# Node.js后端接口文件

const express = require('express');

const router = express.Router();

const conn = require('../db/db');

const svgCaptcha = require('svg-captcha');

const util = require('../util/util');

const multer = require('multer');

// 用户API

let user = {};

/\* GET home page. \*/

router.get('/', function(req, res, next) {

res.render('index', { title: 'Express' });

});

//获取手机验证码

router.get('/api/getPhoneCode', function(req, res) {

let phone = req.query.phone;

let phoneCode = util.randomCode(4);

if (!phoneCode) {

res.json({error\_code:1,message:'获取验证码失败'});

}else{

user[phone] = phoneCode;

res.json({success\_code:200,data:phoneCode});

}

});

//获取图形验证码

router.get('/api/captcha', function (req, res) {

let captcha = svgCaptcha.create({

size: 4, // size of random string

ignoreChars: '0o1i',// filter out some characters like 0o1i

noise: 3 ,// number of noise lines

color: true, //, characters will have distinct colors instead of grey, true if background option is set

background: '#fff', // background color of the svg image

});

//保存图形验证码文本

req.session.captcha = captcha.text.toLowerCase();

//返回验证码数据

res.type('svg');

res.status(200).send(captcha.data);

});

//手机登录

router.post('/api/phoneLogin',function(req, res){

let phone = req.body.phone;

let phoneCode = req.body.phoneCode;

//判断手机验证码是否正确

if (user[phone]===phoneCode) {

let sqlStr = 'SELECT \* from t\_user WHERE phone = ? LIMIT 1 ;';

conn.query(sqlStr,[phone],(error,result,field)=>{

if (error){

res.json({error\_code: 1,message: '请求数据失败'})

} else{

result = JSON.parse(JSON.stringify(result));

//用户存在

if (result[0]){

req.session.userId = result[0].user\_id;

res.cookie('user\_id',result[0].user\_id);

res.json({success\_code: 200});

} else{

//用户不存在

let sqlStr = 'INSERT INTO t\_user(user\_name,phone,avatar,password) VALUES(?,?,?,?)';

let avatarSrc = '/images/avatar/monkey.png';

conn.query(sqlStr,[phone,phone,avatarSrc,123456],(error,result,field)=>{

if (error){

console.log(error);

res.json({error\_code:1,message:'创建用户失败'});

} else{

res.cookie('user\_id',result.insertId);

res.json({success\_code:200})

}

})

}

}

})

} else{

res.json({error\_code:1,message:'验证码不正确'})

}

});

//密码登录

router.post('/api/pwdLogin',function(req,res){

let name = req.body.userName;

let pwd = req.body.password;

let captcha = req.body.captcha;

//判断验证码是否正确

if (captcha.toLowerCase()!==req.session.captcha){

res.json({error\_code:1,message:'验证码不正确'})

} else{

delete req.session.captcha;

let sqlStr = 'SELECT \* from t\_user WHERE user\_name =? LIMIT 1 ;'

conn.query(sqlStr,[name],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'查询用户失败'});

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

if (result[0].password===pwd){

//保存用户id

req.session.userId = result[0].user\_id;

res.cookie('user\_id',result[0].user\_id);

res.json({success\_code:200})

} else{

res.json({error\_code:1,message:'密码错误'});

}

} else{

res.json({error:1,message:'用户不存在'});

}

}

})

}

});

//获取用户信息

router.get('/api/getUserInfo',function(req,res){

let userId = req.query.userId;

if (userId){

let sqlStr = 'SELECT \* from t\_user WHERE user\_id = ? LIMIT 1;';

conn.query(sqlStr,[userId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'获取用户信息失败'});

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

res.json({success\_code:200,data:result[0]})

} else{

res.json({error\_code:1,message:'用户信息不存在'});

}

}

})

}

});

//更新用户头像

router.post('/api/updateUserAvatar',function(req,res){

let {

userId,

avatar,

} = req.body;

if (userId){

let sqlStr = 'SELECT \* from t\_user WHERE user\_id = ? LIMIT 1;';

conn.query(sqlStr,[userId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'用户不存在'});

} else{

//更新数据库

let sqlStr = 'UPDATE t\_user SET avatar = ? WHERE user\_id = ?;';

conn.query(sqlStr,[avatar,userId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'更新用户头像失败'});

} else{

res.json({success\_code:200});

}

});

}

})

}

});

//更新用户名

router.post('/api/updateUserName',function(req,res){

let {

userId,

userName,

} = req.body;

if (userId){

let sqlStr = 'SELECT \* from t\_user WHERE user\_id = ? LIMIT 1;';

conn.query(sqlStr,[userId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'用户不存在'});

} else{

sqlStr = 'SELECT \* FROM t\_user WHERE user\_name = ? AND user\_id <> ? LIMIT 1 ;';

conn.query(sqlStr,[userName,userId],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

res.json({error\_code:1,message:'用户名已存在！'});

} else{

//更新数据库

let sqlStr = 'UPDATE t\_user SET user\_name = ? WHERE user\_id = ?;';

conn.query(sqlStr,[userName,userId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'更新用户名失败'});

} else{

res.json({success\_code:200});

}

});

}

}

});

}

})

}

});

//更新用户性别

router.post('/api/updateUserSex',function(req,res){

let {

userId,

sex,

} = req.body;

if (userId){

let sqlStr = 'SELECT \* from t\_user WHERE user\_id = ? LIMIT 1;';

conn.query(sqlStr,[userId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'用户不存在'});

} else{

//更新数据库

let sqlStr = 'UPDATE t\_user SET sex = ? WHERE user\_id = ?;';

conn.query(sqlStr,[sex,userId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'更新用户性别失败'});

} else{

res.json({success\_code:200});

}

});

}

})

}

});

//更新用户生日

router.post('/api/updateUserBirthday',function(req,res){

let {

userId,

birthday,

} = req.body;

if (userId){

let sqlStr = 'SELECT \* from t\_user WHERE user\_id = ? LIMIT 1;';

conn.query(sqlStr,[userId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'用户不存在'});

} else{

//更新数据库

let sqlStr = 'UPDATE t\_user SET birthday = ? WHERE user\_id = ?;';

conn.query(sqlStr,[birthday,userId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'更新用户生日失败'});

} else{

res.json({success\_code:200});

}

});

}

})

}

});

//更新用户签名

router.post('/api/updateUserSign',function(req,res){

let {

userId,

sign,

} = req.body;

if (userId){

let sqlStr = 'SELECT \* from t\_user WHERE user\_id = ? LIMIT 1;';

conn.query(sqlStr,[userId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'用户不存在'});

