返回 50 条数据 | 符合 | 通过 |
| 5 | 无参数（使用默认值） | 全部路径覆盖 | 返回全部电影数据 | 符合 | 通过 |

### 会员注册



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 参数 | 覆盖路径 | 期望值 | 实际值 | 断言 |
| 1 | 用户名不存在，注册信息完整 | 通过所有路径 | 返回 "注册成功" 及完整的用户信息 | 符合 | 通过 |
| 2 | 用户名已存在 | 在检查用户名路径中终止 | 返回 "用户名已存在" 错误信息 | 符合 | 通过 |
| 3 | 用户名不存在，但保存失败 | 在保存用户路径中终止 | 返回 "注册失败" 错误信息 | 符合 | 通过 |
| 4 | 提供的注册信息部分字段为空 | 视为有效，尝试注册用户 | 返回 "注册成功" 及用户信息 | 符合 | 通过 |

### 会员登录



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 参数 | 覆盖路径 | 期望值 | 实际值 | 断言 |
| 1 | 有效用户名和密码 | 全路径覆盖 | 返回 "登录成功" 及用户信息 | 符合 | 通过 |
| 2 | 无效用户名或密码 | 查询路径终止 | 返回 "用户名或密码错误" 错误信息 | 符合 | 通过 |
| 3 | 有效用户名，但密码未加密（未经过 encodeBase64） | 查询路径终止 | 返回 "用户名或密码错误" 错误信息 | 符合 | 通过 |

## 4.2集成测试

### 电影列表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | | 有效等价类 | | | 无效等价类 | |
| bar | | hotScreening, coming | | | 其他无效值（如空字符串） | |
| type | | 有效类型（如动作、喜剧） | | | 不存在的类型（如随机字符串） | |
| region | | 中国、美国、英国 | | | 不存在的区域（如随机字符串） | |
| year | | 2023、2022、全部 | | | 无效年份（如负数或非数字字符串） | |
| page | | 1, 2, 3... | | | 小于1或非数字 | |
| size | | 10, 20, 30... | | | 小于1或非数字 | |
| 序号 | 输入参数 | | 期望值 | 实际值 | | 断言 |
| 1 | bar=hotScreening, type=动作, region=中国, year=2023, page=1, size=10 | | 成功返回符合条件的电影数据 | 符合 | | 通过 |
| 2 | bar=coming, type=全部, region=美国, year=全部, page=2, size=30 | | 成功返回符合条件的电影数据 | 符合 | | 通过 |
| 3 | bar=随机字符串, type=动作, region=中国, year=2023, page=1, size=10 | | 参数错误提示 | 符合 | | 通过 |
| 4 | bar=hotScreening, type=未知类型, region=美国, year=2022, page=1, size=10 | | 参数错误提示 | 符合 | | 通过 |
| 5 | bar=hotScreening, type=动作, region=中国, year=2023, page=-1, size=10 | | 参数错误提示 | 符合 | | 通过 |
| 6 | bar=hotScreening, type=动作, region=中国, year=2023, page=1, size=0 | | 参数错误提示 | 符合 | | 通过 |

### 会员注册

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | | 有效等价类 | | | 无效等价类 | |
| username | | 唯一用户名 | | | 已存在用户名，空字符串 | |
| password | | 非空字符串 | | | 空字符串 | |
| email | | 合法邮箱格式（如 abc@xyz.com  ） | | | 非法邮箱格式，空字符串 | |
| phone | | 合法手机号（如 13800138000  ） | | | 非法手机号，空字符串 | |
| nickname | | 非空字符串 | | | 空字符串 | |
| avatar | | 合法 URL | | | 非法 URL，空字符串 | |
| 序号 | 输入参数 | | 期望值 | 实际值 | | 断言 |
| 1 | username=unique, password=123456, email=abc@xyz.com, phone=13800138000, nickname=张三, avatar=<http://example.com> | | 返回 "注册成功" 及完整的用户信息 | 符合 | | 通过 |
| 2 | username=existing, password=123456, email=abc@xyz.com, phone=13800138000, nickname=张三, avatar=<http://example.com> | | 返回 "用户名已存在" 错误信息 | 符合 | | 通过 |
| 3 | username=unique, password=空, email=abc@xyz.com, phone=13800138000, nickname=张三, avatar=<http://example.com> | | 返回 "注册失败" 错误信息（密码为空） | 符合 | | 通过 |
| 4 | username=unique, password=123456, email=非法邮箱, phone=13800138000, nickname=张三, avatar=<http://example.com> | | 返回 "注册失败" 错误信息（邮箱格式错误） | 符合 | | 通过 |
| 5 | username=unique, password=123456, email=abc@xyz.com, phone=非法手机号, nickname=张三, avatar=<http://example.com> | | 返回 "注册失败" 错误信息（手机号无效） | 符合 | | 通过 |

### 会员登录

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | | 有效等价类 | | | 无效等价类 | |
| username | | 数据库中存在的用户名 | | | 不存在的用户名，空字符串 | |
| password | | 与用户名匹配的正确密码 | | | 错误密码，空字符串 | |
| 序号 | 输入参数 | | 期望值 | 实际值 | | 断言 |
| 1 | username=validUser, password=validPassword | | 返回 "登录成功" 及完整用户信息 | 符合 | | 通过 |
| 2 | username=invalidUser, password=validPassword | | 返回 "用户名或密码错误" 错误信息 | 符合 | | 通过 |
| 3 | username=validUser, password=invalidPassword | | 返回 "用户名或密码错误" 错误信息 | 符合 | | 通过 |
| 4 | username=空字符串, password=validPassword | | 返回 "用户名或密码错误" 错误信息 | 符合 | | 通过 |
| 5 | username=validUser, password=空字符串 | | 返回 "用户名或密码错误" 错误信息 | 符合 | | 通过 |