



ThingJS 项目离线部署包 用户手册

(Rev.3.2.13)

北京优锆科技有限公司

2023 年 5 月 8 日

目 录

1 概述	4
1.1 说明	4
1.2 硬件配置	4
1.3 离线部署包类型说明	5
2 获取	7
2.1 准备 CPS 文件	7
2.2 申请下载	10
3 安装	14
3.1 Windows 服务器	14
3.2 Linux 服务器	15
4 配置	17
4.1 修改服务启动端口	17
4.2 配置 HTTPS 访问（可选）	18
4.3 修改登录密码	18
4.4 瓦片地址服务配置	19
4.5 倾斜摄影数据配置	22
4.6 单服务器多项目部署配置	23
4.7 服务请求配置	25
4.8 管理页面访问白名单配置	25
4.9 资源请求压缩配置	26
5 授权	27
5.1 访问“项目离线部署管理”界面	27
5.2 授权“临时部署包”	28
5.3 授权“永久部署包”	32
6 更新	33
6.1 获取“更新部署包”	33

6.2 更新项目	33
7 升级	35
7.1 Windows 服务器	35
7.2 Linux 服务器	39
8 迁移	43
8.1 获取新机器码文件	43
8.2 获取迁移码文件	43
8.3 获取新授权码文件	44
8.4 完成新服务器的授权	46
9 日志	47
9.1 查看日志	47
9.2 下载日志	48
10 常见问题 (FAQ)	49
10.1 在线开发正常运行项目，离线部署后访问 127.0.0.1:9000，3D 场景展示出错	49
10.2 Windows 环境下，start.exe 启动失败	49
10.3 Windows 环境下，start.exe 卡住，服务无反应	49
10.4 地图瓦片和离线前不一致	49
10.5 场景、模型未离线打包	50
10.6 360 解压离线部署包有问题	52
10.7 倾斜摄影数据访问不到，控制台报错 404	53
10.8 ChartBuilder 项目离线部署后看不到图表内容	53
10.9 ChartBuilder 项目离线部署后显示 127.0.0.1 拒绝了连接请求	53
10.10 ThingJS 项目离线部署包 V3.2.0 版本与旧版本相比，有什么区别	54
10.11 更新项目后，控制台接口 “/get/PublicKey” 报 404	55
10.12 服务配置在二级路由下，控制台报错 404	55
10.13 森城市携带场景离线部署后场景加载不到该如何配置	56

10.14 加载森拓扑资源时，有资源报 404 该如何解决	57
10.15 修改配置文件不生效如何解决	58
10.16 森大屏资源加载不出来	58
10.17 天空盒资源仍请求线上地址	59
10.18 森大屏资源离线加载时，数据对接错误	59
11 错误码对照表	61
12 修订历史	62

1 概述

本文档是一个对 ThingJS 项目离线部署包的获取、安装、配置、授权、更新、迁移等方面进行说明的技术文档。

本文档会以 ThingJS 平台“在线开发”为默认环境进行介绍。涉及 ThingJS 离线开发环境的内容会将与“在线开发”差异部分在各章节内容中增加小节补充说明。

1.1 说明

在 ThingJS 3D 可视化开发平台“在线开发”(<https://www.thingjs.com>)、ThingJS 离线开发等环境完成项目开发后，开发人员可将所完成开发的项目离线部署到自己的私有服务器上。这时候就会使用到该项目的“ThingJS 项目离线部署包”。

“ThingJS 项目离线部署包”可由已开通“VIP 商业开发者”或购买“离线部署永久授权”的账号从“在线开发”环境进行下载；或由 ThingJS 离线开发环境生成。

ThingJS 免费用户无法下载项目离线部署包。当然，虽然免费用户不能下载离线部署包，但他们可在 ThingJS 平台完整测试 ThingJS API 调用及大部分功能。

(注：一般商业项目的开发和部署，建议通过购买 ThingJS 3D 可视化开发平台的“ThingJS 3D 可视化套餐”(<https://www.thingjs.com/guide/price/>)来完成，以提高商业项目开发的便捷度和效率。)

1.2 硬件配置

1.2.1 安装服务器推荐配置

	配置
CPU	2G 主频 4 核
内存	8G
硬盘	80G, 推荐 SSD
带宽	推荐 100Mbps
操作系统	Linux x64, Windows x64

(注：(1)若需在 CPU 为 ARM 的 Linux 服务器上进行部署，请先咨询平台客服；
(2)不支持在 Docker 中进行部署。)

1.2.2 浏览器客户端推荐配置

	配置
CPU	Intel i7 或同档 CPU 以上
内存	16G
硬盘	80G, 推荐 SSD
显卡	独显 GTX1070 以上, 2G 显存以上

1.3 离线部署包类型说明

“ThingJS 项目离线部署包”分为“临时部署包”和“永久部署包”两种类型。

1.3.1 临时部署包

“临时部署包”仅用于需项目离线部署的用户在离线部署环境测试之用。用户若需在私有服务器部署商业或正式应用的，请使用“永久部署包”。

已开通“VIP 商业开发者”的账号可在“在线开发”中获取“临时部署包”；购买“离线开发授权”的用户可在离线开发环境生成项目的“临时部署包”。

“临时部署包”在完成授权后（自助在 ThingJS 网站获取授权码），默认有效期为 7 天；“临时部署包”的 3D 可视化项目画面背景中会显示 ThingJS 官方水印；7 天授权过期后，原“临时部署包”访问链接将自动显示项目离线部署管理页面，无法再正确加载原 3D 可视化项目画面。此时，用户可再次申请一个临时授权码，或升级到“永久部署包”重新激活来使用。

“ThingJS 项目离线部署包”含“更新项目”功能。无论 3D 场景变更、脚本变更、资源变更等，开发者都可对已部署项目进行更新操作。但“更新项目”功能对“临时部署包”项目是有限制的。对于用“临时部署包”部署的项目，若需要更新的，只能使用“永久部署包”进行项目更新。完成更新后，该“临时部署包”部署的项目也会自动升级为“永久部署包”项目。也就是说，“临时部署包”项目无法通过另一个“临时部署包”来完成更新。

实际上，“临时部署包”不支持作为“更新部署包”进行任何项目更新。无论原部署项目是“临时部署包”项目，还是“永久部署包”项目均不支持。

（注：“临时部署包”升级为“永久部署包”时，机器码文件将自动更新升级，会导致当前授权失效。此时，需申请永久授权码方可再次激活授权。）

1.3.2 永久部署包

只有购买“离线部署永久授权”的账号才可获取“永久部署包”。

“永久部署包”必须使用“永久授权码”进行授权激活（首次获取需联系平台商务人员）。完成授权后授权将永久有效，同时在 3D 可视化项目画面背景中也不会显示 ThingJS 官方水印。

“永久部署包”支持作为“更新部署包”进行项目更新。“永久部署包”在授权成功后，将拥有“服务器迁移”功能。

(注：如需购买“离线部署永久授权”，请咨询在线客服，或者拨打 ThingJS 平台客服电话 400-666-9832。)



2 获取

2.1 准备 CPS 文件

需使用 ThingJS 平台项目离线部署包功能，需先开通了 VIP 商业开发授权。免费用户无法下载项目离线部署包。

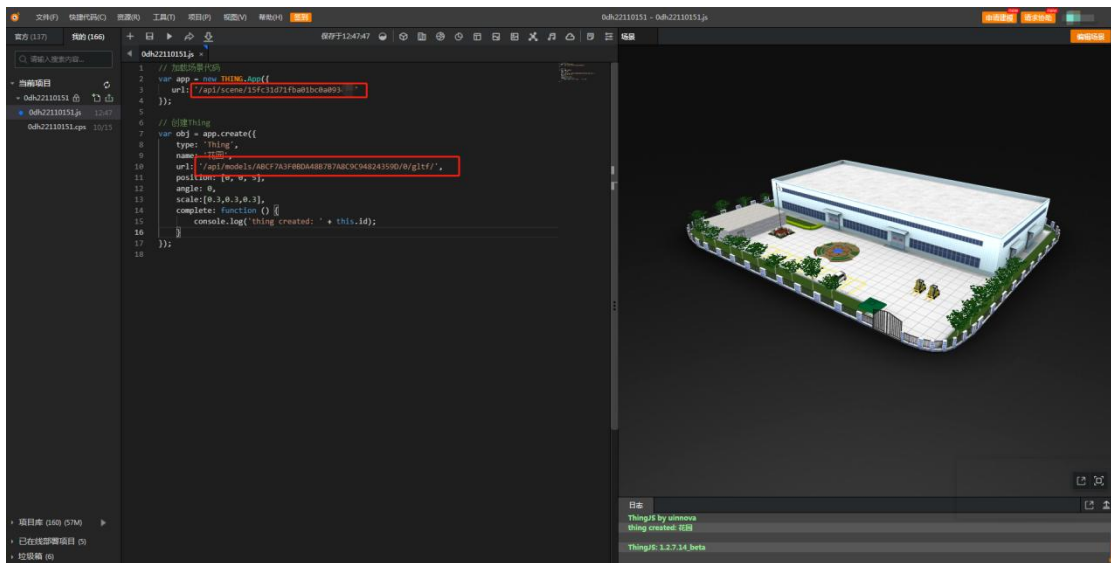
当 VIP 开发者在 ThingJS 平台“在线开发”环境完成一个项目开发，准备进行项目离线部署包下载时，需了解一下项目部署包配置文件：CPS 文件。

一般情况下，ThingJS 平台的离线部署包生成程序会自动打包项目所引用的场景和模型，但因为开发人员程序写法多样，离线部署包生成程序很难一一识别这些实际需要打包的场景、模型，所以“在线开发”环境给每个项目都增加了一个 CPS 配置文件，辅助项目离线部署包打包。

