# 部署

## 资源

资源包下载地址：\\192.168.0.180\软件\GIS\MapGo -> MapGo-1.3.2.zip

注：

* 该地址为公司内部window共享资源，打开方式请参考 [window系统怎么访问其他电脑共享文件夹](http://www.win7zhijia.cn/jiaocheng/win7_25204.html)
* 打开共享文件夹时需要公司域账户登录

word版部署文档下载地址：

[MapGo-1.3.2部署文档.docx](../files/MapGo-1.3.2部署文档.docx)

[MapGo-1.3.1部署文档.docx](../files/MapGo-1.3.1部署文档.docx)

## 基础环境要求

* 操作系统：window 7 或者 window 10（推荐）
* Node版本：11+（长期支持版本）

## 部署步骤

### 下载资源包

资源包中主要包含两个程序：

* Nginx 程序文件

├── nginx-1.18.0 // Nginx  
 │ ├── conf // Nginx 配置文件夹  
 │ │ ├── nginx.conf // Nginx 主要配置文件（端口、代理等）  
 │ │ └── ...  
 │ ├── html // 默认资源文件夹（即默认ip:port/ 指向的路径）  
 │ │ ├── mapgo // 平台代码文件夹  
 │ │ └── ...  
 │ └── ...

* MapGo 程序文件

├── mapgo // 地图可视化配置平台程序  
 │ ├── apps // 用户应用配置文件夹  
 │ │ ├── user // 用户名文件，保存该用户的应用配置  
 │ │ │ ├── 2D // 二维应用配置文件夹  
 │ │ │ │ ├── applicationID // 应用ID文件夹，保存某个应用的具体配置  
 │ │ │ │ │ ├── config.json // 地图配置文件（底图、微件等配置）  
 │ │ │ │ │ ├── interactions.js // 开发者代码文件  
 │ │ │ │ │ ├── LayerManager // 图层列表配置文件夹  
 │ │ │ │ │ │ └── config.json // 具体图层配置  
 │ │ │ │ │ ├── LegendManager // 图例配置文件夹  
 │ │ │ │ │ │ └── config.json // 图例具体配置  
 │ │ │ │ │ └── TimeLineManager // 时间轴配置文件夹  
 │ │ │ │ │ └── config.json // 时间轴具体配置  
 │ │ │ │ └── ...  
 │ │ │ ├── 3D   
 │ │ │ │ ├── applicationID // 应用ID文件夹，保存某个应用的具体配置  
 │ │ │ │ │ ├── config.json // 地图配置文件（底图、微件等配置）  
 │ │ │ │ │ ├── interactions.js // 开发者代码文件  
 │ │ │ │ │ ├── LayerManager // 图层列表配置文件夹  
 │ │ │ │ │ │ └── config.json // 具体图层配置  
 │ │ │ │ │ ├── LegendManager // 图例配置文件夹  
 │ │ │ │ │ │ └── config.json // 图例具体配置  
 │ │ │ │ │ └── TimeLineManager // 时间轴配置文件夹  
 │ │ │ │ │ └── config.json // 时间轴具体配置  
 │ │ │ │ └── ...  
 │ │ │ ├── catalog.json // 用户目录结构  
 │ │ │ └── log\_\*\*.log // 用户操作记录  
 │ │ ├── 2D.json // 二维应用记录   
 │ │ ├── 3D.json // 三维应用记录  
 │ │ └── transfromApps\_\*\*.log // 1.3.2之前版本配置文件结构向1.3.2版本（及以后）转换过程中的变化记录  
 │ ├── gis\_framework // 二维地图框架  
 │ ├── icons // 图标库  
 │ ├── public // 模板库  
 │ │ ├── echarts\_template // Echarts 模板保存位置  
 │ │ │ ├── images // 模板展示效果图片库  
 │ │ │ │ └── ... // 模板展示效果图片（以模板id为文件名称）  
 │ │ │ └── template.json // 模板具体配置  
 │ │ ├── html\_template // HTML 模板保存位置  
 │ │ │ ├── template\_id // 某个模板所能使用的图片  
 │ │ │ │ └── ... // 具体图片  
 │ │ │ └── template.json // 模板具体配置  
 │ ├── service // 平台后台代码  
 │ │ ├── main.js // 后台 主要代码  
 │ │ ├── config.js // 后台配置文件（服务端口和登录接口处理等）  
 │ │ └── ...  
 │ ├── config.js // 平台配置文件（设置后台端口、公告等）  
 │ └── ...

