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------\*/

//验证登录名

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

// 接收参数

let { account,password }=req.body;

// 写sql

const sqlStr = `select \* from staffdata where account='${account}' and password='${password}'`;

// 执行sql

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

if(err) throw err;

// 接收数据库传回的数据返回到前端

if(data.length){

//生成token，把token和数据一起传给前端

const token = jwt.sign(Object.assign({},data[0]),secretKey,{expiresIn:60\*60\*2})

res.send({code:0,msg:"登录成功，欢迎进入启课教育管理系统！",token})

}else {

res.send({code:1,msg:"登录失败，请检查账号或密码！"})

}

})

})

//获取当前账号

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

res.send(req.user.name);

})

module.exports = router;