var express = require('express');

var router = express.Router();

//连接数据库

const connection = require('./js/connect');

//引入jwt

const jwt = require('jsonwebtoken');

//引入express-jwt

const expressJwt = require('express-jwt')

//设置响应头

router.all('\*',(req,res,next)=>{

// 设置响应头解决跨域

res.setHeader("Access-Control-Allow-Origin", "\*");

res.setHeader("Access-Control-Allow-Headers", "authorization"); // 允许通过头部信息authorization 携带token

// 放行

next();

})

/\*---------验证token合法性------\*/

//准备一个签名（密钥）

const secretKey = 'qikesystem';

//验证token的合法性，设置需要保护的API

router.use(expressJwt({

secret:secretKey

}).unless({

path:['/login/checklogin']//排除掉这个地址，其他的url都需要验证（其他的请求都要带上token，才能获得数据，否则不能得到数据）

}));

//路由器拦截,校验token失败时的处理

router.use((err,req,res,next)=>{

// 如果前端发送的请求没有token或者是错误的token就抛出错误

if(err.name==="UnauthorizedError"){

res.status(401).send({code:2,msg:"登录过期，请重新登录！"})

}

})

/\*---------验证token合法性------\*/

//增加礼品

router.post('/giftadd',(req,res)=>{

//接收传过来的数据

let { giftclassify,giftname,giftcode,costscore,price,stocknum,giftunit,issale,giftdesc } = req.body;

// 写sql

const sqlStr = `insert into giftdata(giftclassify,giftname,giftcode,costscore,price,stocknum,giftunit,issale,giftdesc) values('${giftclassify}','${giftname}','${giftcode}','${costscore}','${price}','${stocknum}','${giftunit}','${issale}','${giftdesc}') `;

console.log(sqlStr);

// 执行sql

connection.query(sqlStr,(err,data)=>{

if(err) throw err;

if(data.affectedRows>0){

res.send({code:0,msg:"添加数据成功"});

}else {

res.send({code:0,msg:"添加数据失败"});

}

})

})

//分页刷新列表

router.get('/giftmanage',(req,res)=>{

//接收前端传过来的参数

let {pageSize,currentPage,classify,keyword}=req.query;

//准备sql

let sqlStr = "select \* from giftdata where 1=1";

//如果有分类

if(classify&&classify!=="全部"){

sqlStr+=` and giftclassify='${classify}'`;

}

//如果有关键字

if(keyword){

sqlStr+=` and(giftname like '%${keyword}%' or giftcode like '%${keyword}%')`;

}

//拼接降序排列

sqlStr+=' order by create\_time desc';

//执行sql

connection.query(sqlStr,(err,data)=>{

if(err) throw err;

// 获得总条数

let total=data.length;

// 跳过多少条

let n=(currentPage-1)\*pageSize;

// 每页显示多少条

let m=pageSize;

// 写sql

sqlStr+=` limit ${n},${m}`;

console.log(sqlStr);

// 执行sql

connection.query(sqlStr,(err,data)=>{

if(err) throw err;

// 发送数据回前端

res.send({total,data});

})

})

})

//删除数据

router.get('/giftdel',(req,res)=>{

// 获取前端传过来的参数

let {id}=req.query;

// 写sql

const sqlStr=`delete from giftdata where id='${id}'`;

console.log(sqlStr);

// 执行sql

connection.query(sqlStr,(err,data)=>{

if(err) throw err;

if(data.affectedRows>0){

res.send({code:0,msg:"删除数据成功"})

}else {

res.send({code:1,msg:"删除数据失败"})

}

})

})

//修改回填数据

router.get('/editgift',(req,res)=>{

// 接收参数

let {id}=req.query;

// 写sql

const sqlStr=`select \* from giftdata where id=${id}`;

// 执行sql

connection.query(sqlStr,(err,data)=>{

if(err) throw err;

res.send(data);

})

})

//保存数据

router.post('/savegift',(req,res)=>{

// 接收参数

let { id,giftclassify,giftname,giftcode,costscore,price,stocknum,giftunit,issale,giftdesc } = req.body;

// 写sql

const sqlStr = `update giftdata set giftclassify="${giftclassify}",giftname="${giftname}",giftcode="${giftcode}",costscore="${costscore}",price="${price}",stocknum="${stocknum}",giftunit="${giftunit}",issale="${issale}",giftdesc="${giftdesc}" where id=${id}`;