} else{

//更新数据库

let sqlStr = 'UPDATE t\_user SET sign = ? WHERE user\_id = ?;';

conn.query(sqlStr,[sign,userId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'更新用户签名失败'});

} else{

res.json({success\_code:200});

}

});

}

})

}

});

//更新用户信息

router.post('/api/updateUserInfo',function(req,res){

let {

userId,

userName,

avatar,

password,

sex,

sign,

birthday

} = req.body;

if (userId){

let sqlStr = 'SELECT \* from t\_user WHERE user\_id = ? LIMIT 1;';

conn.query(sqlStr,[userId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'用户不存在'});

} else{

sqlStr = 'SELECT \* FROM t\_user WHERE user\_name = ? AND user\_id <> ? LIMIT 1 ;';

conn.query(sqlStr,[userName,userId],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

res.json({error\_code:1,message:'用户名已存在！'});

} else{

//更新数据库

let sqlStr = 'UPDATE t\_user SET user\_name = ?,avatar = ?,password = ?,sex = ? ,birthday = ?,sign = ?WHERE user\_id = ?;';

conn.query(sqlStr,[userName,avatar,password,sex,birthday,sign,userId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'更新用户信息失败'});

} else{

res.json({success\_code:200})

}

});

}

}

});

}

})

}

});

//加载电影列表

router.get('/api/getMovieList',function (req,res) {

let sqlStr = 'SELECT \* FROM t\_schedule INNER JOIN t\_movie ON t\_schedule.movie\_id = t\_movie.movie\_id;';

conn.query(sqlStr,(error,result,field)=>{

if (error){

console.log(error);

res.json({error\_code:1,message:'获取电影列表失败'})

} else{

result = JSON.parse(JSON.stringify(result));

if (result.length){

result = result.filter((value)=>{

return new Date(value.show\_date+','+value.show\_time)-new Date()>0;

});

for(let i = 0; i < result.length; i++) {

for(let j = i+1; j < result.length; j++) {

if(result[i]['movie\_id'] === result[j]['movie\_id']){

result.splice(j, 1); j = j-1;

}

}

}

res.json({success\_code:200,data:result})

} else{

res.json({error\_code:1,message:'电影列表为空'})

}

}

});

});

//加载电影详细信息

router.get('/api/getMovieDetail',function (req,res) {

let movieId = req.query.movieId;

let sqlStr = 'SELECT \* FROM t\_movie WHERE movie\_id = ? LIMIT 1;';

conn.query(sqlStr,[movieId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'获取电影信息失败'})

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

res.json({success\_code:200,data:result})

} else{

res.json({error\_code:1,message:'该电影不存在'})

}

}

});

});

//是否想看电影

router.post('/api/isWishMovie',function (req,res) {

let {

userId,

movieId

} = req.body;

let sqlStr = 'SELECT \* FROM t\_wishmovie WHERE user\_id = ? AND movie\_id = ? LIMIT 1;';

conn.query(sqlStr,[userId,movieId],(error,result,field)=>{

if (error){

console.log(error);

res.json({error\_code:1,message:'操作失败'});

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

res.json({success\_code:200});

} else {

res.json({error\_code:1,message:'不想看'});

}

}

})

});

//想看电影

router.post('/api/wishMovie',function (req,res) {

let {

userId,

movieId

} = req.body;

let sqlStr = 'INSERT INTO t\_wishmovie(user\_id,movie\_id) VALUES(?,?)';

conn.query(sqlStr,[userId,movieId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'操作失败'});

} else{

let sqlStr = 'SELECT wish\_num from t\_movie WHERE movie\_id = ? LIMIT 1;';

conn.query(sqlStr,[movieId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'操作失败'});

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

//更新数据库

let sqlStr = 'UPDATE t\_movie SET wish\_num = ? WHERE movie\_id = ?;';

conn.query(sqlStr,[result[0].wish\_num+1,movieId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'更新信息失败'});

} else{

res.json({success\_code:200})

}

})

}

}

})

}

})

});

//取消想看电影

router.post('/api/cancelWishMovie',function (req,res) {

let {

userId,

movieId

} = req.body;

let sqlStr = 'DELETE FROM t\_wishmovie WHERE user\_id = ? AND movie\_id =? ;';

conn.query(sqlStr,[userId,movieId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'操作失败'});

} else{

let sqlStr = 'SELECT wish\_num from t\_movie WHERE movie\_id = ? LIMIT 1;';

conn.query(sqlStr,[movieId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'操作失败'});

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

//更新数据库

let sqlStr = 'UPDATE t\_movie SET wish\_num = ? WHERE movie\_id = ?;';

conn.query(sqlStr,[result[0].wish\_num-1,movieId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'更新信息失败'});

} else{

res.json({success\_code:200})

}

})

}

}

})

}

})

});

//获取当前用户评论

router.get('/api/getUserComment',function (req,res) {

let {

userId,

movieId

} = req.query;

let sqlStr = 'SELECT \* FROM t\_comment WHERE user\_id = ? AND movie\_id= ? LIMIT 1;';

conn.query(sqlStr,[userId,movieId],(error,result,field)=>{

if (error){

console.log(error);

res.json({error\_code:1,message:'操作失败'});

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

res.json({success\_code:200,data:result[0]});

} else{

res.json({error\_code:1,message:'用户未评论'});

}

}

})

});

//获取所有用户通过审核的评论

router.get('/api/getAllUserPassComment',function (req,res) {

let {

movieId

} = req.query;

let sqlStr = 'SELECT \* FROM t\_user user INNER JOIN t\_comment comment ON user.user\_id = comment.user\_id WHERE comment.movie\_id = ? AND comment.is\_pass = ? ;';

conn.query(sqlStr,[movieId,1],(error,result,field)=>{

if (error){

console.log(error);

res.json({error\_code:1,message:'操作失败'});

} else{

result = JSON.parse(JSON.stringify(result));

if (result){

res.json({success\_code:200,data:result});

} else{

res.json({error\_code:1,message:'未评论'});

}

}

})

});

//更新用户评论

router.post('/api/updateUserComment',function (req,res) {

let{

userId,

movieId,

score,

commentContent,

commentDate

} = req.body;

let sqlStr = 'SELECT comment\_id from t\_comment WHERE user\_id = ? AND movie\_id = ? LIMIT 1';

conn.query(sqlStr,[userId,movieId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'操作失败'});

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

//评论存在

//更新

let sqlStr = 'UPDATE t\_comment SET user\_score = ?, comment\_content = ?, comment\_date = ?,support\_num = ?,is\_pass = ?,support\_user = ? WHERE comment\_id = ? ;';

conn.query(sqlStr,[score,commentContent,commentDate,0,0,undefined,result[0].comment\_id],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'更新评论失败'});