注： 具体文件目录请看附录一

### 平台后端服务部署

::: warning  
如果没有 Node 环境，请先自行安装 [Node](https://nodejs.org/zh-cn/) 。  
:::

* 在 service 目录下启动 cmd , 运行 npm install 安装依赖。
* 打开 config.js , 设置服务的端口
* 在文档中寻找如下代码并修改

port: 3007, // 服务端口，默认 3007

注： 默认为 3007，如果有端口占用，请选择其他端口

* 设置登录验证和获取所有用户方式
  + OU：公司域账户登录
  + PLATFORM： 内置公司大屏登录和获取所有用户接口（采用的为比较早期的方式，不一定适用）
  + CUSTOM：自定义，具体示例请直接看 **config.js** 文件
* 启动测试

运行：在 services 目录下，启动 cmd ，运行命令 node main.js

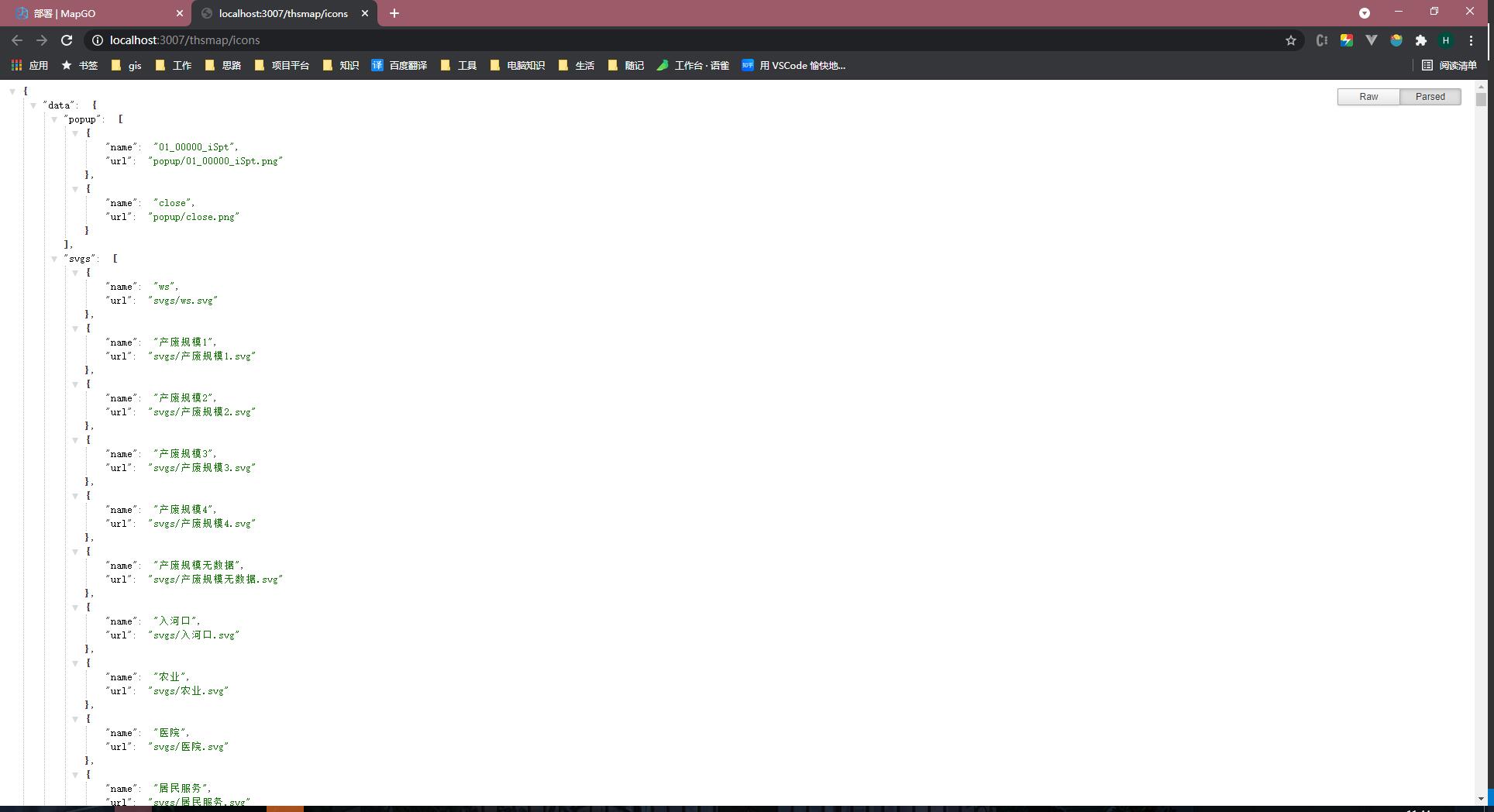
:::tip  
如果启动过程中提示如下信息  
当前的应用结构是 Node 版的，需要向 NestJS 转换，该操作会导致现有应用配置文件发生不可逆操作，请确定您已经把应用配置文件备份后在执行！