为确保项目中所引用的场景和模型能够被完整离线打包，需在在线开发中自行配置项目对应的 CPS 文件后再进行“ThingJS 项目离线部署包”的申请。

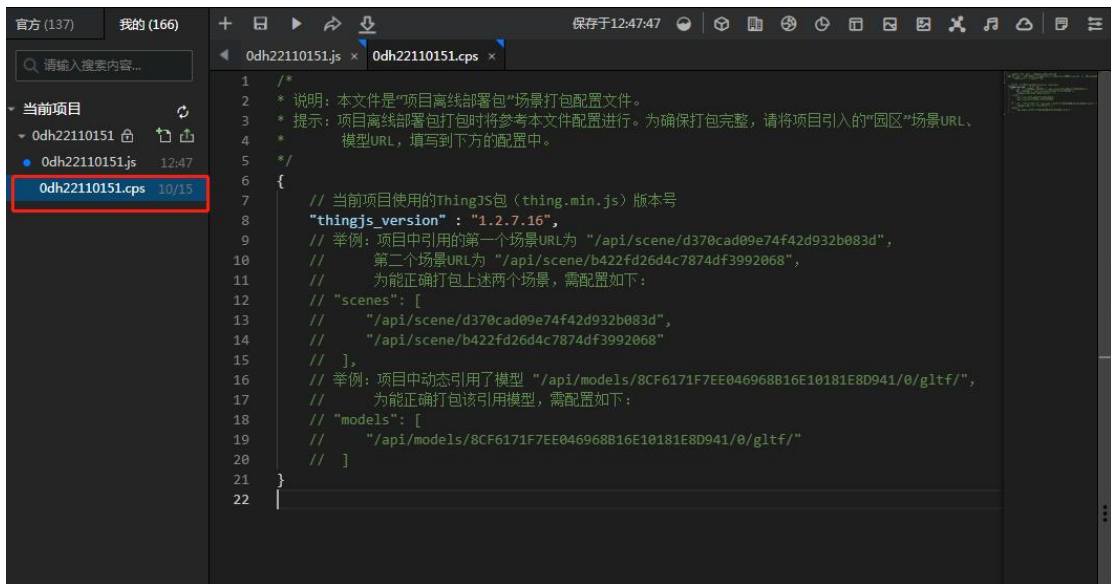
CPS 文件配置方法如下：

例如，在线开发中，需要进行离线部署的项目如下图所示：



项目中引用了一个场景 ('/api/scene/15fc31d71fba01bc0a093xxx'), 一个模型 ('/api/models/ABC7A3F0BDA48B7B7A8C9C94824359D/0/gltf/').

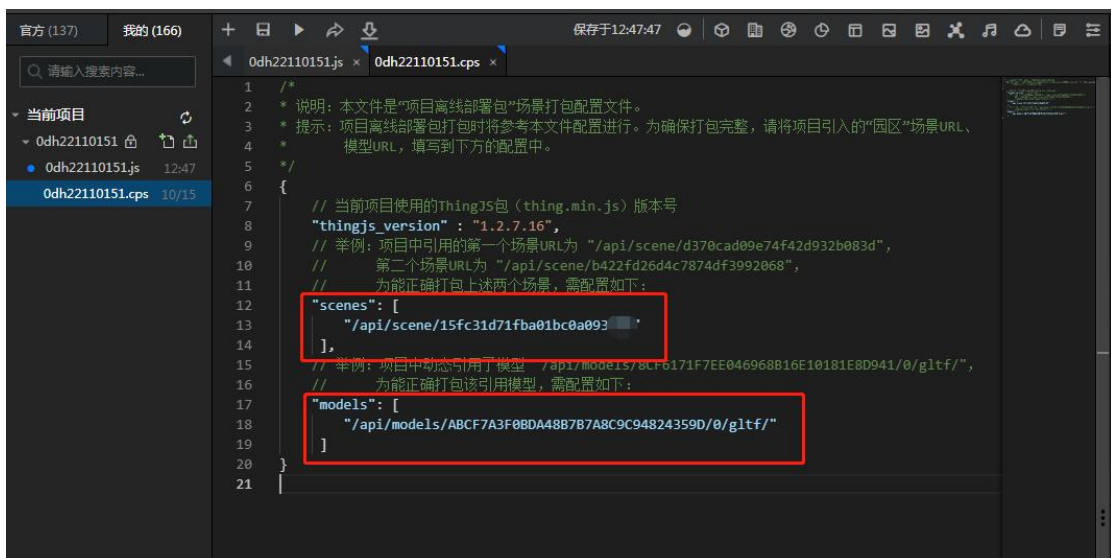
点击打开当前项目下对应的 CPS 文件，CPS 文件初始内容如下图所示：



```

1  /*
2  * 说明：本文件是“项目离线部署包”场景打包配置文件。
3  * 提示：项目离线部署包打包时将参考本文件配置进行。为确保打包完整，请将项目引入的“园区”场景URL、
4  * 模型URL，填写到下方的配置中。
5  */
6  {
7      // 当前项目使用的ThingJS包（thing.min.js）版本号
8      "thingjs_version": "1.2.7.16",
9      // 举例：项目中引用的第一个场景URL为 "/api/scene/d370cad09e74f42d932b083d",
10     // 第二个场景URL为 "/api/scene/b422fd26d4c7874df3992068",
11     // 为能正确打包上述两个场景，需配置如下：
12     // "scenes": [
13     //     "/api/scene/d370cad09e74f42d932b083d",
14     //     "/api/scene/b422fd26d4c7874df3992068"
15     // ],
16     // 举例：项目中动态引用了模型 "/api/models/8CF6171F7EE046968816E10181E8D941/0/gltf/",
17     // 为能正确打包该引用模型，需配置如下：
18     // "models": [
19     //     "/api/models/8CF6171F7EE046968816E10181E8D941/0/gltf/"
20     // ]
21 }
22
    
```

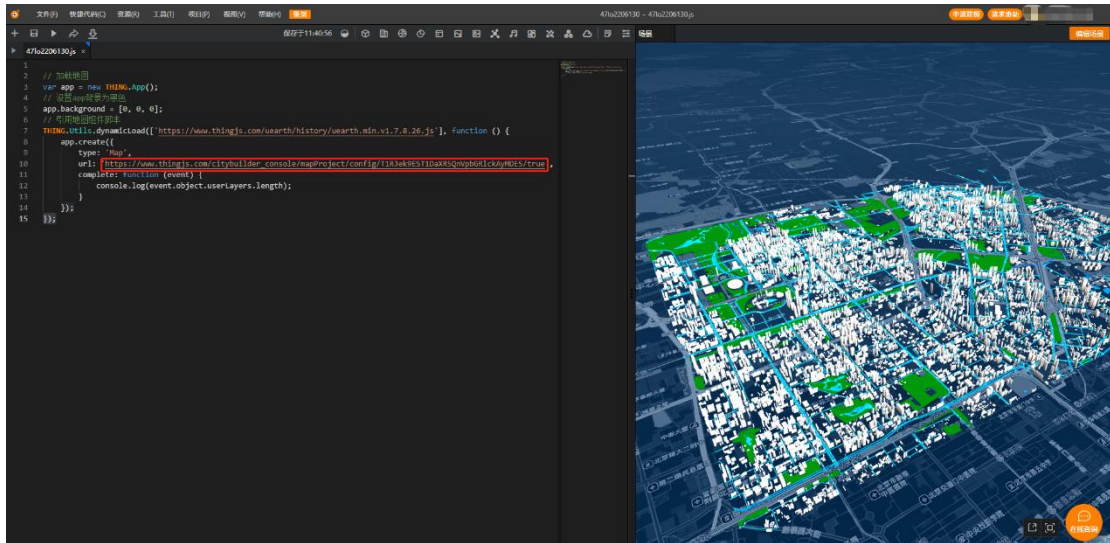
将需要离线的场景 URL 填入 scenes 配置项中，模型 URL 填入 models 配置项中，如下图所示。



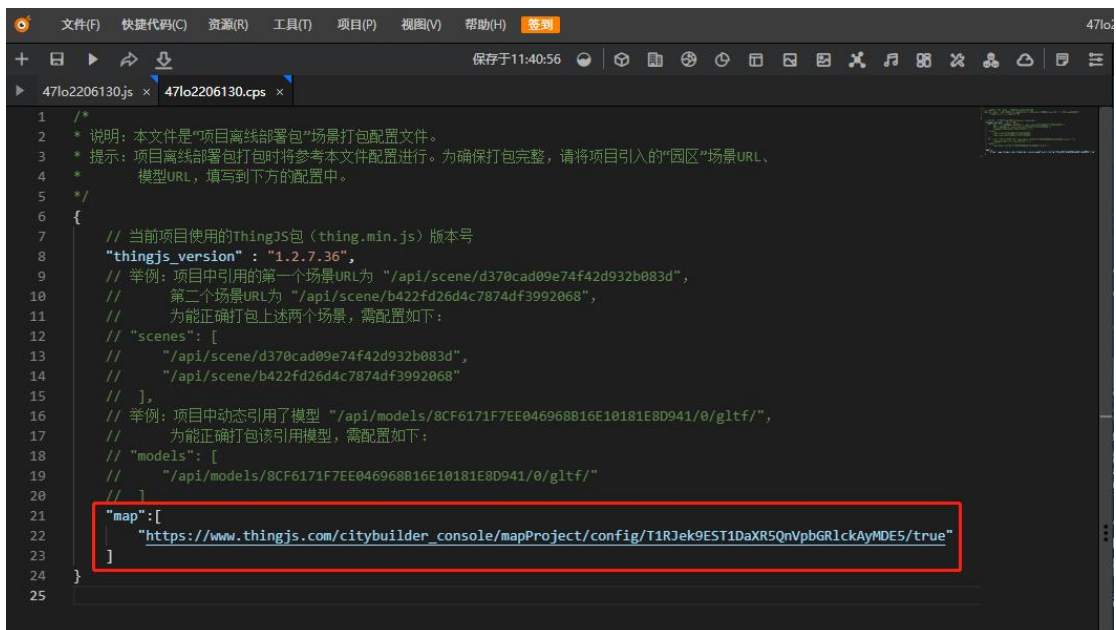
```

1  /*
2  * 说明：本文件是“项目离线部署包”场景打包配置文件。
3  * 提示：项目离线部署包打包时将参考本文件配置进行。为确保打包完整，请将项目引入的“园区”场景URL、
4  * 模型URL，填写到下方的配置中。
5  */
6  {
7      // 当前项目使用的ThingJS包（thing.min.js）版本号
8      "thingjs_version": "1.2.7.16",
9      // 举例：项目中引用的第一个场景URL为 "/api/scene/d370cad09e74f42d932b083d",
10     // 第二个场景URL为 "/api/scene/b422fd26d4c7874df3992068",
11     // 为能正确打包上述两个场景，需配置如下：
12     "scenes": [
13         "/api/scene/15fc31d71fba01bc0a093"
14     ],
15     // 举例：项目中动态引用了模型 "/api/models/8CF6171F7EE046968816E10181E8D941/0/gltf/",
16     // 为能正确打包该引用模型，需配置如下：
17     "models": [
18         "/api/models/ABC7A3F0BDA48B7B7A8C9C94824359D/0/gltf/"
19     ]
20 }
21
    
```

