安心云前端代码走查

2021/4/15

安心云前端代码走查

- 前言
- 基本规范
  - 调试基本
    - 为什么要用 console ?
  - SVN 提交规范性
    - 日志文件
- 开发环境规范性
  - app 启动参数完整性
  - app 启动参数优化
    - 不必要不保留
      - 已废弃参数
      - 注释
    - 不必要不配置
- 生产环境
  - Dockerfile
- web-console
  - 控制台 Warning
  - 魔数
- web-api
  - 僵尸代码
    - 配置项
    - 接口
  - 代码设计
    - 热点图接口

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。

基本规范

调试基本

为什么要用 console ?

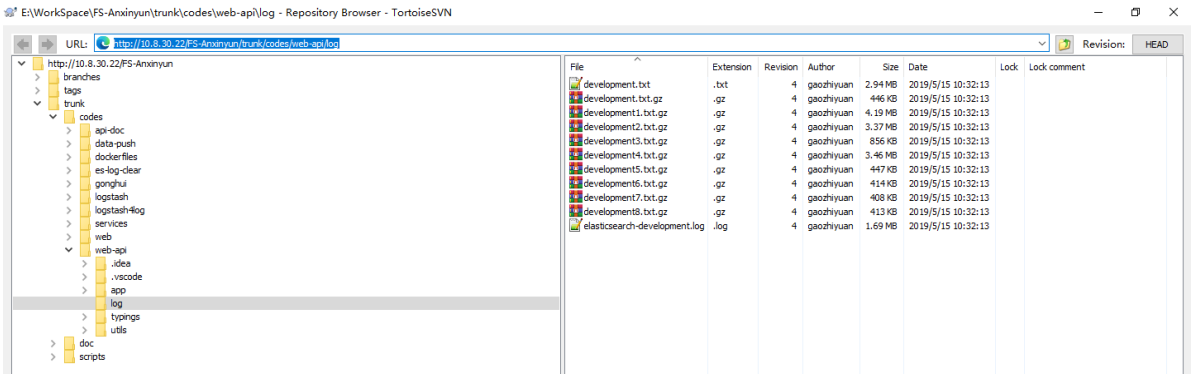
SVN：http://10.8.30.22/FS-Anxinyun/trunk/codes/web-api/app/lib/controllers/auth.js

```
async function login(ctx, next) {
    // Line 90
    console.log(1);
}
```

SVN 提交规范性

日志文件

> SVN：http://10.8.30.22/FS-Anxinyun/trunk/codes/web-api/log



开发环境规范性

> 避免给他人带来不必要的迷惑和困扰。

app 启动参数完整性

程序启动的必要参数是否均已输出至以下两个文件：

- **launch.json**：VSCode 调式环境需要
- **package.json**：外部启动环境需要

app 启动参数优化

不必要不保留

已废弃参数

SVN：http://10.8.30.22/FS-Anxinyun/trunk/codes/web/project/.vscode/launch.json

以下参数已废弃：

```
"--webhdfs http://10.8.30.35:50070",
"--dxappid 03d9d7b037d94899abeeebe580b7b3cb",
"--dxsecret ab37810618c74beb87d985d7acbbd9cd"
```

> 后果：使用`const args = require('args')`解析参数，导致`args.parse(process.argv)`解析失败。

```
# args.parse(process.argv) failed.
The option "fft" is unknown. Here's a list of all available options:

Usage: server.js [options] [command]

Commands:

 help  Display help

Options:

 -e, --es                  es服务URL
 -E, --es-type               es-type名称
```

说明：

- `node server.js <args>` 指定的启动参数必须使用 `args.option()` 确认
- 可以使用 `args.option()` 直接指定启动参数，例如：`args.option(['p', 'port'], '启动端口', 8080)`

注释

SVN：http://10.8.30.22/FS-Anxinyun/trunk/codes/web/project/config.js

```
// args.option('analysis', '数据分析工具地址')
```