您现在已经备份好现有应用配置文件了吗？（Y/N）

请看[附录二](#header-n119)  
:::

测试接口：<http://localhost:3007/thsmap/icons>

成功结果：



注：

* 如果您在上步中修改了服务的接口，请把 3007 修改为对应的端口号

### Nginx 部署

::: tip

1. 资源包中默认提供了 Nginx 文件，并且默认配置了相关代理
2. 如果您不想使用提供的 Nginx , 您把下方两个代理和 mapgo 文件夹正确的部署到您的 Nginx 中即可
3. 以下代理均为默认后台端口，有需要修改为对应的即可
4. 因为 Nginx 和 后台 是在同一台服务器上，所以可以使用 localhost，如果不是，请改为具体 IP  
   :::

#### Nginx 端口设置

在 nginx-1.18.0 -> conf -> nginx.conf 文件中找到 listen 配置项

# http -> server -> listen  
server {  
 # Nginx 端口设置，默认为10229,端口占用修改即可  
 listen 10229;  
 server\_name localhost;  
 charset utf-8;  
 server\_tokens off;  
 #charset koi8-r;  
 # ...  
}

#### Node 服务代理

# 代理 MapGo 平台后端接口  
location /thsmap/ {  
 proxy\_pass http://localhost:3007/thsmap/;  
 # add\_header 'Access-Control-Allow-Origin' '\*';  
 # add\_header Access-Control-Allow-Headers X-Requested-With;  
 # add\_header Access-Control-Allow-Methods GET,POST,OPTIONS;  
 # proxy\_connect\_timeout 300s;  
}

#### WebScokt 代理

# 代理 MapGo 后端 WebSocket 接口  
location /socket.io/ {  
 proxy\_pass http://localhost:3007/socket.io/;  
  
 proxy\_http\_version 1.1; # 告诉nginx使用HTTP/1.1通信协议，这是websoket必须要使用的协议  
 proxy\_set\_header Upgrade $http\_upgrade; # 当它想要使用WebSocket时，响应http升级请求  
 proxy\_set\_header Connection "upgrade"; # 当它想要使用WebSocket时，响应http升级请求  
}

#### 启动测试

#### 启动

在 nginx-1.18.0 目录下，启动 cmd，运行命令 start nginx

#### 测试

在浏览器地址栏中访问 <http://localhost:10229/thsmap/icons>

返回结果和 <http://localhost:3007/thsmap/icons> 一致即可

注：端口有修改请自行改为相应的即可

### 平台配置

将 mapgo 下的 config.js 的 HOST、nodeHOST、proxyURL 这四个配置修改为 Nginx 代理之后的地址

:::tip  
配置中的ip和端口是指在浏览器中能指向项目部署的IP和端口  
:::

const appConfig = {  
  
 appPath: 'mapgo', // IP:port/${appPath}/apps (路径有的话需要在最后加 /)  
  