如项目中引入了地图场景，如下图所示：



则需要将地图场景 URL 填入 map 配置项中，如下图所示。



保存即配置完成。

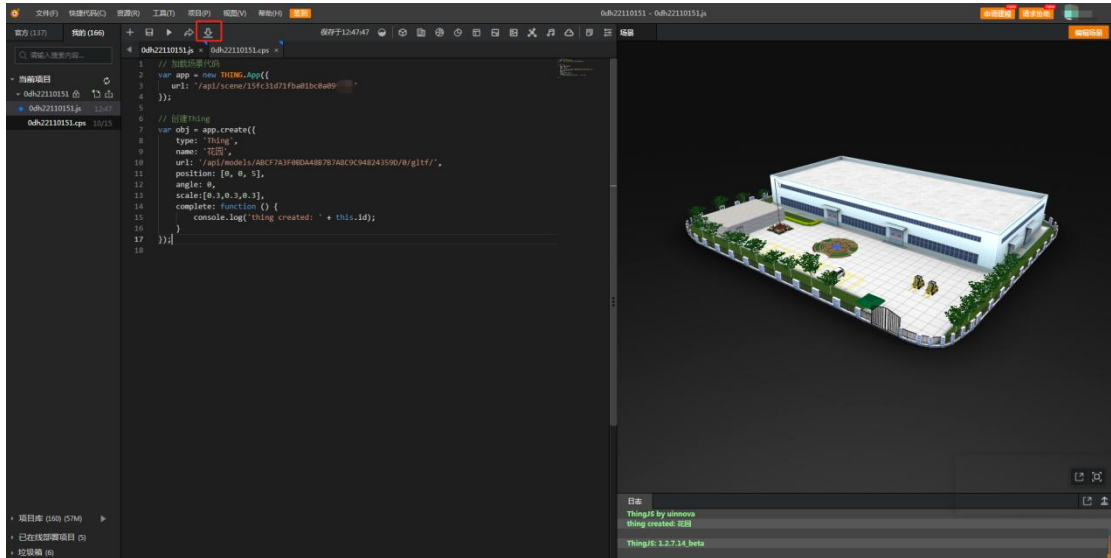
场景文件将被打包至离线部署包的 /web/scene/ 目录下，模型文件将被打包至离线部署包的 /web/models/ 目录下，地图文件将被打包至离线部署包的 /web/cityBuilder/ 目录下。

2.2 申请下载

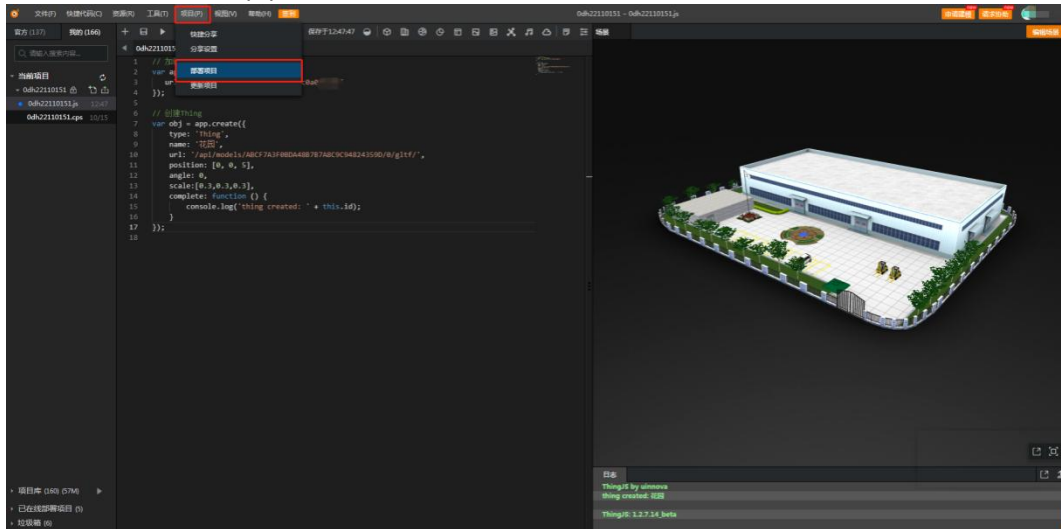
2.2.1 入口

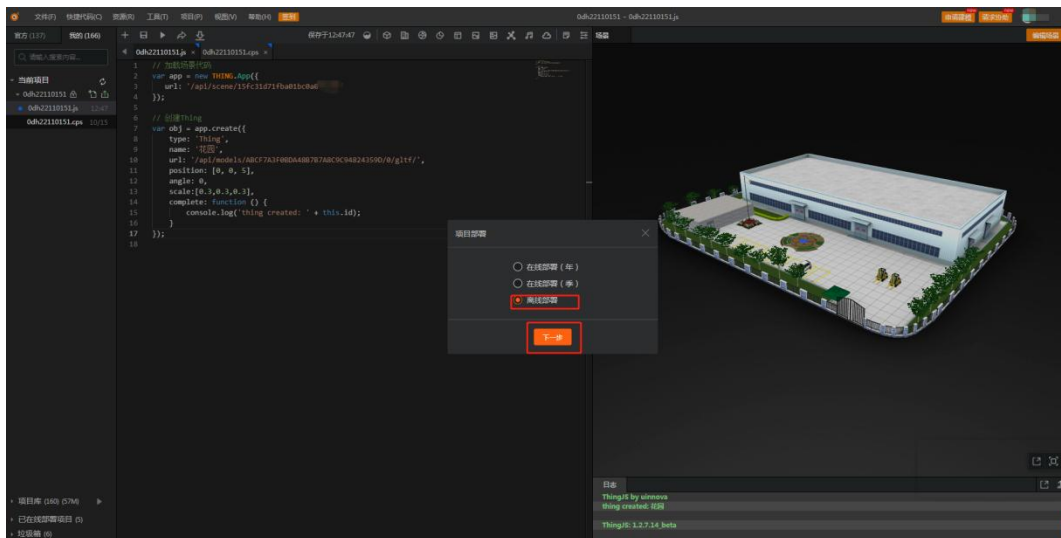
在线开发中，申请“ThingJS 项目离线部署包”需到“离线部署管理”面板中进行，“离线部署管理”面板的入口有两处：

