

# 昌南大屏代码走查

---

REVIEWER: 彭玲 TIME: 2021/6/25

---

## 昌南大屏代码走查

项目提交规范

不必要不提交

README.md

开发环境配置

lanch.json

package.json

代码规范

代码逻辑

语法问题

函数误用

map() 错误地做遍历用

eval() -> JSON.parse()

函数参数被重新赋值

Unused code

数据库访问问题

条件表达式

变量重复声明

注释

注释说明

已废弃代码

---

## 项目提交规范

---

### 不必要不提交

- log 文件夹及日志文件
- package-lock.json 文件有何特殊用途？如无必要，不做提交。

### README.md

README 是项目的“简历”，目的是提供关于项目的**有用**信息，比如，项目开发环境（Node.js 版本、数据库及其它启动参数、包的特殊版本等）、生产环境、版本功能等。

昌南大屏 webapi 的 README.md 提供下面的信息，不但没有有用信息，相反，还会产生误导的不好影响。

### ### 启动参数

---

```
-a, --anxinyun    安心云api
-e, --es          es服务URL
-p, --port        启动端口
```

---

### ### 日志路径

~/log/runtime.txt

## 开发环境配置

### launch.json

`.vscode/launch.json` 开发调试参数慎用商用环境，原则上不用，尤其是涉及增删改操作。

```
"args": [
  "-a https://openapi.anxinyun.cn/api/v1",
  "-p 4000",
  "-e node35:9200,node36:9200,node37:9200",
  "-g postgres://changnan:123456@10.8.30.156/changnan",
  "-i province_1119A",
  "-w water_state_72",
  "-l http://10.244.123.20:9087",
  "-t https://toilet.anxinyun.cn/api", // 智慧公厕商用地址
  "-s https://smartsite.anxinyun.cn/api", // 智慧工地商用地址
  "-f http://10.8.30.35:8887",
  "--fw http://10.8.30.185:5000",
  "-m http://10.244.186.202:8881"
]
```

### package.json

`scripts` 脚本配置中，项目启动参数：

- 如无特殊情况，参数（如服务端口 `-p`）可与 `.vscode/launch.json` 保持一致，以减少其他环境的访问配置修改（如web侧、postman工具等）；
- 如无必要，参数可在 `config.js` 中设置**默认值**，以减少开发环境配置项的维护工作。

## 代码规范

- 文件名  
统一文件命名：socialConstruction.js、anxinyun\_proxy.js、openapi-validator.js
- 文件头  
典型地，无文件头、文件头创建人信息不实、无修改人信息等：

```
/**
 * Created by PengLing on 2017/7/10.
 */
```

- 代码干净整洁

app/lib/auth-code.js 存在其他项目上指定的鉴权码，但跟本项目无关。

## 代码逻辑

excludeAll 作用是 webapi 的路由鉴权开关，其他对 ctx.fs 属性的注入没必要通过 excludeAll 再赋值：

```
// app/lib/middlewares/authenticator.js

let excludeAll = [];
excludeAll[0] = opts.exclude && opts.exclude === "*" ? true : false;
excludeAll[1] = opts.port;
excludeAll[2] = opts.imgPort;
excludeAll[3] = opts.parts;
excludeAll[4] = opts.overload;
excludeAll[5] = opts.airDay;
excludeAll[6] = opts.sectionWater;
excludeAll[7] = opts.toileturl;
excludeAll[8] = opts.siteurl;
excludeAll[9] = opts.xiaolanUrl;
excludeAll[10] = opts.mixingurl;

// 判断url是否不鉴权
let excluded = excludeAll[0] || excludes.isExcluded(path, method);
ctx.fs.port = excludeAll[1];
ctx.fs.imgPort = excludeAll[2];
ctx.fs.parts = excludeAll[3];
ctx.fs.overload = excludeAll[4];
ctx.fs.airDay = excludeAll[5];
ctx.fs.sectionWater = excludeAll[6];
ctx.fs.toileturl = excludeAll[7];
ctx.fs.siteurl = excludeAll[8];
ctx.fs.xiaolanUrl = excludeAll[9];
ctx.fs.mixingurl = excludeAll[10];
```

## 语法问题

**说明：**以下对语法问题进行了分类，各类均列举了个例，请检查同类问题的其他代码。

## 函数误用

### map() 错误地做遍历用

Consider using "forEach" instead of "map" as its return value is not being used here.

```
// app/lib/controllers/dataCache.js

if (rslt && rslt.length > 0) {
  types.map(type => {
    const has = rslt.filter(x => x.type == type)
    if (has.length > 0) {
      data[type] = JSON.parse(has[0].caches)
    }
  })
}
```

### eval() -> JSON.parse()

eval() 通常比其他替代方法更慢

```
// app/lib/controllers/agitator.js

var message = eval('(' + res.res.text + ')');
```

### 函数参数被重新赋值

```
// app/lib/controllers/gis/index.js

function getSquatData(list, newToken, datalist) {
  datalist = [];
  list.map(s => {
    let squaturl = `http://10.244.77.206:8084/get/toilet/${s.id}/squat?
token=${newToken}`;
    let squatData = request.get(squaturl).set("Content-Type",
"application/json");
    datalist.push(squatData);
  });
  return datalist;
}
```

## Unused code

- filter() 返回值未被使用，理解 filter() 的作用么？

```
// app/lib/controllers/ecology.js

rslt.filter(f => {
  if (f.item == item)
    return f.count++
});
```

- 未使用的包引用

```
// app/lib/controllers/agitator.js

const moment = require("moment");
```

- 未使用的变量

```
// app/lib/controllers/gis/index.js

let zhaomingObj = {}
```

- 无效的条件语句 (空的块语句)

```
// app/lib/controllers/gis/index.js

if (deviceStateCache) {

} else {

}

}
```

## 数据库访问问题

多次访问同一数据库:

```
// app/lib/controllers/cityManagementIncident.js

let one = await models.CityManagementIncidentStatistics.findOne()

let one = await models.CityManagementIncidentStatistics.findOne()

let onePre = await models.CityManagementIncidentStatistics.findOne()
```

不需要的数据库访问:

```
// app/lib/controllers/dataCache.js

let all = await models.LeaderAlarm.findAll()
```

## 条件表达式

方法调用? or 赋值?

```
// app/lib/controllers/gis/index.js

wqstationmessage.structureGroups.map(s => {
  s.structures.map(x => {
    x.stations.map(q => {
      q.factorProto.code == '8001' ? stations.push({ id: q.id, name:
q.name, longitude: q.extras.longitude, latitude: q.extras.latitude }) : ''
    })
  })
})
```

赋值语句的异味写法：

```
// app/lib/controllers/gis/index.js

aqi_day.substring(0, 1) == "0" ? aqi_day = aqi_day.substring(1, 2) : ''
```

## 变量重复声明

```
// app/lib/controllers/gis/index.js

let res = await request

let res = await request
```

## 注释

---

### 注释说明

对函数功能、**业务逻辑**等重要的信息进行必要的注释说明。

### 已废弃代码

app/lib/controllers/dataCache.js

```
// if (all && all.length > 0) {
//   order.map(x => {
//     let obj = all.find(a => a.type == x)
//     if (obj) {
//       obj.data = JSON.parse(obj.data)
//       message.push(obj)
//     }
//   })
// }
```