} else{

res.json({success\_code:200})

}

})

} else{

//评论不存在

let sqlStr = 'INSERT INTO t\_comment(user\_id,movie\_id,user\_score,comment\_content,comment\_date,support\_num,is\_pass) VALUES(?,?,?,?,?,?,?)';

conn.query(sqlStr,[userId,movieId,score,commentContent,commentDate,0,0],(error,result,field)=>{

if (error){

console.log(error);

res.json({error\_code:1,message:'操作失败'});

} else{

res.json({success\_code:200})

}

})

}

}

})

});

//获取当前评论

router.get('/api/getCommentByID',function (req,res) {

let {

commentId

} = req.query;

let sqlStr = 'SELECT \* FROM t\_comment WHERE comment\_id = ? LIMIT 1;';

conn.query(sqlStr,[commentId],(error,result,field)=>{

if (error){

console.log(error);

res.json({error\_code:1,message:'操作失败'});

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

res.json({success\_code:200,data:result[0]});

} else{

res.json({error\_code:1,message:'用户未评论'});

}

}

})

});

//更新当前评论的用户点赞

router.post('/api/updateUserSupport',function (req,res) {

let {

commentId,

supportNum,

supportUser

} = req.body;

let sqlStr = 'UPDATE t\_comment SET support\_num = ? , support\_user = ? WHERE comment\_id = ?;';

conn.query(sqlStr,[supportNum,supportUser,commentId],(error,result,field)=>{

if (error){

console.log(error);

res.json({error\_code:1,message:'操作失败'});

} else{

res.json({success\_code:200});

}

})

});

//加载影院列表

router.get('/api/getCinemaList',function (req,res) {

let sqlStr = 'SELECT \* FROM t\_schedule INNER JOIN t\_cinema ON t\_schedule.cinema\_id = t\_cinema.cinema\_id INNER JOIN t\_movie ON t\_schedule.movie\_id = t\_movie.movie\_id;';

conn.query(sqlStr,(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'获取影院列表失败'})

} else{

result = JSON.parse(JSON.stringify(result));

if (result.length){

result = result.filter((value)=>{

return new Date(value.show\_date+','+value.show\_time)-new Date()>0;

});

for(let i = 0; i < result.length; i++) {

for(let j = i+1; j < result.length; j++) {

if(result[i]['cinema\_id'] === result[j]['cinema\_id']){

result.splice(j, 1); j = j-1;

}

}

}

}

res.json({success\_code:200,data:result})

}

});

});

//加载当前影院详细信息

router.get('/api/getCurrentCinemaDetail',function (req,res) {

let cinemaId = req.query.cinemaId;

let sqlStr = 'SELECT \* FROM t\_cinema WHERE cinema\_id = ? LIMIT 1;';

conn.query(sqlStr,[cinemaId],(error,result,field)=>{

if (error){

console.log(error);

res.json({error\_code:1,message:'获取当前影院信息失败'})

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

res.json({success\_code:200,data:result[0]})

} else{

res.json({error\_code:1,message:'影院不存在'})

}

}

});

});

//加载当前影院排片

router.get('/api/getCurrentCinemaMovieSchedule',function (req,res) {

let cinemaId = req.query.cinemaId;

console.log(cinemaId);

let sqlStr = 'SELECT \* FROM t\_schedule WHERE cinema\_id = ?;';

conn.query(sqlStr,[cinemaId],(error,result,field)=>{

if (error){

console.log(error);

res.json({error\_code:1,message:'获取当前影院排片信息失败'})

} else{

result = JSON.parse(JSON.stringify(result));

if (result){

let tempMovieArr = [];

result.forEach((value)=>{

if (new Date()-new Date(value.show\_date+','+value.show\_time)<=0){

tempMovieArr.push(value.movie\_id);

}

});

tempMovieArr = Array.from(new Set(tempMovieArr));

let movieArray = [];

let movieScheduleArray = [];

tempMovieArr.forEach((value)=>{

sqlStr = 'SELECT \* FROM t\_movie WHERE movie\_id = ? LIMIT 1;';

conn.query(sqlStr,[value],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

movieArray.push(result[0]);

}

}

});

sqlStr = 'SELECT \* FROM t\_schedule schedule INNER JOIN t\_movie movie ON schedule.movie\_id = movie.movie\_id WHERE schedule.movie\_id = ? AND schedule.cinema\_id = ?;';

conn.query(sqlStr,[value,cinemaId],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

if (result){

movieScheduleArray.push(result);

}

}

})

});

setTimeout(()=>{

res.json({

success\_code:200,

data:{

hasMovieInfo:movieArray,

movieScheduleInfo:movieScheduleArray,

}

});

},500);

} else{

res.json({error\_code:1,message:'当前影院排片信息为空'})

}

}

});

});

//加载当前影院详细信息

router.get('/api/getScheduleById',function (req,res) {

let scheduleId = req.query.scheduleId;

let sqlStr = 'SELECT \* FROM t\_schedule WHERE schedule\_id = ? LIMIT 1;';

conn.query(sqlStr,[scheduleId],(error,result,field)=>{

if (error){

console.log(error);

res.json({error\_code:1,message:'获取当前排片信息失败'})

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

res.json({success\_code:200,data:result[0]})

} else{

res.json({error\_code:1,message:'排片信息不存在'})

}

}

});

});

//加载当前影院详细信息

router.post('/api/updateScheduleSeat',function (req,res) {

let {

scheduleId,

seatInfo

} = req.body;

let sqlStr = 'UPDATE t\_schedule SET seat\_info = ? WHERE schedule\_id = ? LIMIT 1;';

conn.query(sqlStr,[seatInfo,scheduleId],(error,result,field)=>{

if (error){

console.log(error);

res.json({error\_code:1,message:'更新排片座位信息失败'})

} else{

res.json({success\_code:200,data:result[0]})

}

});

});

//加载当前电影排片

router.get('/api/getCurrentMovieSchedule',function (req,res) {

let movieId = req.query.movieId;

let sqlStr = 'SELECT \* FROM t\_schedule WHERE movie\_id = ?;';

conn.query(sqlStr,[movieId],(error,result,field)=>{

if (error){

console.log(error);

res.json({error\_code:1,message:'获取当前影院排片信息失败'})

} else{

result = JSON.parse(JSON.stringify(result));

if (result){

let tempDateArr = [];

result.forEach((value)=>{

if (new Date()-new Date(value.show\_date+','+value.show\_time)<=0){

tempDateArr.push(value.show\_date);