(1) 快捷入口，如下图所示位置：



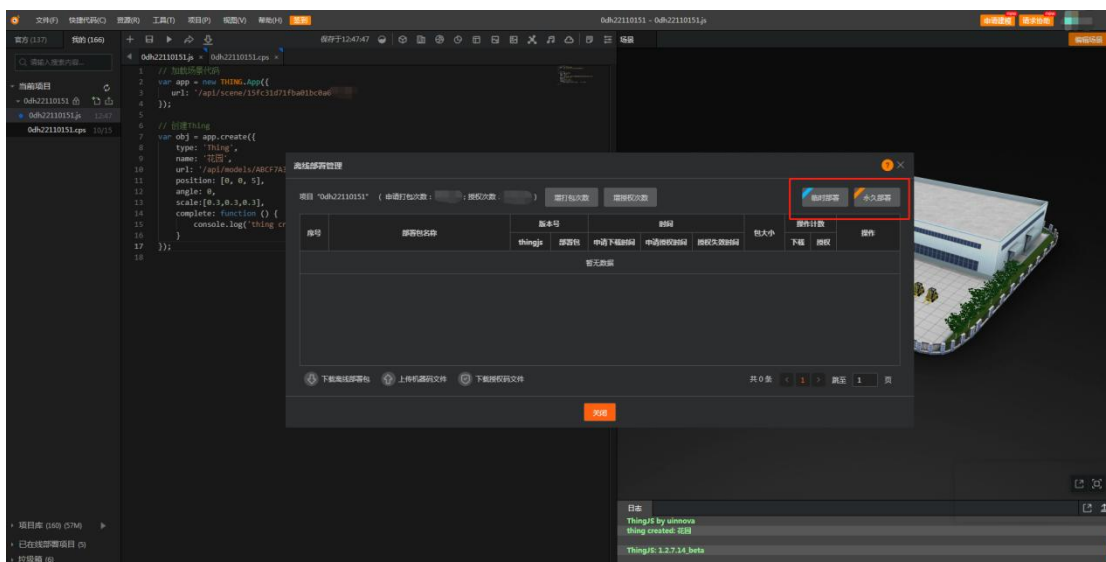
(2) 菜单—项目(P)—部署项目—离线部署





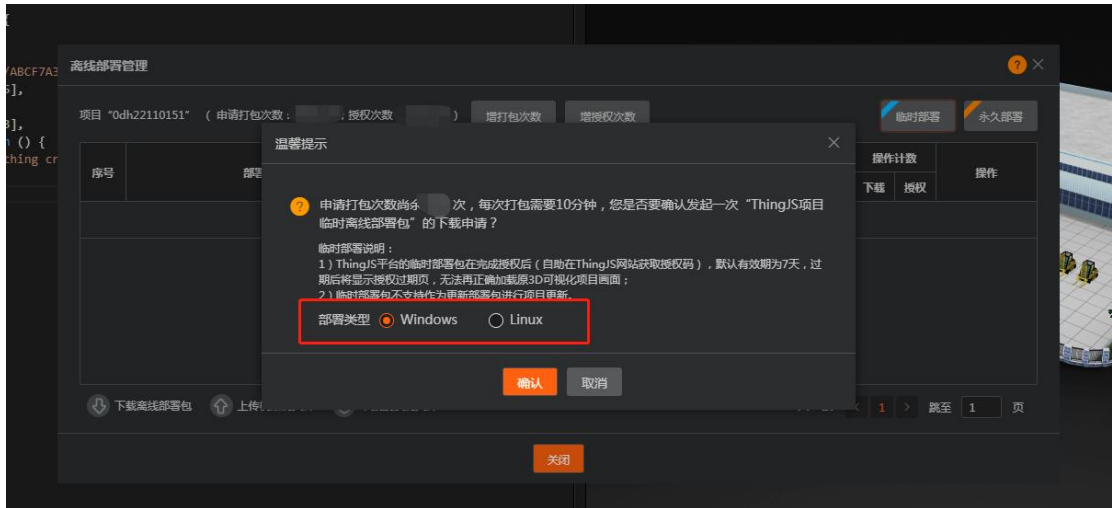
2.2.2 申请

“离线部署管理”面板如下图所示：

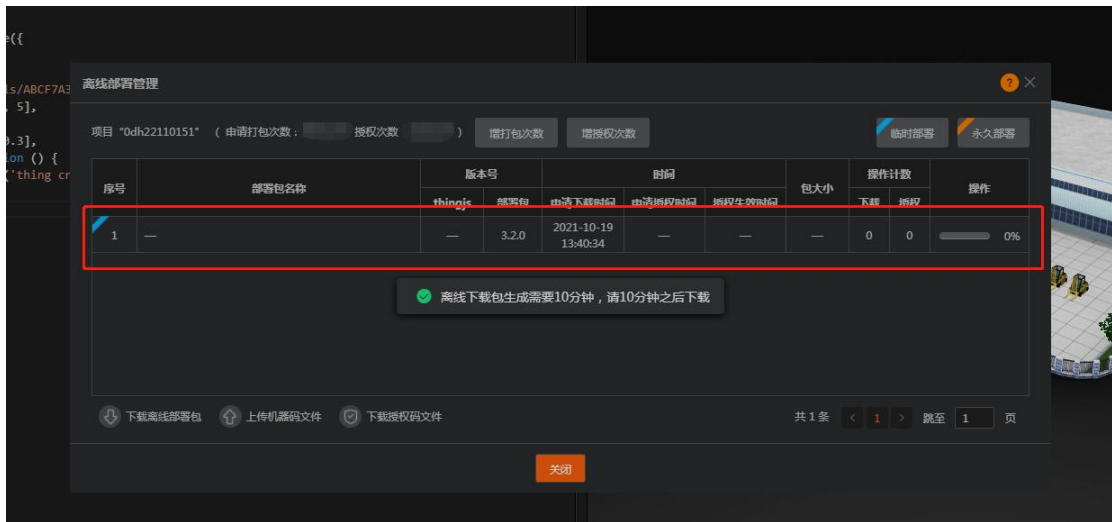


根据需求选择对应的离线部署包类型（临时部署、永久部署）进行申请。

在弹出的对话框中选择需要部署离线部署包的服务器系统（Windows、Linux）。



确认后，“离线部署管理”面板中将新增一条“离线部署包”记录，“操作”栏会显示部署包生成百分比，请耐心等待“离线部署包”生成。






2.2.3 下载

“离线部署包”生成后，即可进行下载操作，点击“下载”按钮，将“离线部署包”下载至本地。

离线部署管理

项目 "Odh22110151" (申请打包次数: ; 授权次数:) 增打包次数 增授权次数 临时部署 永久部署

序号	部署包名称	版本号		时间			包大小	操作计数		操作
		thingjs	部署包	申请下载时间	申请授权时间	授权失效时间		下载	授权	
1	1_3_0dh22110151_211019134038.zip	1.2.7.16	3.2.0	2021-10-19 13:40:34	—	—	76.0MB	0	0	  

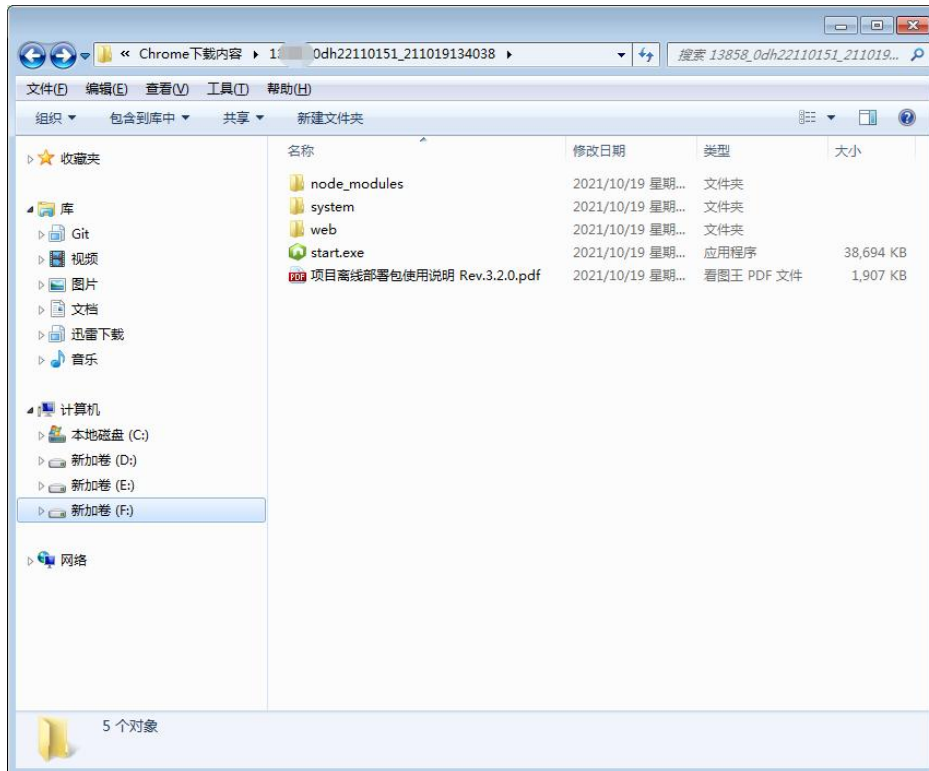
下载离线部署包 上传机器码文件 下载授权码文件
共 1 条 < 1 > 跳至 1 页

关闭

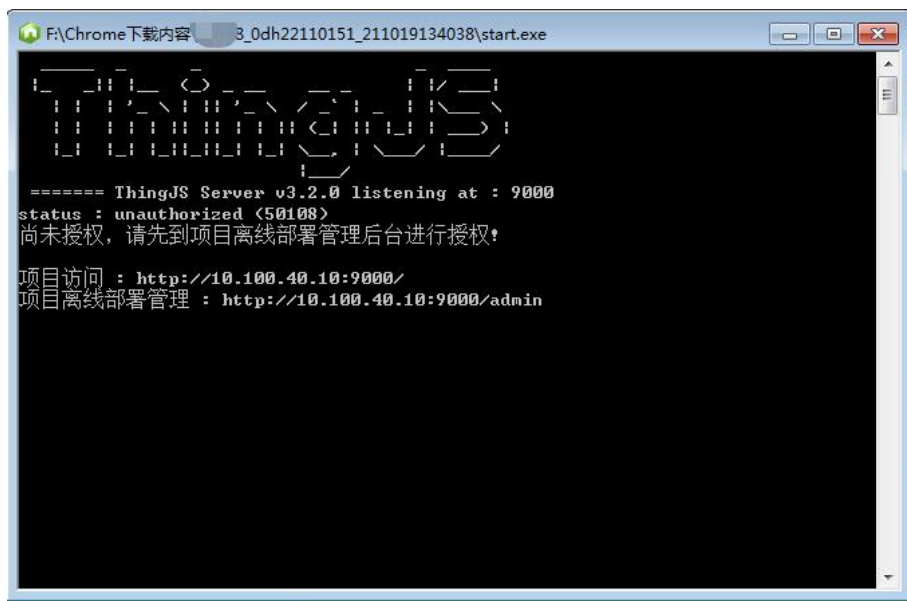
3 安装

3.1 Windows 服务器

将获取到的 Windows 版本的“ThingJS 项目离线部署包”压缩包解压至需要部署的目录下，解压后如下图所示：



双击运行 start.exe 即可启动离线部署服务，离线部署服务默认的端口号为 9000，如需修改请参考第 4.1 节的内容。服务使用期间，请勿关闭服务控制台窗口。



3.2 Linux 服务器

将获取到的 Linux 版本的“ThingJS 项目离线部署包”压缩包解压至需要部署的目录下。注意：需使用 7za 解压，命令为：7za x offline.zip -r -o./offline。

```
znu3s4ylgZ 0dh22101280]# 7za x offline.zip -r -o./offline
```

解压后文件结构如下图所示：

```
4096 1月 30 18:37 node_modules
025364 1月 30 18:41 start
4096 1月 30 13:03 system
4096 1月 30 14:18 web
427578 1月 30 13:03 离线部署包使用说明3.0.pdf
1e7-0dh22101280]#
```

通过命令 `chmod u+x start` 为 `start` 文件增加执行权限。

```
gZ 0dh22101280]# chmod u+x start
gZ 0dh22101280]# ll

4096 1月 30 18:37 node_modules
5364 1月 30 18:41 start
4096 1月 30 13:03 system
4096 1月 30 14:18 web
7578 1月 30 13:03 离线部署包使用说明3.0.pdf
```

执行 `start` 启动服务，离线部署服务默认的端口号为 9000，如需修改请参考第 4.1 节的内容。

```
29104400]# ./start

===== ThingJS Server v3.2.0 listening at 0.0.0.0 9000
status : unauthorized (50108)
尚未授权，请先到项目离线部署管理后台进行授权！

项目访问 : http://172.22.22:9000/
项目离线部署管理 : http://172.22.22:9000/admin
```

为保证进程能够保持在后台运行，可在 `screen`（多重视窗管理程序）中执行，或使用其他后台执行的方法。

可通过命令 `netstat -naptl | less` 或 `ps axu | grep start | less` 查看端口情况。

```
21858 2.2 1.6 945572 34544 pts/0 Sl 17:58 0:00 ./start
```


