

# 昌南大屏 (web-v1571) 代码走查

REVIEWER: 彭玲 TIME: 2021/7/20

## 昌南大屏 (web-v1571) 代码走查

前言

npm 包优化

"exceljs": "^0.6.2"

"koa-proxy": "^0.6.0"

"koa-convert": "^1.2.0"

"koa-view": "0.0.3"

dependencies 优化

密码泄漏

接口请求问题

间接请求

重复请求

代码优化

文件清理

代码清理

console.log()

已废弃代码的注释

新手拷贝

配置固化

webapi 地址 (\_api)

项目 pcode

不使用不引用

## 前言

个例延展：对于发现的问题，需发挥主观能动性，推及当前模块的其他代码、当前项目的其他模块，以及其他项目等。

## npm 包优化

"exceljs": "^0.6.2"

"exceljs": "^0.6.2" 与 Node12+ 不兼容，导致 `ReferenceError: primordials is not defined` 问题。（必须使用 node12 以下版本。）

```
{
  message: 'primordials is not defined',
  stack: 'ReferenceError: primordials is not defined\n' +
    '    at fs.js:36:5\n' +
```

```

    '    at req_ (E:\\workSpace\\FS-
Anxinyun\\trunk\\codes\\web\\console\\node_modules\\natives\\index.js:143:24)\\n'
+
    '    at Object.req [as require] (E:\\workSpace\\FS-
Anxinyun\\trunk\\codes\\web\\console\\node_modules\\natives\\index.js:55:10)\\n'
+
    '    at Object.<anonymous> (E:\\workSpace\\FS-
Anxinyun\\trunk\\codes\\web\\console\\node_modules\\fstream\\node_modules\\grace
ful-fs\\fs.js:1:37)\\n' +
    '    at Module._compile (internal/modules/cjs/loader.js:999:30)\\n' +
    '    at Object.Module._extensions..js
(internal/modules/cjs/loader.js:1027:10)\\n' +
    '    at Module.load (internal/modules/cjs/loader.js:863:32)\\n' +
    '    at Function.Module._load (internal/modules/cjs/loader.js:708:14)\\n' +
    '    at Module.require (internal/modules/cjs/loader.js:887:19)\\n' +
    '    at require (internal/modules/cjs/helpers.js:74:18)\\n' +
    '    at Object.<anonymous> (E:\\workSpace\\FS-
Anxinyun\\trunk\\codes\\web\\console\\node_modules\\fstream\\node_modules\\grace
ful-fs\\graceful-fs.js:3:27)\\n' +
    '    at Module._compile (internal/modules/cjs/loader.js:999:30)\\n' +
    '    at Object.Module._extensions..js
(internal/modules/cjs/loader.js:1027:10)\\n' +
    '    at Module.load (internal/modules/cjs/loader.js:863:32)\\n' +
    '    at Function.Module._load (internal/modules/cjs/loader.js:708:14)\\n' +
    '    at Module.require (internal/modules/cjs/loader.js:887:19)\\n' +
    '    at require (internal/modules/cjs/helpers.js:74:18)\\n' +
    '    at Object.<anonymous> (E:\\workSpace\\FS-
Anxinyun\\trunk\\codes\\web\\console\\node_modules\\fstream\\lib\\reader.js:4:10
)\\n' +
    '    at Module._compile (internal/modules/cjs/loader.js:999:30)\\n' +
    '    at Object.Module._extensions..js
(internal/modules/cjs/loader.js:1027:10)\\n' +
    '    at Module.load (internal/modules/cjs/loader.js:863:32)\\n' +
    '    at Function.Module._load (internal/modules/cjs/loader.js:708:14)'
}

```

`"koa-proxy": "^0.6.0"`

将 koa-proxy 升级到最新版本，基于 Promise 对象。

`"koa-convert": "^1.2.0"`

koa-proxy 升级后，不再需要 koa-convert 包。

`"koa-view": "0.0.3"`

移除 `client/index.js` (无需该模块)，卸载 koa-view 包。

## dependencies 优化

区分 devDependencies 和 dependencies 应用场景，非生产环境下的依赖包不放在 dependencies 中。

```
"dependencies": {
  "babel-preset-es2017": "^6.24.1",
  "echarts": "^4.4.0",
  ...
}
```