}

});

tempDateArr = Array.from(new Set(tempDateArr));

tempDateArr.sort((a,b)=>{

return new Date(a)-new Date(b);

});

let cinemaArray = [];

let cinemaScheduleArray = [];

tempDateArr.forEach((value)=>{

sqlStr = 'SELECT \* FROM t\_schedule WHERE show\_date = ?';

conn.query(sqlStr,[value],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

if (result){

cinemaArray.push(result);

}

}

});

sqlStr = 'SELECT \* FROM t\_schedule schedule INNER JOIN t\_cinema cinema ON schedule.cinema\_id = cinema.cinema\_id WHERE schedule.movie\_id = ? AND schedule.show\_date = ?;';

conn.query(sqlStr,[movieId,value],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

if (result){

cinemaScheduleArray.push(result);

}

}

});

});

setTimeout(()=>{

res.json({

success\_code:200,

data:{

hasCinemaInfo:cinemaArray,

cinemaScheduleInfo:cinemaScheduleArray,

}

});

},500);

} else{

res.json({error\_code:1,message:'当前电影排片信息为空'})

}

}

});

});

//根据名字匹配电影

router.get('/api/matchMovieByName',function (req,res) {

let movieName = req.query.movieName;

let sqlStr = 'SELECT \* FROM t\_schedule INNER JOIN t\_movie ON t\_schedule.movie\_id = t\_movie.movie\_id WHERE name LIKE ?;';

conn.query(sqlStr,['%'+movieName+'%'],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

if (result.length){

result = result.filter((value)=>{

return new Date(value.show\_date+','+value.show\_time)-new Date()>0;

});

for(let i = 0; i < result.length; i++) {

for(let j = i+1; j < result.length; j++) {

if(result[i]['movie\_id'] === result[j]['movie\_id']){

result.splice(j, 1); j = j-1;

}

}

}

}

res.json({success\_code:200,data:result});

}

});

});

//根据名字匹配影院

router.get('/api/matchCinemaByName',function (req,res) {

let cinemaName = req.query.cinemaName;

let sqlStr = 'SELECT \* FROM t\_schedule INNER JOIN t\_cinema ON t\_schedule.cinema\_id = t\_cinema.cinema\_id INNER JOIN t\_movie ON t\_schedule.movie\_id = t\_movie.movie\_id WHERE cinema\_name LIKE ?;';

conn.query(sqlStr,['%'+cinemaName+'%'],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

if (result.length){

result = result.filter((value)=>{

return new Date(value.show\_date+','+value.show\_time)-new Date()>0;

});

for(let i = 0; i < result.length; i++) {

for(let j = i+1; j < result.length; j++) {

if(result[i]['cinema\_id'] === result[j]['cinema\_id']){

result.splice(j, 1); j = j-1;

}

}

}

}

res.json({success\_code:200,data:result});

}

});

});

//用户下单

router.post('/api/order',function (req,res) {

let {

userId,

scheduleId,

orderPhone,

orderDate,

ticketNum,

totalPrice,

orderSeatInfo,

payType

} = req.body;

let phoneCode = util.randomCode(6);

let sqlStr = 'INSERT INTO t\_order(user\_id,schedule\_id,order\_phone,order\_date,ticket\_num,ticket\_total\_price,order\_seat\_info,pay\_type,phone\_code) VALUES(?,?,?,?,?,?,?,?,?); ';

conn.query(sqlStr,[userId,scheduleId,orderPhone,orderDate,ticketNum,totalPrice,orderSeatInfo,payType,phoneCode],(error,result,field)=>{

if (error){

console.log(error);

} else{

res.json({success\_code:200,data:{phone\_code:phoneCode}});

}

});

});

//获取个人订单信息

router.get('/api/getOrderByUserId',function (req,res) {

let userId = req.query.userId;

let sqlStr = 'SELECT \* FROM t\_order INNER JOIN t\_schedule ON t\_order.schedule\_id = t\_schedule.schedule\_id INNER JOIN t\_movie ON t\_movie.movie\_id = t\_schedule.movie\_id INNER JOIN t\_cinema ON t\_cinema.cinema\_id = t\_schedule.cinema\_id WHERE user\_id = ?;';

conn.query(sqlStr,[userId],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

res.json({success\_code:200,data:result});

}

});

});

//获取个人想看电影

router.get('/api/getWishMovieByUserId',function(req,res){

let userId = req.query.userId;

let sqlStr = 'SELECT \* FROM t\_wishmovie INNER JOIN t\_movie ON t\_wishmovie.movie\_id = t\_movie.movie\_id WHERE user\_id = ?;';

conn.query(sqlStr,[userId],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

res.json({success\_code:200,data:result});

}

})

});

//获取个人看过的电影

router.get('/api/getIsWatchedMovieByUserId',function(req,res){

let userId = req.query.userId;

let sqlStr = 'SELECT \* FROM t\_comment INNER JOIN t\_movie ON t\_comment.movie\_id = t\_movie.movie\_id WHERE user\_id = ? AND is\_pass = 1;';

conn.query(sqlStr,[userId],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

res.json({success\_code:200,data:result});

}

})

});

//管理员API

//获取当前页用户

//密码登录

router.post('/api/admin/login',function(req,res){

let name = req.body.name;

let password = req.body.password;

let sqlStr = 'SELECT \* FROM t\_admin WHERE name = ? LIMIT 1 ;';

conn.query(sqlStr,[name,password],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'查询用户失败'});

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

if (result[0].password===password){

//保存用户id

req.session.adminId = result[0].admin\_id;

res.cookie('admin\_id',result[0].admin\_id);

res.json({success\_code:200})

} else{

res.json({error\_code:1,message:'密码错误'});

}

} else{

res.json({error:1,message:'用户不存在'});

}

}

})

});

//获取用户信息

router.get('/api/admin/getAdminInfo',function(req,res){

let adminId = req.query.adminId;

let sqlStr = 'SELECT \* FROM t\_admin WHERE admin\_id = ? LIMIT 1 ;';

conn.query(sqlStr,[adminId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'查询用户失败'});

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

res.json({success\_code:200,data:result[0]});

} else{

res.json({error:1,message:'用户不存在'});

}

}

})

});

router.get('/api/admin/getCurrentPageUser',function(req,res){

let {

currentPage, pageSize,input

} = req.query;

let start = Number((currentPage-1)\*pageSize);

pageSize = Number(pageSize);

let sqlStr = 'SELECT \* FROM t\_user WHERE user\_name LIKE ? ORDER BY user\_id;';

let total;

conn.query(sqlStr,["%"+input+"%"],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

total = result.length;