若要关闭离线部署服务，可通过 kill -9 PID 命令结束相应进程。

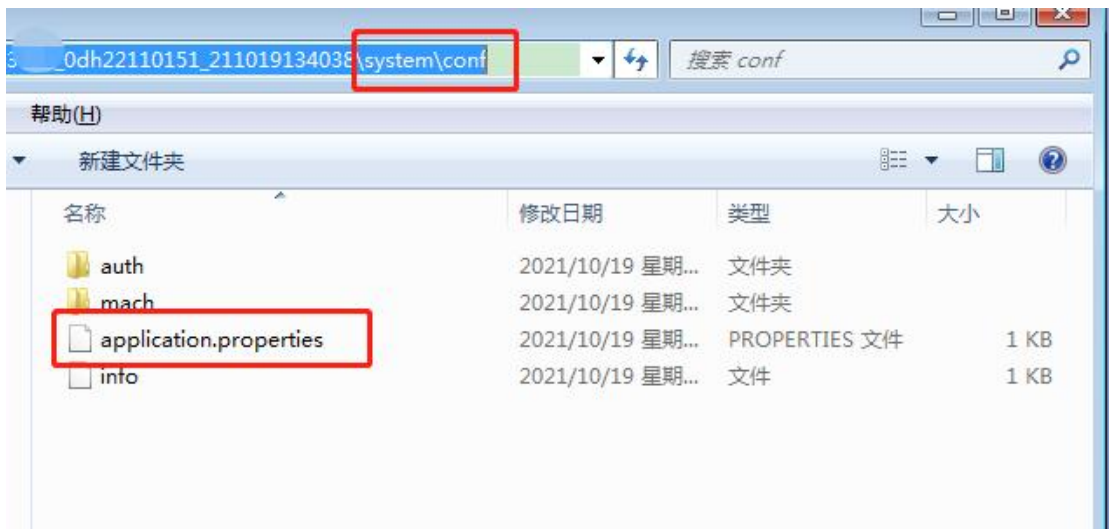
```
h22101280]# kill -9 21858
```

4 配置

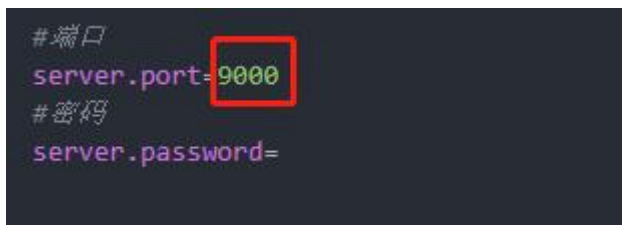
本章介绍了几种常见的“ThingJS 项目离线部署包”支持的配置方法，请勿自行更改未说明的离线部署包中其他的文件内容，以免发生不可预期的部署问题，导致部署失败。

4.1 修改服务启动端口

“ThingJS 项目离线部署服务”默认端口号为 9000，可以通过配置离线部署包中 system\conf 目录下的 application.properties 文件来修改端口号。



用文本编辑器打开 application.properties，修改 server.port 为新端口号。



注意不要和原来的服务端口冲突，若端口冲突会无法启动离线部署服务，如下图所示：



4.2 配置 HTTPS 访问（可选）

（注：若服务不需要开启 HTTPS 访问，可跳过该节。）

通过配置 Nginx，开启 HTTPS 服务，并代理到离线部署服务，配置如下图所示：

```
server {
    listen      443 ssl http2 default_server;
    listen     [::]:443 ssl http2 default_server;
    server_name _;
    root       /r[redacted];

    ssl_certificate "/[redacted].pem";
    ssl_certificate_key "/[redacted].key";
    ssl_session_timeout 5m;
    ssl_session_cache shared:SSL:1m;
    ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_prefer_server_ciphers on;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    location / {
        add_header Cache-Control no-cache;
        proxy_pass http://127.0.0.1:9000;
        client_max_body_size 500M;
    }
}
```

4.3 修改登录密码

为确保离线部署项目的安全，请在首次使用默认密码登录后尽快修改登录密码。

成功登录“项目离线部署管理”界面后，如图所示，点击“修改密码”，



在弹出的对话框中输入需要修改的新密码，即可完成登录密码的修改。



密码将以“密文”的形式保存在离线部署包 `system\conf` 目录下的 `application.properties` 文件中，如忘记密码，可自行“删除”配置文件中的密码设置项，即可自动恢复默认密码（admin），如下图所示。

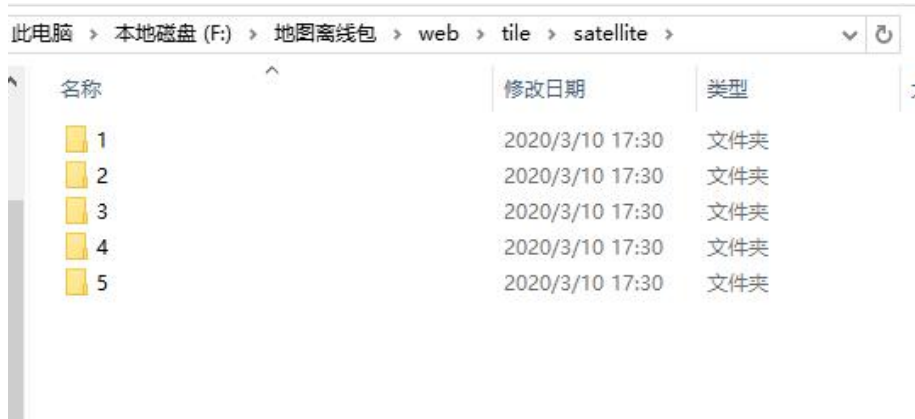
（注：请勿擅自填写密码设置项的内容。）



4.4 瓦片地址服务配置

对于离线的地图项目，因为地图瓦片资源数据量较大，离线部署包默认提供前五级离线瓦片内容，存放在“`web\tile\satellite`”目录下，如需更多层级的离线瓦片内容需要开发人员自己来做。当前支持加载 ZXY 格式的地图服务，如果有

瓦片地图的散图(例如使用工具下载的地图瓦片), 可以直接将散图目录整体拷贝到 web\tile\satellite 下, 替换旧的目录。



如果已有自己的地图服务, 可在创建地图的时候设置, 方法如下:

- (1) 标准版及专业版地图, 可打开/web/cityBuilder 中对应地图场景目录下的 map.json 文件, 找到配置瓦片图层的地方:

```
]{
  "data": {
    "cameraFlyTo": {
      "heading": 0,
      "angle": 0,
      "pitch": 45,
      "time": 2000,
      "lonlat": [116.36727, 39.90951, 0],
      "height": 754.30459
    },
    "tileLayer": [{
      "name": "Google影像",
      "style": {
        "template": "normal",
        "brightness": 1
      },
      "type": "TileLayer",
      "url": "/tile/satellite/{z}/{x}/{y}.jpg"
    }
  ],
}
```

将红框中的地址替换为要使用的地图服务地址。

- (2) 森城市地图, 打开/web/cityBuilder 中对应地图场景目录下的 source.cfg 文件, 找到配置瓦片图层的地方:

```

source.cfg.txt
1 {
2   "title": {
3     "url": "https://webst01.is.autonavi.com/appmaptile?lang=zh_cn&size=1&scale=1&style=6&x={x}&y={y}&z={z}"
4   },
5   "cameraFlyTo": {
6     "heading": 4.970851810855358,
7     "angle": 0,
8     "pitch": 89.42704220487036,
9     "time": 0,
10    "lonlat": [
11      107.66280194520374,
12      32.61818636997671,
13      2.0489096641540528e-8
14    ],
15    "height": 19999758.20850401
16  },
17  "vectorEarthLayers": [
18    {
19      "name": "全球国家线",
20      "id": 3329772,
21      "url": "/geojson/2156143be15b4d4d90a489b2157fa28d.geojson"
22    },
23    {
24      "name": "全球国家线",
25      "id": 3329773,
26      "url": "/geojson/705222aeede74e48b0a4a8bcb07338e9.geojson"
27    },
28    {
29      "name": "中国省级行政区线",
30      "id": 3329770,
31      "url": "/geojson/ceb6df5a0b4c47eeba50e6e2felcd96c.geojson"
32    },
33    {
34      "name": "中国国界线"
35    }
36  ]
37 }
    
```

将红框中的地址替换为要使用的地图服务地址，或修改为“/tile/satellite/{z}/{x}/{y}.jpg”访问离线部署包中的离线瓦片内容。

```

source.cfg.txt
1 {
2   "title": {
3     "url": "/tile/satellite/{z}/{x}/{y}.jpg"
4   },
5   "cameraFlyTo": {
6     "heading": 4.970851810855358,
7     "angle": 0,
8     "pitch": 89.42704220487036,
9     "time": 0,
10    "lonlat": [
11      107.66280194520374,
12      32.61818636997671,
13      2.0489096641540528e-8
14    ],
15    "height": 19999758.20850401
16  },
17  "vectorEarthLayers": [
18    {
19      "name": "全球国家线",
20      "id": 3329772,
21      "url": "/geojson/2156143be15b4d4d90a489b2157fa28d.geojson"
22    },
23    {
24      "name": "全球国家线",
25      "id": 3329773.
    }
26  ]
27 }
    
```

打开加载地图代码块的文件，添加如下配置：

```
var app = new THING.App();
app.background = [0, 0, 0];
THING.Utils.dynamicLoad(['./source/uearth.min.js'], function () {
  app.create({
    type: 'Map',
    url: "./cityBuilder/1/map.bundle.json",
    resourceConfig: {
      resourcePrefix: './',
      loadDataFirst: true,
      isProxima: true,
      changeCampusLevel: false,
      externalConfigUrl: './cityBuilder/1/source.cfg'
    },
    complete: function (event) {
      console.log(event.object.userLayers.length);
    }
  });
});
```