不必要不配置

优化目标 (尽可能少地传递参数，减少繁琐配置)：项目中，必填参数给默认值 (无需判断存在性) 。

```
// 启动参数
args.option(['p', 'port'], '启动端口');
args.option(['x', 'iota-proxy'], 'iota代理URL');
args.option(['w', 'webhdfs'], 'webhdfsURL');
args.option(['k', 'kafka'], 'kafka服务URL');
args.option(['e', 'es'], 'es服务URL');
args.option('es-type', 'es-type名称');
args.option('platform-name', 'ES index 平台前缀');
args.option(['g', 'pg'], 'postgre服务URL');
args.option('redis', 'redis服务URL');

args.option('qnak', 'qiniuAccessKey');
args.option('qnsk', 'qiniuSecretKey');
args.option('qnbkt', 'qiniuBucket');
args.option('qndmn', 'qiniuDomain');
args.option('fc-svr-orinin', '文件中心本地化存储: WebApi 服务器地址(必填), 该服务器提供文件上传web服务');
args.option('fc-upload-path', '文件中心默认上传目录(可选), attachment.upload() API 可指定上传目录');
args.option('fc-local-root-path', '文件中心本地化存储: 服务器工作目录下的根路径(可选)');
args.option('fc-local-child-path', '文件中心本地化存储: 服务器工作目录下的子路径(可选)');

const flags = args.parse(process.argv);
```

```
const ANXINCLOUD_PROXY_IOTA = process.env.ANXINCLOUD_PROXY_IOTA ||
flags.iotaProxy;
const ANXINCLOUD_HDFS_HTTP = process.env.ANXINCLOUD_HDFS_HTTP || flags.webhdfs;
const ANXINCLOUD_KAFKA_BROKERS = process.env.ANXINCLOUD_KAFKA_BROKERS ||
flags.kafka;
const ANXINCLOUD_KAFKA_TOPIC = process.env.ANXINCLOUD_KAFKA_TOPIC ||
flags.kafkaTopic;
const ANXINCLOUD_ES_NODES_REST = process.env.ANXINCLOUD_ES_NODES_REST ||
flags.es;
const ANXINCLOUD_DB = process.env.ANXINCLOUD_DB || flags.pg;
const ANXINCLOUD_REDIS = (process.env.ANXINCLOUD_REDIS_HOST &&
process.env.ANXINCLOUD_REDIS_PORT) ?

 `redis://:${process.env.ANXINCLOUD_REDIS_PASSWD}@${process.env.ANXINCLOUD_REDIS
_HOST}:${process.env.ANXINCLOUD_REDIS_PORT}` : flags.redis
const ANXINCLOUD_PYRPC_URL = process.env.ANXINCLOUD_PYRPC_URL ||
'http://anxinyun-pyrpc:15000/api';

const PLATFORM_NAME = process.env.PLATFORM_NAME || flags.platformName ||
'anxinyun';

const ANXINCLOUD_QINIU_DOMAIN_QNDMN_RESOURCE =
process.env.ANXINCLOUD_QINIU_DOMAIN_QNDMN_RESOURCE || flags.qndmn;
const ANXINCLOUD_QINIU_BUCKET_RESOURCE =
process.env.ANXINCLOUD_QINIU_BUCKET_RESOURCE || flags.qnbkt;
const ANXINCLOUD_QINIU_AK = process.env.ANXINCLOUD_QINIU_AK || flags.qnak;
const ANXINCLOUD_QINIU_SK = process.env.ANXINCLOUD_QINIU_SK || flags.qnsk;
const ANXINCLOUD_FC_UPLOAD_PATH = process.env.ANXINCLOUD_FC_UPLOAD_PATH ||
flags.fcUploadPath;
const ANXINCLOUD_FC_LOCAL_SVR_ORIGIN =
process.env.ANXINCLOUD_FC_LOCAL_SVR_ORIGIN || flags.fcSvrOrigin;
const ANXINCLOUD_FC_LOCAL_ROOT_PATH = process.env.ANXINCLOUD_FC_LOCAL_ROOT_PATH
|| flags.fcLocalRootPath;
const ANXINCLOUD_FC_LOCAL_CHILD_PATH =
process.env.ANXINCLOUD_FC_LOCAL_CHILD_PATH || flags.fcLocalChildPath;

const ANXINCLOUD_FC_CLOUD = !!(ANXINCLOUD_QINIU_AK && ANXINCLOUD_QINIU_SK &&
ANXINCLOUD_QINIU_BUCKET_RESOURCE && ANXINCLOUD_QINIU_DOMAIN_QNDMN_RESOURCE);

if (!ANXINCLOUD_PROXY_IOTA || !ANXINCLOUD_HDFS_HTTP || !ANXINCLOUD_KAFKA_BROKERS
|| !ANXINCLOUD_ES_NODES_REST || !ANXINCLOUD_DB || !ANXINCLOUD_REDIS) {
    console.log('缺少启动参数，异常退出');
    args.showHelp();
    process.exit(-1);
}
```