}

})

sqlStr = 'SELECT \* FROM t\_user WHERE user\_name LIKE ? ORDER BY user\_id LIMIT ?,?;';

conn.query(sqlStr,["%"+input+"%",start,pageSize],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

res.json({success\_code:200,data:result,total:total});

}

})

});

var datatime = './public/images/avatar/';

//将图片放到服务器

var storage = multer.diskStorage({

// 如果你提供的 destination 是一个函数，你需要负责创建文件夹

destination: datatime,

// //给上传文件重命名，获取添加后缀名

filename: function (req, file, cb) {

cb(null, new Date().getTime()+'.jpg');

}

});

var upload = multer({

storage: storage

});

// let upload = multer({dest:'./public/images/avatar'}).any();

router.post('/api/admin/upLoadImg',upload.any(),function (req,res) {

res.json({success\_code:200,data:req.files});

console.log(req.files);

});

//更新用户信息

router.post('/api/admin/updateUserInfo',function(req,res){

let {

userId,

userName,

avatar,

password,

sex,

phone,

sign,

birthday

} = req.body;

if (userId){

let sqlStr = 'SELECT \* from t\_user WHERE user\_id = ? LIMIT 1;';

conn.query(sqlStr,[userId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'用户不存在'});

} else{

let sqlStr = 'SELECT \* FROM t\_user WHERE user\_name = ? AND user\_id <> ? LIMIT 1;';

conn.query(sqlStr,[userName,userId],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

res.json({error\_code:1,message:'用户名已存在！'});

} else{

sqlStr = 'SELECT \* FROM t\_user WHERE phone = ? AND user\_id <> ? LIMIT 1;';

conn.query(sqlStr,[phone,userId],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

res.json({error\_code:1,message:'手机号码已注册！'});

} else{

//更新数据库

let sqlStr = 'UPDATE t\_user SET user\_name = ?,avatar = ?,password = ?,sex = ? ,phone = ?,birthday = ?,sign = ? WHERE user\_id = ?;';

conn.query(sqlStr,[userName,avatar,password,sex,phone,birthday,sign,userId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'更新用户信息失败'});

console.log(error);

} else{

res.json({success\_code:200})

}

});

}

}

})

}

}

});

}

})

}

});

//删除用户信息

router.post('/api/admin/deleteUserInfo',function(req,res){

let {

userId

} = req.body;

if (userId){

let sqlStr = 'SELECT t\_schedule.schedule\_id, t\_schedule.seat\_info,order\_seat\_info FROM t\_order INNER JOIN t\_schedule ON t\_order.schedule\_id = t\_schedule.schedule\_id WHERE user\_id = ?';

conn.query(sqlStr,[userId],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

console.log(result);

if (result){

result.forEach((value)=>{

let tempArr = [];

value.seat\_info = JSON.parse(value.seat\_info);

value.order\_seat\_info = JSON.parse(value.order\_seat\_info);

value.seat\_info.forEach((v)=>{

if(value.order\_seat\_info.indexOf(v)===-1){

tempArr.push(v);

}

});

tempArr = JSON.stringify(tempArr);

sqlStr = 'UPDATE t\_schedule SET seat\_info = ? WHERE schedule\_id = ?;';

conn.query(sqlStr,[tempArr,value.schedule\_id],(error,result,field)=>{

if (error){

console.log(error);

}

})

})

}

sqlStr = 'DELETE FROM t\_user WHERE user\_id =?';

conn.query(sqlStr,[userId],(error,result,field)=>{

if (error){

console.log(error);

} else {

res.json({success\_code:200});

}

})

}

})

}

});

//添加用户信息

router.post('/api/admin/addUserInfo',function(req,res){

let {

userName,

avatar,

phone,

password,

sex,

sign,

birthday

} = req.body;

if (!avatar){

avatar = '/images/avatar/monkey.png'

}

let sqlStr = 'SELECT \* FROM t\_user WHERE user\_name = ? LIMIT 1;';

conn.query(sqlStr,[userName],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

res.json({error\_code:1,message:'用户名已存在！'});

} else{

sqlStr = 'SELECT \* FROM t\_user WHERE phone = ? LIMIT 1';

conn.query(sqlStr,[phone],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

res.json({error\_code:1,message:'手机号码已注册！'});

} else{

sqlStr = 'INSERT INTO t\_user(user\_name,avatar,phone,password,sex,sign,birthday) VALUES(?,?,?,?,?,?,?);';

conn.query(sqlStr,[userName, avatar,phone, password, sex, sign, birthday],(error,result,field)=>{

if (error){

console.log(error);

} else{

res.json({success\_code:200});

}

})

}

}

});

}

}

});

});

//获取当前页电影

router.get('/api/admin/getCurrentPageMovie',function(req,res){

let {

currentPage, pageSize,input

} = req.query;

let start = Number((currentPage-1)\*pageSize);

pageSize = Number(pageSize);

let sqlStr = 'SELECT \* FROM t\_movie WHERE name LIKE ? ORDER BY movie\_id';

let total;

conn.query(sqlStr,["%"+input+"%"],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

total = result.length;

}

});

sqlStr = 'SELECT \* FROM t\_movie WHERE name LIKE ? ORDER BY movie\_id LIMIT ?,?;';

conn.query(sqlStr,["%"+input+"%",start,pageSize],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

res.json({success\_code:200,data:result,total:total});

}

})

});

//更新电影信息

router.post('/api/admin/updateMovieInfo',function(req,res){

let {

movieId,

movieName,

poster,

director,

actor,

long,

type,

language,

publicDate,

intro

} = req.body;

let sqlStr = 'SELECT \* FROM t\_movie WHERE name = ? AND movie\_id <> ? LIMIT 1;';

conn.query(sqlStr,[movieName,movieId],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

res.json({error\_code:1,message:'电影名已存在！'});

} else{

//更新数据库

let sqlStr = 'UPDATE t\_movie SET name = ?,poster = ?,director = ?,actor = ? ,movie\_long = ?,type = ?,language = ?,public\_date = ?,intro = ? WHERE movie\_id = ?;';

conn.query(sqlStr,[movieName,poster,director,actor,long,type,language,publicDate,intro,movieId],(error,result,field)=>{

if (error){

console.log(error);

} else{

res.json({success\_code:200})

}

})

}

}

});

});

