

勘误

1.sequelize全局禁用时间戳功能

```
const sequelize = new Sequelize('kkb', 'kaikeba_admin', 'admin', {  
  ...  
  define: {  
    timestamps: false,  
    freezeTableName: true  
  }  
});
```

2.新增公开课错误消息未初始化

RESTful API

1. GET/POST/PUT/DELETE
2. 编写接口: res.json(obj)
3. 接口测试方法:

跨域：浏览器同源策略（协议、主机名、端口）引起的前端接口调用问题

常见解决方案：

1. JSONP(Json With Padding)，前端+后端，绕过跨域

原理：前端动态添加script到当前页面，其src指向接口URL，服务器返回一个函数执行语句，该函数名有callback参数决定

实现：

前端：指定请求方式为jsonp, jquery中 {dataType:true}

后端：res.jsonp(data)

接口必须是get方法，传递参数只能通过url的方式

2. PROXY 代理，后端方案

原理：通过同源服务器发送请求至接口服务器，再由同源服务器转发结果给前端，从而规避跨域

webpack-dev-server

proxy

3. CORS (Cross) ，后端方案

原理：CORS是W3C规范，真正意义上解决跨域问题。他需要服务器对请求检查，对响应头做特殊处理，从而解决跨域

实现：

- 1.对于简单请求，添加：res.set('Access-Control-Allow-Origin', <http://localhost:8080>);

简单请求：方法是GET、POST、HEADER，同时对于post请求，不能有自定义请求头，同时编码方式必须是application/x-www-urlencoded, multipart/form-data,text/plain之一

2.对于预检测preflight请求，添加：

1) 预检测路由 router.options()

2) 响应头添加res.set('Access-Control-Allow-Headers', 'Content-Type');

3) 如果请求方法是PUT或DELETE添加res.set('Access-Control-Allow-Methods', 'GET,POST,PUT');

3.对于证书类型Credentials：请求中携带cookie的请求，需要额外处理

1) 在options和请求路由中均添加：res.set('Access-Control-Allow-Credentials', 'true');

4.cors模块：npm i cors -S

1) 基本使用

```
const cors = require('cors')
app.use(cors()); //门户大开，相当于Access-Control-Allow-Origin: *

app.use(cors({ //只对localhost:8080开，且可携带cookie
  origin: 'http://localhost:8080',
  credentials: true
}));

//白名单
var whitelist = ['http://example1.com', 'http://example2.com']
app.use(cors({
  origin: function (origin, callback) {
    if (whitelist.indexOf(origin) !== -1) {
      callback(null, true)
    } else {
      callback(new Error('Not allowed by CORS'))
    }
  }
}));

//异步配置
var whitelist = ['http://example1.com', 'http://example2.com']
var corsOptionsDelegate = function (req, callback) {
  var corsOptions;
  if (whitelist.indexOf(req.header('Origin')) !== -1) {
    corsOptions = { origin: true } // 生成配置项
  } else {
    corsOptions = { origin: false }
  }
  callback(null, corsOptions) // 将选项作为参数2
}
```

公开课报名功能开发

作业：

- 1.复习以往知识
- 2.简单预习[angular](#)
- 3.准备[秒滴](#)账号和短信验证模板，适当了解其短信验证码api