注：以下代码块可直接复制使用。

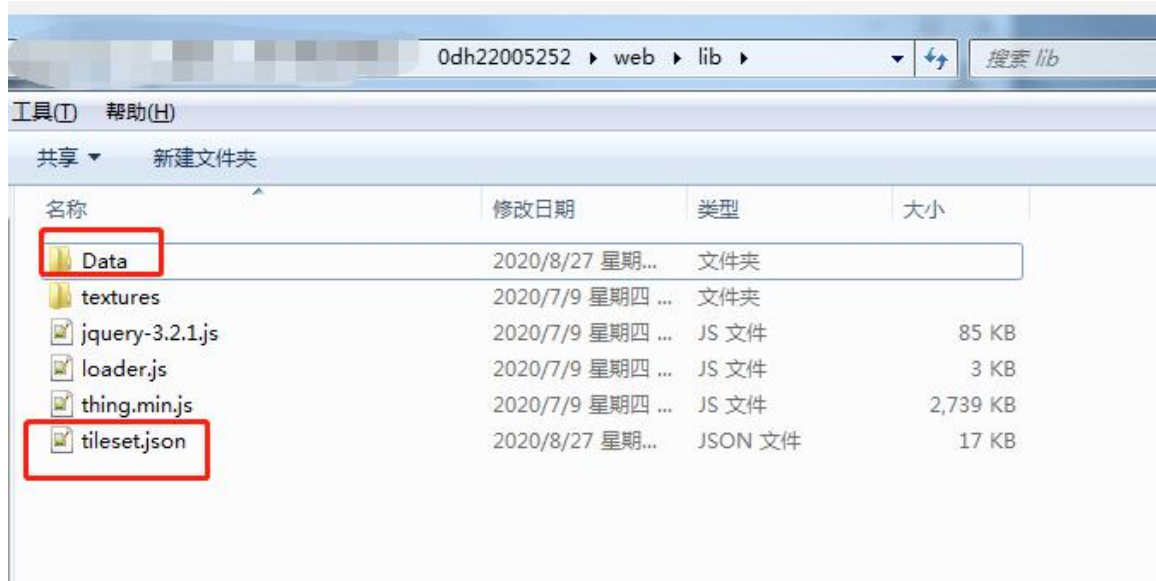
```
resourceConfig: {
  resourcePrefix: './',
  loadDataFirst: true,
  isProxima: true,
  changeCampusLevel: false,
  externalConfigUrl: './cityBuilder/1/source.cfg'
}
```

4.5 倾斜摄影数据配置

ThingJS 3D 可视化平台支持倾斜摄影数据，若需要在项目中使用倾斜摄影，平台提供倾斜摄影 osgb 数据转换处理上图服务，可咨询平台客服。

在项目离线部署时，请将 ThingJS 平台为您转换处理后的倾斜摄影数据复制到部署目录的\web\lib 文件夹下，需保证路径中有正确的 tileset.json 文件以及相对

的倾斜摄影文件夹。如下图所示：



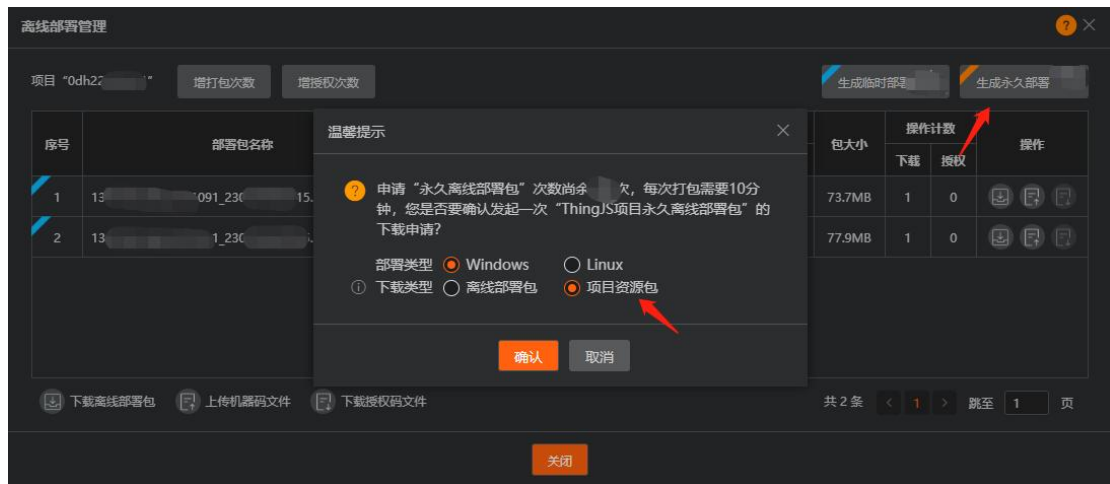
然后需要手动修改离线部署包中的主脚本，将 3dtiles 的服务地址更换为离线部署包相对地址 “./lib/tileset.json”。如下图所示：

```
//添加一个倾斜摄影图层,url是一个3dtiles服务地址
var tile3dLayer = app.create({
  type: 'Tile3dLayer',
  name: 'tileset',
  position: position,
  angles: angles,
  url: './lib/tileset.json'
});
```

4.6 单服务器多项目部署配置

在默认情况下，仅可以下载单服务器单项目部署包，如有单服务器部署多项目的需求，需要购买“单服务器多项目部署授权”。

购买“单服务器多项目部署授权”后，将拥有下载多项目对应的“项目资源包”的功能，该“项目资源包”含有项目资源文件及对应的更新程序。

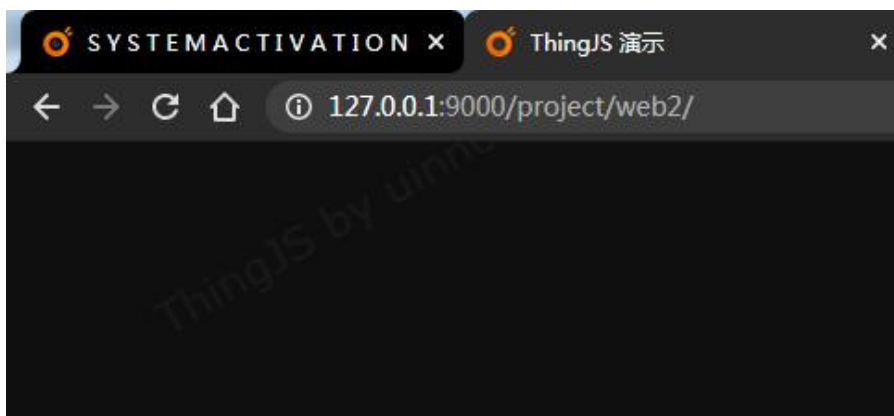


将得到的“项目资源包”解压至已离线部署的根目录下，如下图所示。



Windows 环境下可以直接执行更新程序 update.exe，Linux 环境下执行 ./update。执行前确保原部署服务已停止，否则无法正确完成更新。

已部署的离线部署包将升级为“单服务器多项目部署包”，重新启动离线部署服务，向官方申请对应项目数的“多项目授权码”，授权通过后，即可通过“/project/目录名/”的方式访问对应的项目。



如果是离线开发产品需要进行单服务器多项目部署的话，需要下载需要部署的第二个项目的离线部署包，将第二个项目的离线部署包资源中的 web 目录拷贝到离线部署服务区对应的目录下，命名为 web1，如果有更多项目的话，需要手动拷贝每个项目的 web 目录资源，在对应的服务器目录下逐步命名为 web1, web2...;然后重新启动离线部署服务，向官方申请对应项目数的“多项目授权码”，授权通过后，即可通过“/project/目录名/”的方式访问对应的项目。

离线开发产品需要进行单服务器多项目部署更新流程：原 web 目录下的更新是走上传更新操作；其他项目目录下的是替换对应的项目目录

4.7 服务请求配置

离线部署服务请求 Response Headers 中 Access-Control-Allow-Origin 默认不允许跨域，可以通过配置离线部署包中 system\conf 目录下的 application.properties 文件来修改。如下图所示，修改 server.origin 为允许请求的 origin 白名单。

```
#端口
server.port=9000
#密码
server.password=
#允许跨域
server.origin=127.0.0.1,localhost,www.thingjs.com
#允许访问管理页面
server.manager=127.0.0.1
#允许使用gzip压缩
server.gzip=js,geojson
```

注：配置完成以后需要重新启动服务。

4.8 管理页面访问白名单配置

如需要设置仅某些地址可访问离线部署管理页面，可以通过修改离线部署包中 system\conf 目录下的 application.properties 配置文件实现。如下图所示，修改 server.manager 为允许访问离线部署管理页面的白名单。

```
#端口
server.port=9000
#密码
server.password=
#允许跨域
server.origin=127.0.0.1,localhost,www.thingjs.com
#允许访问管理页面
server.manager=127.0.0.1
#允许使用gzip压缩
server.gzip=js,geojson
```

注：配置完成以后需要重新启动离线部署服务。

4.9 资源请求压缩配置

为减少文件的网络传输时间，可以通过修改离线部署包中 system\conf 目录下的 application.properties 配置文件实现。如下图所示，修改 server.gzip 为需要进行压缩的文件后缀，如：js、geojson、bin、json 等。

```
#端口
server.port=9000
#密码
server.password=
#允许跨域
server.origin=127.0.0.1,localhost,www.thingjs.com
#允许访问管理页面
server.manager=127.0.0.1
#允许使用gzip压缩
server.gzip=js,geojson
```

注：配置完成以后需要重新启动离线部署服务。gzip 压缩对除 “/web/lib/”、 “/web/source/”、 “/web/dist/” 及 “/web/static/” 文件夹下的文件生效。

5 授权

5.1 访问“项目离线部署管理”界面

根据服务启动信息，访问“项目离线部署管理”界面（<http://ip:port/admin>）。

```
29104400]# ./start

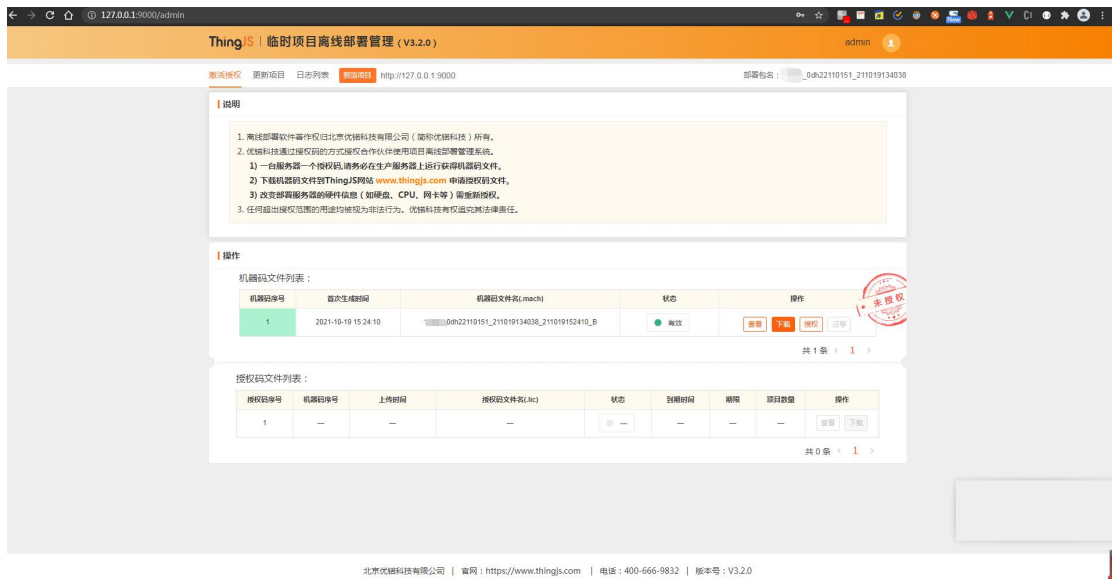
===== ThingJS Server v3.2.0 listening at 0.0.0.0 9000
status : unauthorized (50108)
尚未授权，请先到项目离线部署管理后台进行授权！

项目访问 : http://172.22.9000/
项目离线部署管理 : http://172.22.9000/admin
```