//添加电影信息

router.post('/api/admin/addMovieInfo',function(req,res){

let {

movieName,

poster,

director,

actor,

long,

type,

language,

publicDate,

intro

} = req.body;

let sqlStr = 'SELECT \* FROM t\_movie WHERE name = ? LIMIT 1;';

conn.query(sqlStr,[movieName],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

res.json({error\_code:1,message:'电影名已存在！'});

} else{

let sqlStr = 'INSERT INTO t\_movie(name,poster,director,actor,movie\_long,type,language,public\_date,intro) VALUES(?,?,?,?,?,?,?,?,?);';

conn.query(sqlStr,[movieName,poster,director,actor,long,type,language,publicDate,intro],(error,result,field)=>{

if (error){

console.log(error);

} else{

res.json({success\_code:200});

}

})

}

}

});

});

datatime = './public/images/movie/';

//将图片放到服务器

storage = multer.diskStorage({

// 如果你提供的 destination 是一个函数，你需要负责创建文件夹

destination: datatime,

// //给上传文件重命名，获取添加后缀名

filename: function (req, file, cb) {

cb(null, new Date().getTime()+'.jpg');

}

});

upload = multer({

storage: storage

});

router.post('/api/admin/upLoadMovieImg',upload.any(),function (req,res) {

res.json({success\_code:200,data:req.files});

console.log(req.files);

});

//删除电影信息

router.post('/api/admin/deleteMovieInfo',function(req,res){

let {

movieId

} = req.body;

let sqlStr = 'DELETE FROM t\_movie WHERE movie\_id =?';

conn.query(sqlStr,[movieId],(error,result,field)=>{

if (error){

console.log(error);

} else {

res.json({success\_code:200});

}

})

});

//获取当前页影院

router.get('/api/admin/getCurrentPageCinema',function(req,res){

let {

currentPage, pageSize,input

} = req.query;

let start = Number((currentPage-1)\*pageSize);

pageSize = Number(pageSize);

let sqlStr = 'SELECT \* FROM t\_cinema WHERE cinema\_name LIKE ? ORDER BY cinema\_id ;';

let total;

conn.query(sqlStr,["%"+input+"%"],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

total = result.length;

}

});

sqlStr = 'SELECT \* FROM t\_cinema WHERE cinema\_name LIKE ? ORDER BY cinema\_id LIMIT ?,?;';

conn.query(sqlStr,["%"+input+"%",start,pageSize],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

res.json({success\_code:200,data:result,total:total});

}

})

});

//更新影院信息

router.post('/api/admin/updateCinemaInfo',function(req,res){

let {

cinemaId,

cinemaName,

cinemaPhone,

address

} = req.body;

if (cinemaId){

let sqlStr = 'SELECT \* from t\_cinema WHERE cinema\_id = ? LIMIT 1;';

conn.query(sqlStr,[cinemaId],(error,result,field)=>{

if (error){

console.log(error);

} else{

sqlStr = 'SELECT \* FROM t\_cinema WHERE cinema\_name = ? AND cinema\_id <> ? LIMIT 1 ;';

conn.query(sqlStr,[cinemaName,cinemaId],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

res.json({error\_code:1,message:'影院名已存在！'});

} else{

//更新数据库

let sqlStr = 'UPDATE t\_cinema SET cinema\_name = ?,cinema\_phone = ?,specified\_address = ? WHERE cinema\_id = ?;';

conn.query(sqlStr,[cinemaName,cinemaPhone,address,cinemaId],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'更新影院信息失败'});

console.log(error);

} else{

res.json({success\_code:200})

}

})

}

}

});

}

})

}

});

//删除影院信息

router.post('/api/admin/deleteCinemaInfo',function(req,res){

let {

cinemaId

} = req.body;

if (cinemaId){

let sqlStr = 'DELETE FROM t\_cinema WHERE cinema\_id =?';

conn.query(sqlStr,[cinemaId],(error,result,field)=>{

if (error){

console.log(error);

} else {

res.json({success\_code:200});

}

})

}

});

//添加影院信息

router.post('/api/admin/addCinemaInfo',function(req,res){

let {

cinemaName,

cinemaPhone,

address

} = req.body;

sqlStr = 'SELECT \* FROM t\_cinema WHERE cinema\_name = ? LIMIT 1 ;';

conn.query(sqlStr,[cinemaName],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

res.json({error\_code:1,message:'影院名已存在！'});

} else{

let sqlStr = 'INSERT INTO t\_cinema(cinema\_name,cinema\_phone,specified\_address) VALUES(?,?,?);';

conn.query(sqlStr,[cinemaName,cinemaPhone,address],(error,result,field)=>{

if (error){

console.log(error);

} else{

res.json({success\_code:200});

}

})

}

}

});

});

//获取当前页评论

router.get('/api/admin/getCurrentPageComment',function(req,res){

let {

currentPage, pageSize,input

} = req.query;

let start = Number((currentPage-1)\*pageSize);

pageSize = Number(pageSize);

let sqlStr = 'SELECT \* FROM t\_comment INNER JOIN t\_movie ON t\_comment.movie\_id = t\_movie.movie\_id INNER JOIN t\_user ON t\_user.user\_id=t\_comment.user\_id WHERE t\_movie.name LIKE ? ORDER BY comment\_id;';

let total;

conn.query(sqlStr,["%"+input+"%"],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

total = result.length;

}

});

sqlStr = 'SELECT \* FROM t\_comment INNER JOIN t\_movie ON t\_comment.movie\_id = t\_movie.movie\_id INNER JOIN t\_user ON t\_user.user\_id=t\_comment.user\_id WHERE t\_movie.name LIKE ? ORDER BY comment\_id LIMIT ?,?;';

conn.query(sqlStr,["%"+input+"%",start,pageSize],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

res.json({success\_code:200,data:result,total:total});

}

})

});

//通过当前评论

router.post('/api/admin/passCurrentComment',function (req,res) {

let commentId = req.body.commentId;

let movieId = req.body.movieId;

if (commentId){

let sqlStr = 'UPDATE t\_comment SET is\_pass = 1 WHERE comment\_id = ?;';

conn.query(sqlStr,[commentId],(error,result,field)=>{

if (error){

console.log(error);

} else{

sqlStr = 'SELECT user\_score FROM t\_comment WHERE movie\_id = ? AND is\_pass = ?;';

conn.query(sqlStr,[movieId,1],(error,result,field)=>{

if (error){

console.log(error);

} else {

result = JSON.parse(JSON.stringify(result));

let avgScore = 0;

if (result.length){

result.forEach((val)=>{

avgScore += (Number(val.user\_score));

});

avgScore = (avgScore/Number(result.length)).toFixed(1);