## 密码泄漏

```
getAnxinyunTokenData = () => {
  let axyLogin = `https://console.anxinyun.cn/_api/login`;
  let loginData = {
    domain: 'anxinyun',
    password: '123456',
    username: 'changnan'
  }
  Request.postByDirect(axyLogin, loginData).then(res => {
    this.setState({ anxinyunToken: res.token, anxinyunUserId: res.id });
  })
}
```

## 接口请求问题

### 间接请求

典型地，以 `/_api/` 为前缀的请求为 web 侧对其 webapi 接口的请求，无特殊情况（如未发布 webapi 公共服务），应直接向 webapi 发送请求，而非从前端走代理。

```
let axyLogin = `https://console.anxinyun.cn/_api/login`;
```

### 重复请求

```
/// client/src/sections/default/containers/bigScreen.js

componentDidMount() {
  this.getNcxenvData();
  this.getAnxinyunTokenData();
  this.props.dispatch(getYSAccessToken());
  this.getDeviceData();

  this.getLeaderData(); //领导驾驶舱数据
}

getAnxinyunTokenData = () => {
  let axyLogin = `https://console.anxinyun.cn/_api/login`;
  let loginData = {
    domain: 'anxinyun',
    password: '123456',
    username: 'changnan'
  }
  Request.postByDirect(axyLogin, loginData).then(res => {
```

```

        this.setState({ anxinyunToken: res.token, anxinyunUserId: res.id });
    })
}

getDeviceData() {
    let axyLogin = `https://console.anxinyun.cn/_api/login`;
    let loginData = {
        domain: 'anxinyun',
        password: '123456',
        username: 'changnan'
    }
    Request.postByDirect(axyLogin, loginData).then(res => {

        let pdata = {
            "orgId": this.props.user.orgId,
            "anxinyunToken": res.token,
            "anxinyunUserId": res.id
        }
        ...
    }
}
}

```

## 代码优化

### 文件清理

- package-lock.json (若无包的特殊版本限定)

### 代码清理

#### console.log()

生产环境下，禁止使用 console() 打印调试信息，关键信息记录日志。

- 养成良好的编程习惯，利用浏览器调试工具，而非 console()

```

/// client/src/sections/default/components/gis/module/leader.js

```

```

console.log(leaderAlarms)

```

- 利用 webapck 工具

```

plugins: [
    new webpack.optimize.UglifyJsPlugin({
        comments: false, // 删除所有的注释
        compress: {
            drop_console: true // 删除所有的 `console` 语句
        }
    }
),
]

```

## 已废弃代码的注释

同样，养成良好的编程习惯，对注释勿滥用，仅对必要的有用信息进行注释说明。（利用 webapck 工具，见上面配置。）

```
/// client/src/layout/index.js

// setInterval(() => {
//     console.log('src-layout-index.js-循环打印当前sotre值', store.getState());
// }, 10000);
```

## 新手拷贝

先理解，再内化为己用，切勿无脑地拷贝粘贴。

## 配置固化

### webapi 地址 (`_api`)

- 固化 webapi 地址有何特殊考虑？
- 是否理解项目中使用 `_api` 的目的？
- webapi 部署地址变动后，修改代码 - 重新构建？

```
/// client/src/utils/webapi.js

// const rootUrl = '_api';
const rootUrl = 'http://localhost:4000'; //本地开发
// const rootUrl = 'https://changnan.anxinyun.cn/api'
// const rootUrl = 'https://changnandemo.anxinyun.cn/api'
export const buildUrl = url => {
    const apiUrl = `${rootUrl}/${url}`; //本地开发
    // const apiUrl = `${rootUrl}/${url}`;
    const user = JSON.parse(sessionStorage.getItem('user'));
    if (user == null) {
        return apiUrl;
    }
    let connector = url.indexOf('?') === -1 ? '?' : '&';
    return `${apiUrl}${connector}token=${user.token}`;
};
```

### 项目 pcode

```
/// client/src/sections/auth/actions/auth.js

export function login(username, password) {
    // let pcode = "1ff1353f-c704-4765-a1fe-943b70538f61"; //环境网络-商用
    let pcode = "06af6e8e-b6aa-427b-96c2-e5a4d0371c75"; //商用
    //let pcode = 'c52ca637-2330-4651-b900-3b5730a8e832'; //测试
}
```

## 不使用不引用

```
/// routes/video/curPositon.js

/// routes/video/heartBeat.js

/// routes/video/PTZControl.js

/// routes/video/real-time-video.js

/// routes/video/video-play-back.js

const proxy = require('koa-proxy');
const convert = require('koa-convert');
```