访问“项目离线部署管理”界面，需要进行登录操作，用户名和密码初始默认都为“admin”（为确保离线部署项目的安全，请在首次登录后尽快修改密码，参考第 4.3 节内容），登录界面如下图所示：



登录成功后即可进入到“项目离线部署管理”界面，如下图所示：



5.2 授权“临时部署包”

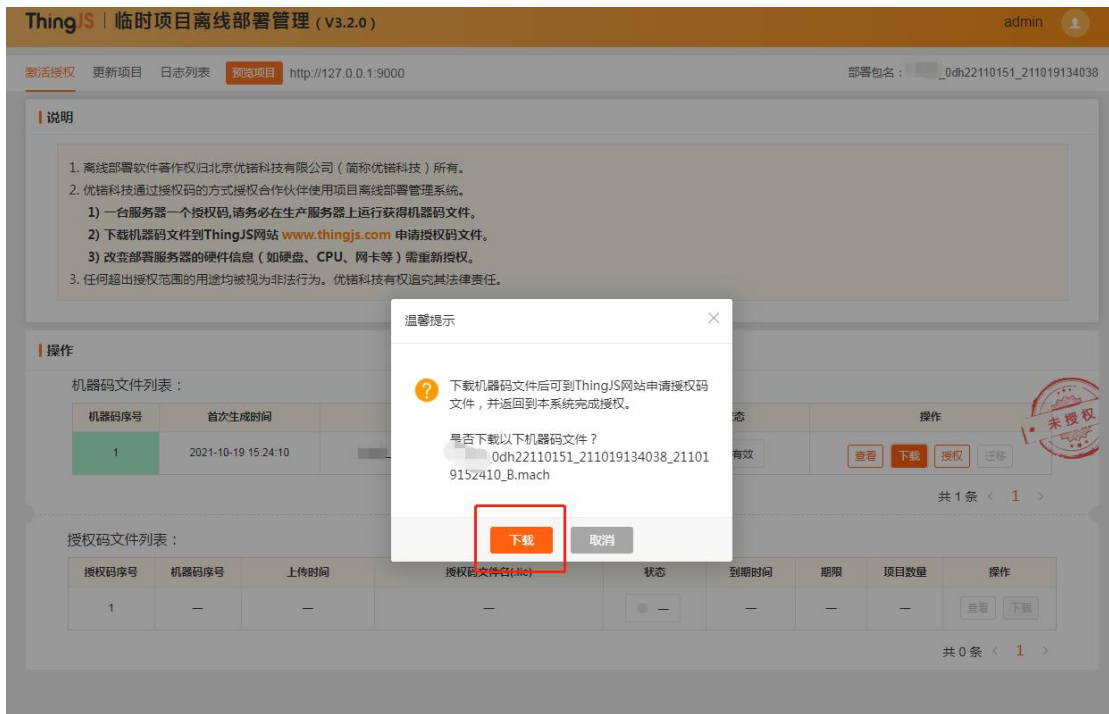
ThingJS 3D 可视化开发平台当前有效的“VIP 商业开发者”，可在“ThingJS 在线开发平台” (<https://www.thingjs.com/guide/?m=sample>) 获取 7 天临时授权码文件对“临时部署包”进行授权。请按以下章节描述的步骤逐步进行操作。

5.2.1 下载机器码文件

首先在机器码文件列表中，下载当前有效的机器码文件：



将该机器码文件下载至本地：

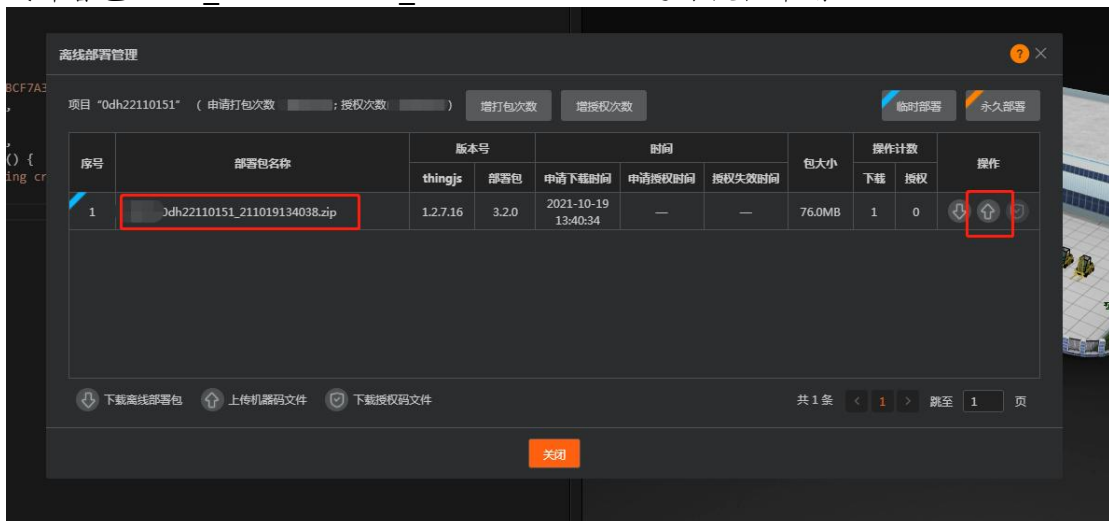


5.2.2 申请临时授权码文件

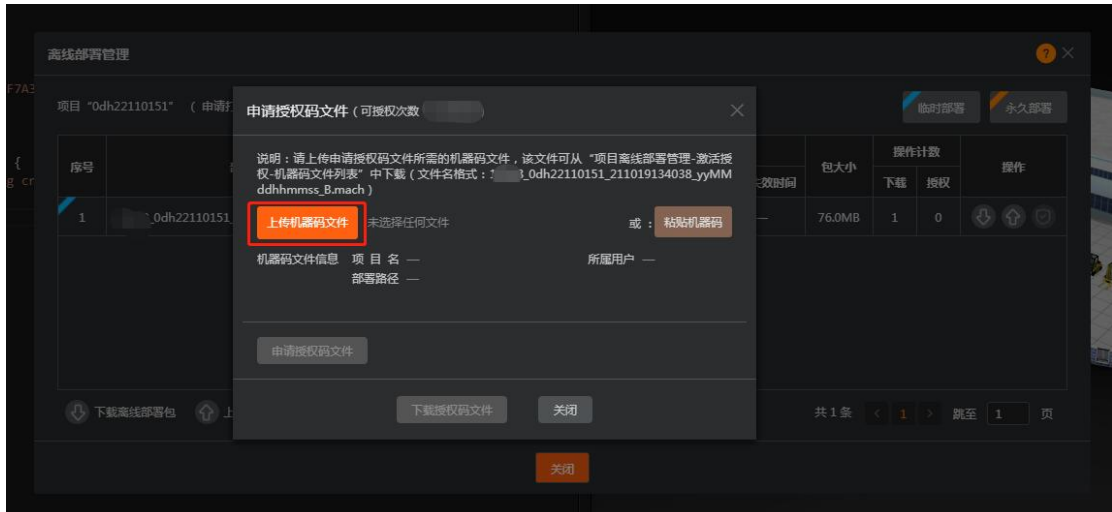
访问“ThingJS 在线开发平台”，打开部署项目（如本例所示，部署项目为“Odh22110151”）的“离线部署管理”面板，找到机器码文件对应的离线部署包的记录，点击“上传机器码文件”按钮。

如本例所示，机器码文件为

“xxx_Odh22110151_211019134038_211019152410_B.mach”，需找到对应的离线部署包“xxx_Odh22110151_211019134038”进行授权申请：



在弹出的对话框中上传对应的机器码文件（3.2.0 版本以下的离线部署包用户，可通过粘贴机器码的方式获取）。



成功上传正确的机器码文件后，即可申请获取对应的授权码文件（本例中对应的授权码文件为“xxx_Odh22110151_211019134038_xxxxxxxx_B.lic”），下载授权码文件到本地。如图所示：