生产环境

Dockerfile

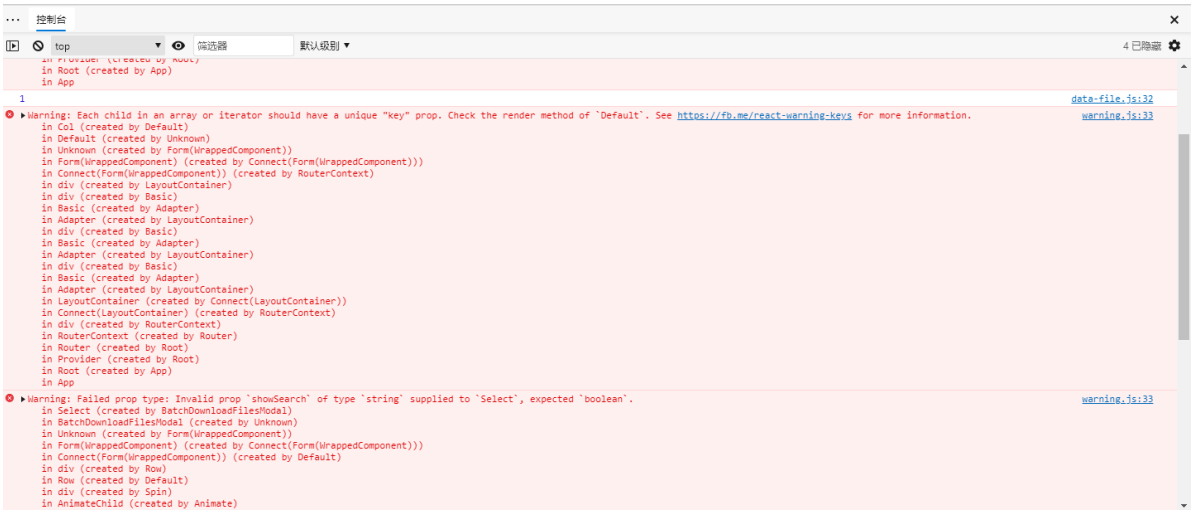
目前 Jenkins 构建环境使用 `nodejs-12`，因此，各项目下 `Dockerfile` 对应的 Docker 镜像源应修改为 `nodejs-12` 版本：

```
# 以前构建 nodejs-8 对应：FROM node:8-alpine
FROM repository.anxinyun.cn/base-images/nodejs12:20.10.12.2
```

web-console

控制台 Warning

代码不规范引入的告警信息：



魔数

> SVN：http://10.8.30.22/FS-Anxinyun/trunk/codes/web/console/client/src/sections/project-monitor/components/things/newlyBuild.js

```
<NewlyBuild currentState={0} />
<NewlyBuild currentState={1} />
```

web-api

僵尸代码

配置项

config.js 原代码中配置项多余（已删除无用配置项）：

```
{
    entry: require('./app').entry,
    opts: {
        qiniu: {
            // accessKey: ANXINCLOUD_QINIU_AK,
            // secretKey: ANXINCLOUD_QINIU_AK,
            // bucket: ANXINCLOUD_QINIU_BUCKET_RESOURCE,
            domain: ANXINCLOUD_QINIU_DOMAIN_QNDMN_RESOURCE
        },
    }
}
```

接口

- SVN：http://10.8.30.22/FS-Anxinyun/trunk/codes/web-api/app/lib/routes/alarm/index.js

```
/**
 * @api {POST} /alarms/download 导出告警数据.
 * @apiVersion 1.0.0
 * @apiGroup Alarm
 */
app.fs.api.logAttr['POST/alarms/download'] = { content: '导出告警数据', visible:
true };
router.post('/alarms/download', alarm.download(opts.alarmTypeCode));
```

- SVN：http://10.8.30.22/FS-Anxinyun/trunk/codes/web-api/app/lib/routes/pan/index.js

```
/**
 * @api {GET} /structures/:id/hdfs-file 下载组织数据文件.
 * @apiVersion 1.2.0
 * @apiGroup DataFile
 */
app.fs.api.logAttr['GET/hdfs-file/structures/:id/download'] = { content: '下载组织
数据文件', visible: true };
app.fs.auth.authAttr['GET/hdfs-file/structures/:id/download'] =
AuthorizationCode.DownloadFile;
router.get('/hdfs-file/structures/:id/download', dataFile.download);
```

代码设计

热点图接口

2D/3D 热点图增删改查接口中，对七牛云文件管理可优化，优化后，**无需再配置七牛域名**：

> SVN：http://10.8.30.22/FS-Anxinyun/trunk/codes/web-api/app/lib/controllers/station-deploy.js

关键代码：

```
/**
 * @api {Get} /heatmaps/:heatmapId/stations/deploy?factorId=id 获取组织下全部热点图
测点部署信息.
 * @apiVersion 0.1.0
 * @apiGroup StationDelay
```

```
 */
app.fs.api.logAttr['GET/organization/:orgId/heatmaps/stations/deploy'] = {
content: '获取组织下全部热点图测点部署信息', visible: true };
router.get('/organization/:orgId/heatmaps/stations/deploy',
station_deploy.findOrgStations(opts));

function findOrgStations(opts) {
    return async function (ctx) {

        let model_ = {
            id: m.id,
            typeId: m.type,
            name: m.params.name,
            portrait: opts.qiniu.domain + '/' + m.params.portrait, //
@Optimizable 上传接口中处理了："url"中包含了 qiniu.domain>
        };

    }
}
```