}

sqlStr = 'UPDATE t\_movie SET score = ? WHERE movie\_id = ?;';

conn.query(sqlStr,[avgScore,movieId],(error,result,field)=>{

if (error){

console.log(error);

} else{

res.json({success\_code:200});

}

})

}

});

}

})

}

});

//删除当前评论

router.post('/api/admin/deleteCurrentComment',function (req,res) {

let commentId = req.body.commentId;

let movieId = req.body.movieId;

if (commentId){

let sqlStr = 'DELETE FROM t\_comment WHERE comment\_id = ?;';

conn.query(sqlStr,[commentId],(error,result,field)=>{

if (error){

console.log(error);

} else{

sqlStr = 'SELECT user\_score FROM t\_comment WHERE movie\_id = ? AND is\_pass = ?;';

conn.query(sqlStr,[movieId,1],(error,result,field)=>{

if (error){

console.log(error);

} else {

result = JSON.parse(JSON.stringify(result));

let avgScore = 0;

if (result.length){

result.forEach((val)=>{

avgScore += (Number(val.user\_score));

});

avgScore = (avgScore/Number(result.length)).toFixed(1);

}

sqlStr = 'UPDATE t\_movie SET score = ? WHERE movie\_id = ?;';

conn.query(sqlStr,[avgScore,movieId],(error,result,field)=>{

if (error){

console.log(error);

} else{

res.json({success\_code:200});

}

})

}

});

}

})

}

});

//获取当前页订单

router.get('/api/admin/getCurrentPageOrder',function(req,res){

let {

currentPage, pageSize,input

} = req.query;

let start = Number((currentPage-1)\*pageSize);

pageSize = Number(pageSize);

let sqlStr = 'SELECT \* FROM t\_order INNER JOIN t\_schedule ON t\_order.schedule\_id = t\_schedule.schedule\_id INNER JOIN t\_movie ON t\_movie.movie\_id = t\_schedule.movie\_id INNER JOIN t\_cinema ON t\_cinema.cinema\_id = t\_schedule.cinema\_id INNER JOIN t\_user ON t\_order.user\_id = t\_user.user\_id WHERE t\_movie.name LIKE ? ORDER BY order\_id;';

let total;

conn.query(sqlStr,["%"+input+"%"],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

total = result.length;

}

});

sqlStr = 'SELECT \* FROM t\_order INNER JOIN t\_schedule ON t\_order.schedule\_id = t\_schedule.schedule\_id INNER JOIN t\_movie ON t\_movie.movie\_id = t\_schedule.movie\_id INNER JOIN t\_cinema ON t\_cinema.cinema\_id = t\_schedule.cinema\_id INNER JOIN t\_user ON t\_order.user\_id = t\_user.user\_id WHERE t\_movie.name LIKE ? ORDER BY order\_id LIMIT ?,?;';

conn.query(sqlStr,["%"+input+"%",start,pageSize],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

res.json({success\_code:200,data:result,total:total});

}

})

});

//获取所有订单

// 重新定义获取所有订单的接口路由，路径可按需调整

router.get('/api/admin/getAllOrders', function (req, res) {

// 构建查询所有订单的SQL语句，保持原有的多表关联查询逻辑，通过电影名模糊查询（如果有输入）

let sqlStr = 'SELECT \* FROM t\_order ' +

'INNER JOIN t\_schedule ON t\_order.schedule\_id = t\_schedule.schedule\_id ' +

'INNER JOIN t\_movie ON t\_movie.movie\_id = t\_schedule.movie\_id ' +

'INNER JOIN t\_cinema ON t\_cinema.cinema\_id = t\_schedule.cinema\_id ' +

'INNER JOIN t\_user ON t\_order.user\_id = t\_user.user\_id';

// 获取前端传来的模糊查询参数input，如果存在则添加模糊查询条件到SQL语句中

let { input } = req.query;

if (input) {

sqlStr += ` WHERE t\_movie.name LIKE '%${input}%'`;

}

sqlStr += ' ORDER BY order\_id';

// 执行数据库查询

conn.query(sqlStr, (error, result, fields) => {

if (error) {

console.log(error);

// 如果查询出现错误，返回包含错误码和错误信息的JSON响应给前端

res.status(500).json({ success\_code: 500, error\_message: '数据库查询失败，请稍后再试' });

} else {

// 对查询结果进行处理，原代码中有序列化和反序列化操作，此处保留，可根据实际情况决定是否需要

result = JSON.parse(JSON.stringify(result));

// 返回成功码为200以及包含所有订单数据的JSON响应给前端

res.json({ success\_code: 200, data: result });

}

});

});

//删除订单

router.post('/api/admin/deleteOrder',function(req,res){

let {

orderId,

scheduleId,

orderSeatInfo

} = req.body;

if (orderId){

let sqlStr = 'SELECT seat\_info FROM t\_schedule WHERE schedule\_id = ?';

conn.query(sqlStr,[scheduleId],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = result[0].seat\_info;

result = JSON.parse(result);

orderSeatInfo = JSON.parse(orderSeatInfo);

console.log(typeof result);

console.log(typeof orderSeatInfo);

let tempArr = [];

result.forEach((value)=>{

if (orderSeatInfo.indexOf(value)===-1){

tempArr.push(value);

}

});

console.log(tempArr);

result = JSON.stringify(tempArr);

sqlStr = 'UPDATE t\_schedule SET seat\_info = ? WHERE schedule\_id = ?;';

conn.query(sqlStr,[result,scheduleId],(error,result,field)=>{

if (error){

console.log(error);

} else{

sqlStr = 'DELETE FROM t\_order WHERE order\_id =?';

conn.query(sqlStr,[orderId],(error,result,field)=>{

if (error){

console.log(error);

} else {

res.json({success\_code:200});

}

})

}

})

}

})

}

});

//获取当前页影厅

router.get('/api/admin/getCurrentPageHall',function(req,res){

let {

currentPage, pageSize,input

} = req.query;

let start = Number((currentPage-1)\*pageSize);

pageSize = Number(pageSize);

let sqlStr = 'SELECT \* FROM t\_hall INNER JOIN t\_cinema ON t\_cinema.cinema\_id = t\_hall.cinema\_id WHERE t\_cinema.cinema\_name LIKE ? ORDER BY hall\_id;';

let total;

conn.query(sqlStr,["%"+input+"%"],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

total = result.length;

}

});

sqlStr = 'SELECT \* FROM t\_hall INNER JOIN t\_cinema ON t\_cinema.cinema\_id = t\_hall.cinema\_id WHERE t\_cinema.cinema\_name LIKE ? ORDER BY hall\_id LIMIT ?,?;';

conn.query(sqlStr,["%"+input+"%",start,pageSize],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

res.json({success\_code:200,data:result,total:total});