// 执行sql

connection.query(sqlStr,(err,data)=>{

if(err) throw err;

if(data.affectedRows>0){

res.send({code:0,msg:"修改数据成功"});

}else {

res.send({code:0,msg:"修改数据失败"});

}

})

})

//批量删除

router.get('/batchdel',(req,res)=>{

// 接收参数

let {ids}=req.query;

// 写sql

const sqlStr =`delete from giftdata where id in (${ids})`;

console.log(sqlStr);

// 执行sql

connection.query(sqlStr,(err,data)=>{

if(err) throw err;

if(data.affectedRows>0){

res.send({code:0,msg:"批量删除成功"})

}else {

res.send({code:1,msg:"批量删除失败"})

}

})

})

//验证礼品码是否正确

router.get('/checkcode',(req,res)=>{

// 获取参数

let { code } = req.query;

// 写sql

const sqlStr = `select \* from giftdata where giftcode='${code}'`;

// 执行sql

connection.query(sqlStr,(err,data)=>{

if(err) throw err;

if(data.length){

if(data[0].stocknum>0){

res.send({code:0,data:data[0]});

}else {

res.send({code:1,data:"礼品已兑换完"})

}

}else {

res.send({code:1,data:"礼品码不存在"})

}

})

})

//礼品兑换

router.post('/giftsaleout',(req,res)=>{

// 接收参数

let { giftname,giftcode,costscore,outnum,total,id,stocknum,studentnum,studentname,studentscore}=req.body;

console.log(studentscore);

// 存礼品兑换数据

// 生成事件戳

let ordernum=new Date().getTime();

// 写sql

let sqlStr=`insert into orderdata (ordernum,giftname,giftcode,costscore,outnum,total,studentname) values (?,?,?,?,?,?,?)`;

const sqlParams=[ordernum,giftname,giftcode,costscore,outnum,total,studentname];

connection.query(sqlStr,sqlParams,(err,data)=>{

if(err) throw err;

if(data.affectedRows>0){

// 减少礼品库中的库存

sqlStr=`update giftdata set stocknum='${stocknum}' where id='${id}'`;

connection.query(sqlStr,(err,data)=>{

if(err) throw err;

if(data.affectedRows>0){

//修改学生积分

sqlStr=`update studentdata set score='${studentscore}' where id='${studentnum}'`;

console.log(sqlStr);

connection.query(sqlStr,(err,data)=>{

if(err) throw err;

if(data.affectedRows>0){

res.send({code:0,score:studentscore})

}else {

res.send({code:1,score:studentscore})

}

})

}else {

res.send({code:1})

}

})

}

})

})

//兑换过程中兑换列表响应

router.get('/getexchange',(req,res)=>{

const sqlStr=`select`

})

//验证学生是否存在

router.post('/checkstudent',(req,res)=>{

let {stunum}=req.body;

const sqlStr=`select \* from studentdata where id=${stunum}`;

connection.query(sqlStr,(err,data)=>{

if(err) throw err;

if(data.length>0){

res.send({code:0,data:data[0]});

}else {

res.send({code:1,data:"该学生不存在！"})

}

})

})

//商品兑换列表

router.get('/giftexchangelist',(req,res)=>{

//接收前端传过来的参数

let {pageSize,currentPage,pickerTime,keyword}=req.query;

//准备sql

let sqlStr = "select \* from orderdata where 1=1";

//如果有分类

if(pickerTime.length>0){

sqlStr+=` and (create\_time between '${pickerTime[0]}' and '${pickerTime[1]}')`;

}

//如果有关键字

if(keyword){

sqlStr+=` and(giftname like '%${keyword}%' or studentname like '%${keyword}%')`;

}

//拼接降序排列

sqlStr+=' order by create\_time desc';

console.log(sqlStr);

//执行sql

connection.query(sqlStr,(err,data)=>{

if(err) throw err;

// 获得总条数

let total=data.length;

// 跳过多少条

let n=(currentPage-1)\*pageSize;

// 每页显示多少条

let m=pageSize;

// 写sql

sqlStr+=` limit ${n},${m}`;

console.log(sqlStr);

// 执行sql

connection.query(sqlStr,(err,data)=>{

if(err) throw err;

// 发送数据回前端

res.send({total,data});

})

})

})

module.exports = router;