 HOST: '121.46.19.47:10229', // 项目部署的 ip + port  
  
 nodeHOST: '121.46.19.47:10229', // node服务的 ip + port  
  
 enableDownload: true, // 是否启用下载功能  
  
 proxyURL: 'http://121.46.19.47:10229/thsmap/proxy', // GIS框架的代理地址,  
  
 isUpdateProxyURL: true, // 是否每次都更新配置文件中的代理地址  
  
 enableVerificationCode: false, // 是否启用验证码  
  
 isGISFrameBuild: true, // GIS框架是否打包  
  
 // 全局替换接口中的某些字段（一般是迁移的时候，地址有统一变动时用，注意：应用只有再次保存的时候，地址才会真的被修改）  
  
 // 替换范围：1.框架配置文件（代理地址不参与修改）2.图层 3.图例 4.开发者代码  
  
 enableURLReplaceStr: {  
  
 enable: false, // 是否启用  
  
 replaceResources: [{  
  
 replaceStr: '', // 要替换的字符串  
  
 replacedStr: '', // 替换后的字符串  
  
 }],  
 },  
  
 helpDocLink: 'http://121.46.19.2:20724/MapGo\_doc/', // 可视化平台帮助文档地址  
  
 noticeOption: { // 公告配置  
  
 nable: false,  
  
 notices: [  
 {  
 content: '公告内容',  
   
 startTime: '2021-01-05 11:00', // 公告开始提示时间  
  
 endTime: '2021-01-08 18:30', // 公告停止提示时间  
 }  
 ],  
 },  
};