}

})

});

//删除影厅

router.post('/api/admin/deleteHall',function(req,res){

let {

cinemaId,

hallName

} = req.body;

if (cinemaId){

let sqlStr = 'SELECT schedule\_id FROM t\_schedule WHERE cinema\_id = ? AND hall\_name = ?;';

conn.query(sqlStr,[cinemaId,hallName],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

console.log(result);

if (result.length){

result.forEach((value)=>{

sqlStr = 'DELETE FROM t\_schedule WHERE schedule\_id = ?;';

conn.query(sqlStr,[value.schedule\_id],(error,result,field)=>{

if (error){

console.log(error);

}

})

})

}

sqlStr = 'DELETE FROM t\_hall WHERE cinema\_id = ? AND name = ?';

conn.query(sqlStr,[cinemaId,hallName],(error,result,field)=>{

if (error){

console.log(error);

} else {

res.json({success\_code:200});

}

})

}

});

}

});

//更新影厅信息

router.post('/api/admin/updateHallInfo',function(req,res){

let {

hallId,

cinemaId,

hallOldName,

hallNewName

} = req.body;

if (cinemaId){

let sqlStr = 'SELECT \* FROM t\_hall WHERE name = ? AND cinema\_id = ? AND hall\_id <> ? LIMIT 1;';

conn.query(sqlStr,[hallNewName,cinemaId,hallId],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

res.json({error\_code:1,message:'该影院的影厅已存在！'});

} else{

sqlStr = 'UPDATE t\_schedule SET hall\_name = ? WHERE cinema\_id = ? AND hall\_name = ?';

conn.query(sqlStr,[hallNewName,cinemaId,hallOldName],(error,result,field)=>{

if (error){

console.log(error);

} else{

//更新数据库

let sqlStr = 'UPDATE t\_hall SET name = ? WHERE cinema\_id = ? AND name = ?;';

conn.query(sqlStr,[hallNewName,cinemaId,hallOldName],(error,result,field)=>{

if (error){

res.json({error\_code:1,message:'更新影影厅信息失败'});

console.log(error);

} else{

res.json({success\_code:200})

}

})

}

});

}

}

});

}

});

//获取所有影院

router.get('/api/admin/getAllCinema',function(req,res){

let sqlStr = 'SELECT cinema\_id,cinema\_name FROM t\_cinema;';

conn.query(sqlStr,(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

res.json({success\_code:200,data:result});

}

})

});

//添加影厅信息

router.post('/api/admin/addHallInfo',function(req,res){

let {

cinemaId,

hallName

} = req.body;

let sqlStr = 'SELECT \* FROM t\_hall WHERE name = ? AND cinema\_id = ? LIMIT 1;';

conn.query(sqlStr,[hallName,cinemaId],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

if (result[0]){

res.json({error\_code:1,message:'该影院的影厅已存在！'});

} else{

let sqlStr = 'INSERT INTO t\_hall(cinema\_id,name) VALUES(?,?);';

conn.query(sqlStr,[cinemaId,hallName],(error,result,field)=>{

if (error){

console.log(error);

} else{

res.json({success\_code:200});

}

});

}

}

});

});

//获取当前页电影

router.get('/api/admin/getCurrentPageMovieSchedule',function(req,res){

let {

currentPage, pageSize,input

} = req.query;

let start = Number((currentPage-1)\*pageSize);

pageSize = Number(pageSize);

let sqlStr = 'SELECT \* FROM t\_schedule INNER JOIN t\_movie ON t\_schedule.movie\_id = t\_movie.movie\_id INNER JOIN t\_cinema ON t\_cinema.cinema\_id = t\_schedule.cinema\_id WHERE t\_movie.name LIKE ? ORDER BY schedule\_id ';

let total;

conn.query(sqlStr,["%"+input+"%"],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

total = result.length;

}

});

sqlStr = 'SELECT \* FROM t\_schedule INNER JOIN t\_movie ON t\_schedule.movie\_id = t\_movie.movie\_id INNER JOIN t\_cinema ON t\_cinema.cinema\_id = t\_schedule.cinema\_id WHERE t\_movie.name LIKE ? ORDER BY schedule\_id LIMIT ?,?;';

conn.query(sqlStr,["%"+input+"%",start,pageSize],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

res.json({success\_code:200,data:result,total:total});

}

})

});

//获取所有电影

router.get('/api/admin/getAllMovie',function(req,res){

let sqlStr = 'SELECT \* FROM t\_movie;';

conn.query(sqlStr,(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

res.json({success\_code:200,data:result});

}

})

});

//获取影院的影厅

router.get('/api/admin/getHallByCinemaId',function(req,res){

let cinemaId = req.query.cinemaId;

console.log(cinemaId);

let sqlStr = 'SELECT hall\_id,name FROM t\_hall WHERE cinema\_id = ?;';

conn.query(sqlStr,[cinemaId],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

console.log(result);

res.json({success\_code:200,data:result});

}

})

});

//获取排片的某天的时间段安排

router.get('/api/admin/getHasScheduleDateTime',function(req,res){

let {

cinemaId,hallName,showDate

} = req.query;

let sqlStr = 'SELECT show\_time FROM t\_schedule WHERE cinema\_id = ? AND hall\_name = ? AND show\_date = ?;';

conn.query(sqlStr,[cinemaId,hallName,showDate],(error,result,field)=>{

if (error){

console.log(error);

} else{

result = JSON.parse(JSON.stringify(result));

res.json({success\_code:200,data:result});

}

})

});

//添加影院信息

router.post('/api/admin/addScheduleInfo',function(req,res){

let {

movieId,cinemaId,hallName,showDate,showTime,price

} = req.body;

let sqlStr = 'INSERT INTO t\_schedule(movie\_id,cinema\_id,hall\_name,show\_date,show\_time,price) VALUES(?,?,?,?,?,?);';

conn.query(sqlStr,[movieId,cinemaId,hallName,showDate,showTime,price],(error,result,field)=>{

if (error){

console.log(error);

} else{

res.json({success\_code:200});

}

})

});

//删除影厅

router.post('/api/admin/deleteMovieSchedule',function(req,res){

let {

scheduleId

} = req.body;

if (scheduleId){

let sqlStr = 'DELETE FROM t\_schedule WHERE schedule\_id = ?';

conn.query(sqlStr,[scheduleId],(error,result,field)=>{

if (error){

console.log(error);

} else {

res.json({success\_code:200});

}

})

}

});

module.exports = router;