### 平台测试运行

确保 Nginx 和 Node 正常运行，访问 <http://121.46.19.47:10229/mapgo/#/Login>

输入正确的账号密码能进应用列表界面即可





## 注意事项

* 平台升级时，采用的是文件替换的方式，所以您的自定义代码需要自行替换调试

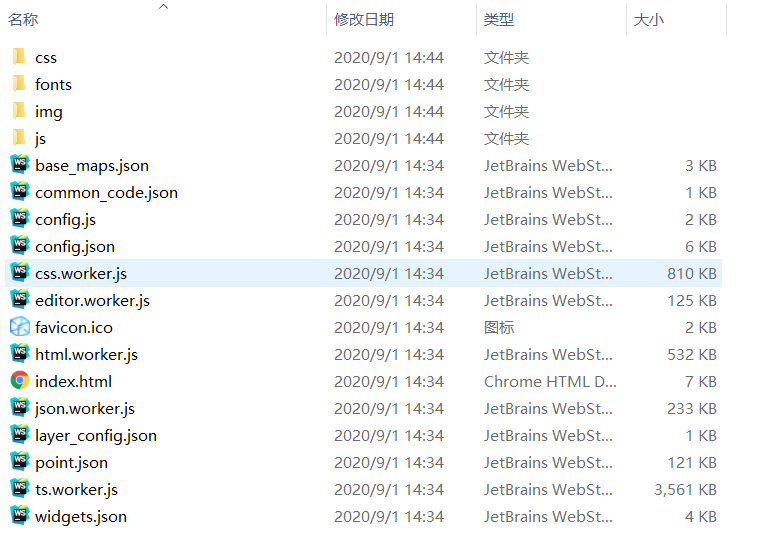
## 意见反馈

如果您有任何关于平台部署的疑问，请直接联系公司 GIS 组，我们将会研究解决您的问题并及时更新部署文档来方便您的使用，感谢您的支持，谢谢 😜！

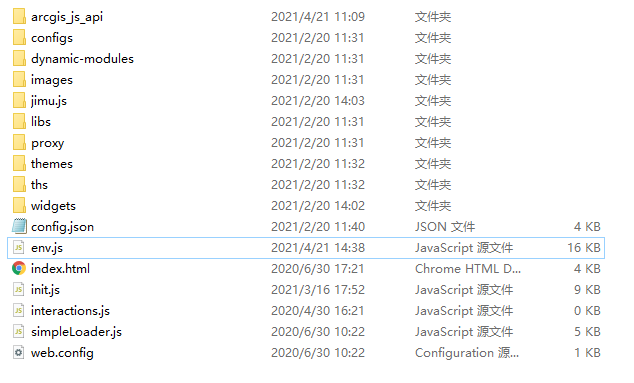
## 附录

### 附录一

#### 配置平台包



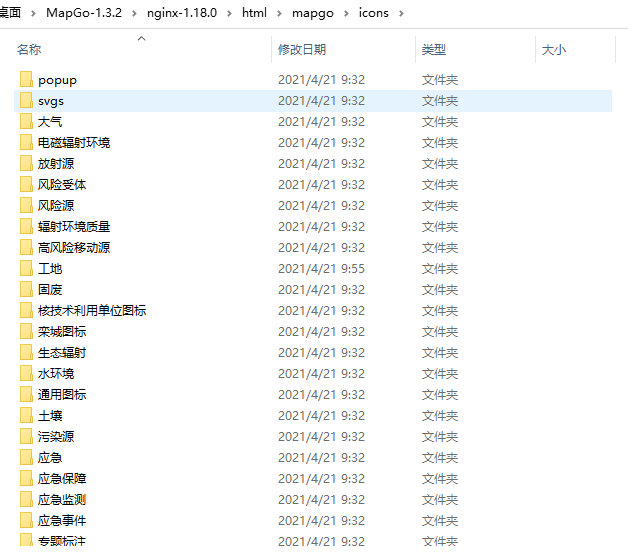
#### 地图框架包



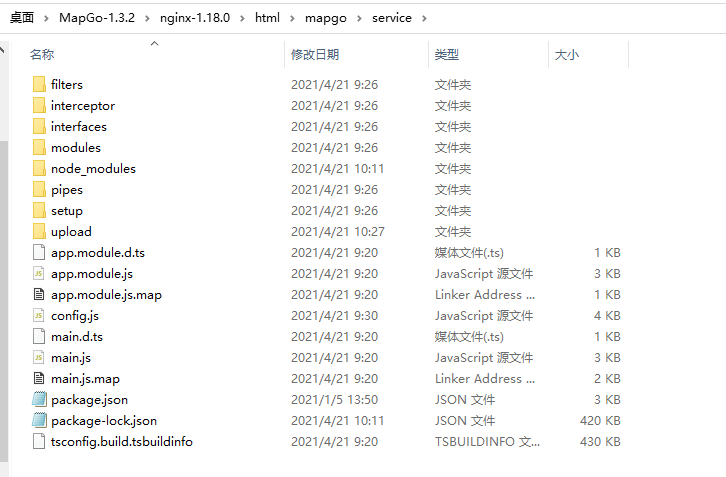
#### 图层样式模板



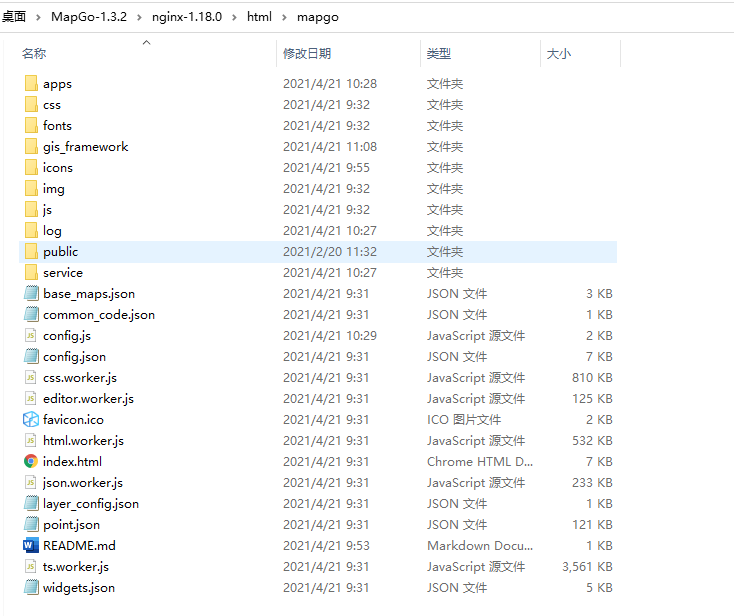
#### 图标库



#### Node 服务



#### 完整资源目录



### 附录二

这里将说明后台程序启动时发生的情况，主要适用于从1.3.1及以前版本向1.3.2及以上版本升级时进行说明

#### 原因

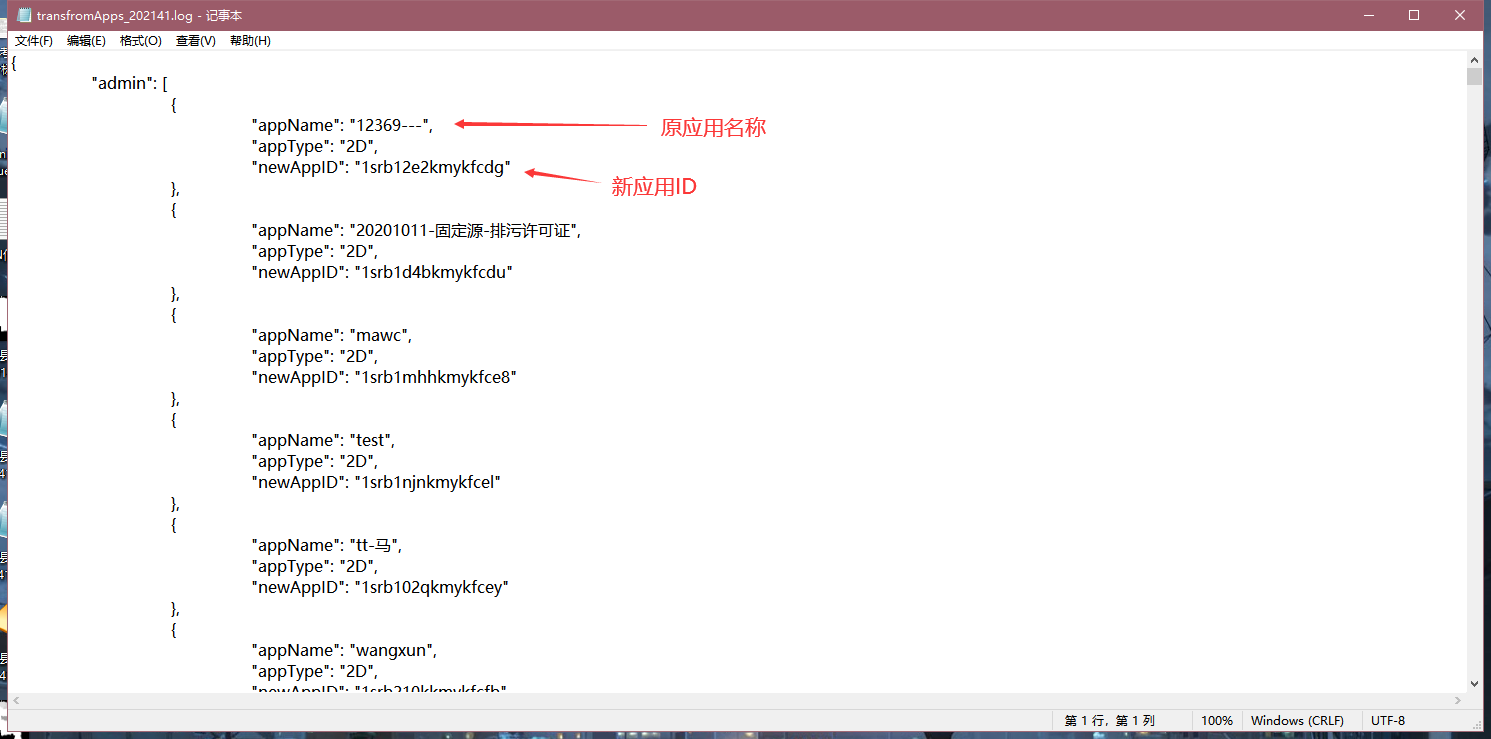
因为在1.3.2的时候，进行了配置文件结构和图层配置的改变，所有需要在后台启动的时候进行文件和配置进行适应性修改

#### 解决

##### 低版本向高版本升级

* 先把平台进行完整的备份，尤其是apps
* 重启新的后台程序，在提示备份时，输入Y，尝试进行文件转换
* 如果成功，请进入系统测试，如果失败，请联系GIS组

升级成功后，在apps文件夹下会生成文件 transfromApps\_生成日期.log，里面保存了转换前后应用文件路径的变化，可以根据该文档修改其他正在使用平台预览地址的路径



##### 新部署

因为这种情况的判定条件时apps文件夹下是否存在2D.json文件，所以，直接输入Y即可