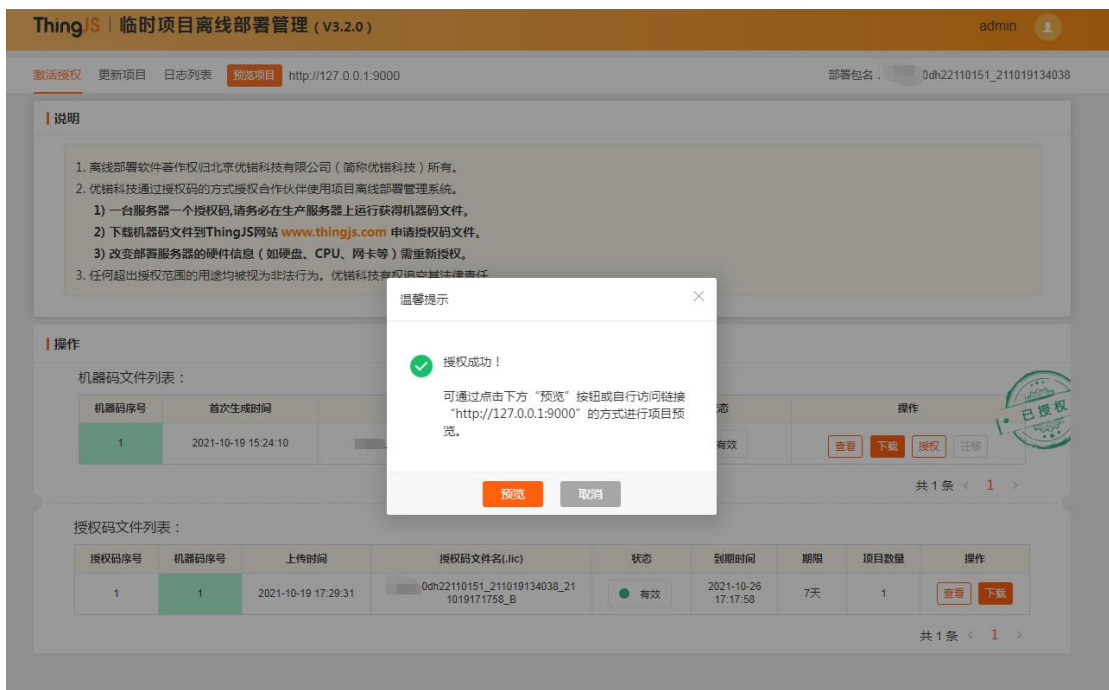
5.2.3 完成授权

返回到“项目离线部署管理”界面，在当前有效的机器码文件一栏中选择“授权”按钮，上传所获取的对应的授权码文件。



如本例所示，为机器码文件

“xxx_0dh22110151_211019134038_211019152410_B.mach”，上传对应的授权码文件“xxx_0dh22110151_211019134038_xxxxxxx_B.lic”，即可完成授权。



授权成功后，可通过点击“预览”按钮直接进行项目预览，或通过访问根目录链接进行项目预览。

需要注意的是，“临时部署包”通过“临时授权码”授权成功后，有效期为 7 天。如需永久授权，请更新升级为“永久部署包”（参考第 6 章），或直接使用“永久部署包”进行授权。

5.3 授权“永久部署包”

“永久部署包”必须使用“永久授权码”进行授权激活。

同 5.2.1 所述，需先获得离线部署包当前有效的“机器码文件”，然后将该“机器码文件”发送给 ThingJS 平台商务人员，获取离线部署包对应的“永久授权码文件”，最后返回到“项目离线部署管理”界面，同 5.2.3 所述完成“永久部署包”的授权。

6 更新

为方便开发人员对已部署的“ThingJS 项目离线部署包”进行项目内容的更新，“ThingJS 项目离线部署包”提供“更新项目”的功能。

6.1 获取“更新部署包”

首先，同第 2 章所述，获取一个当前已部署项目对应的需要进行更新的“离线部署包”（只可更新当前已部署项目的离线部署包）作为“更新部署包”，需要注意的是，只有“永久部署包”才可以作为“更新部署包”进行项目更新。

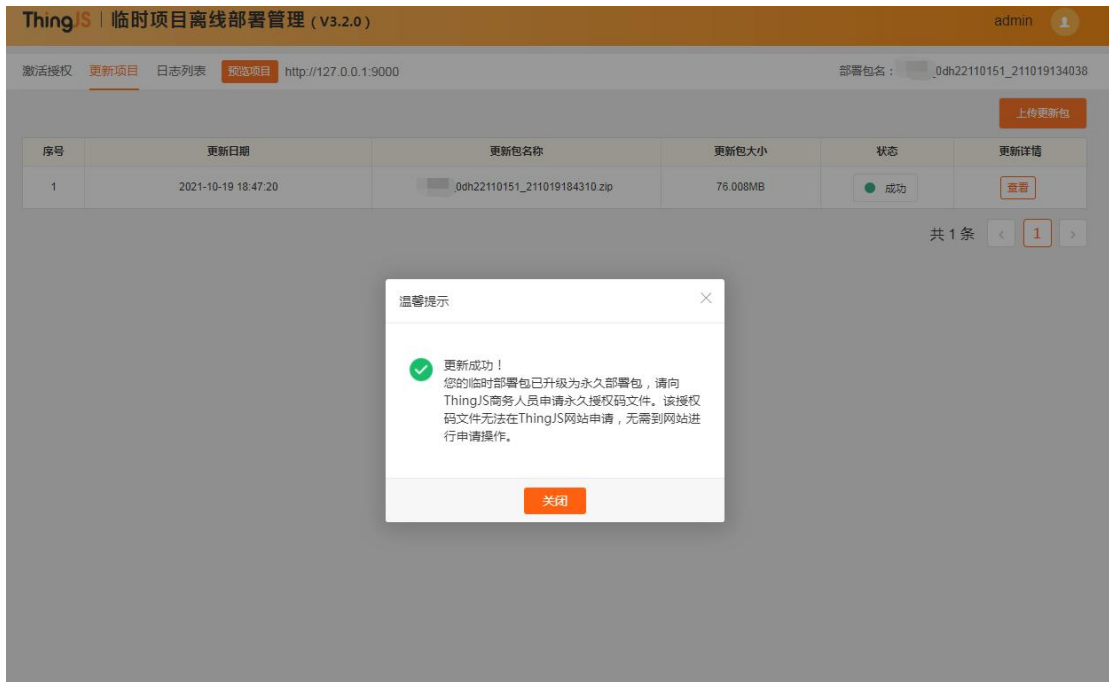
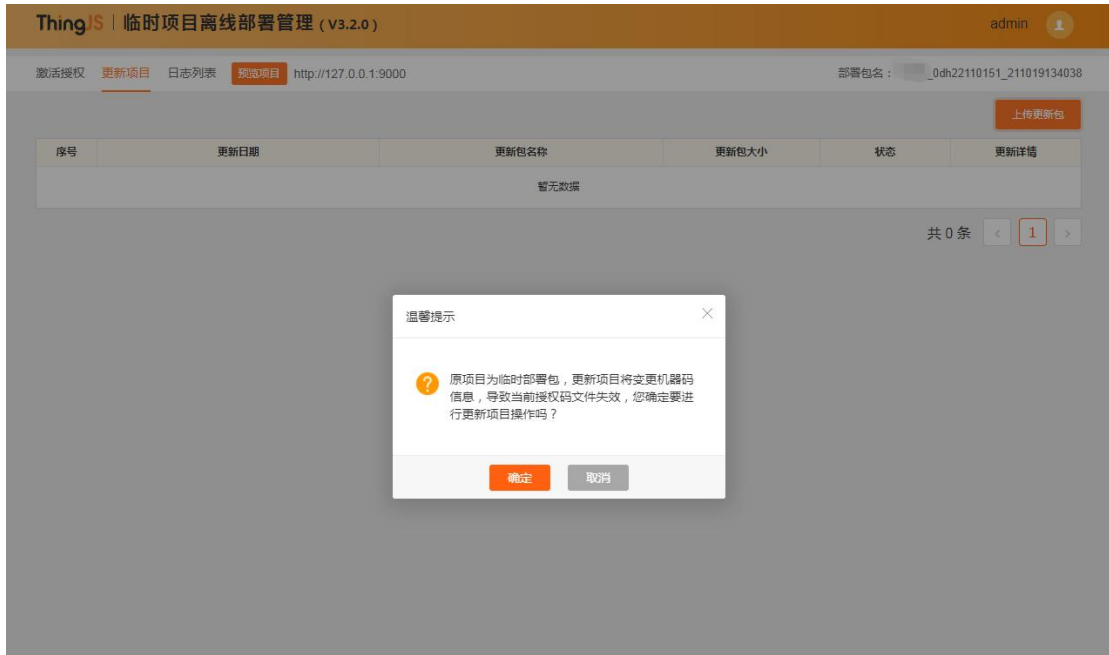
6.2 更新项目

在“项目离线部署管理”界面中，切换页签到“更新项目”，点击“上传更新包”按钮，上传获取到的“更新部署包”。



如果您当前已部署的“离线部署包”为“临时部署包”，“更新项目”成功后，当前“临时部署包”自动更新升级为“永久部署包”，当前的临时授权将失效，升级为“永久部署包”后，需对该“永久部署包”进行“永久授权”（参考第 5.3 节内容）。

如果您当前已部署的“离线部署包”为“永久部署包”，“更新项目”将仅更新项目内容，并不会影响已有授权。



更新成功,并正确授权后,即可预览更新项目内容后的离线部署项目。

7 升级

为了不断提升我们的产品质量和用户体验，我们对离线部署产品进行改进和优化，并推出了更加完善的升级版本，希望给您带来更加出色的使用体验。下面是升级工作的一些步骤说明和注意事项，请您认真阅读。

（该升级仅为部署服务升级，不涉及用户业务系统）

注：请根据实际情况自行判断适当的升级方法，并进行相应操作：

***当前版本在 v3.2.0 以上用户（不含 v3.2.0）：**

如您已部署 ThingJS 离线部署服务，且版本在 v3.2.0 以上，可按手册说明到官网下载更新包，自行完成升级即可。

***当前版本为 v3.2.0 的用户：**

如您当前部署的版本为 v3.2.0 版本，需在安装更新包的基础上，按手册说明进行后续操作步骤。

***当前版本为 v3.2.0 以下的用户：**

推荐您参照第 8 章通过迁移操作，升级您的离线部署服务器版本。

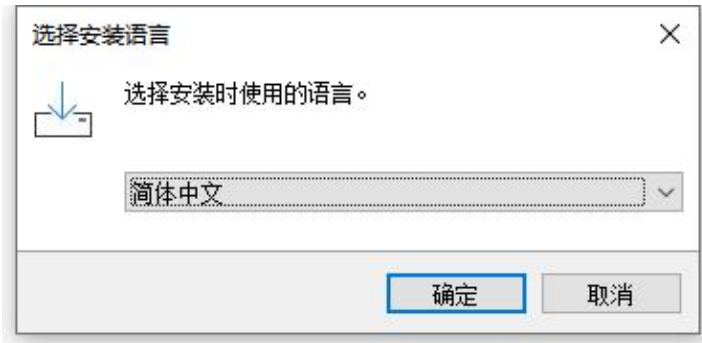
7.1 Windows 服务器

打开低代码在线开发-平台产品页面（<https://www.thingjs.com/guide/offlineDeployment/>），下载项目离线部署更新包（Windows 版），如下图所示：





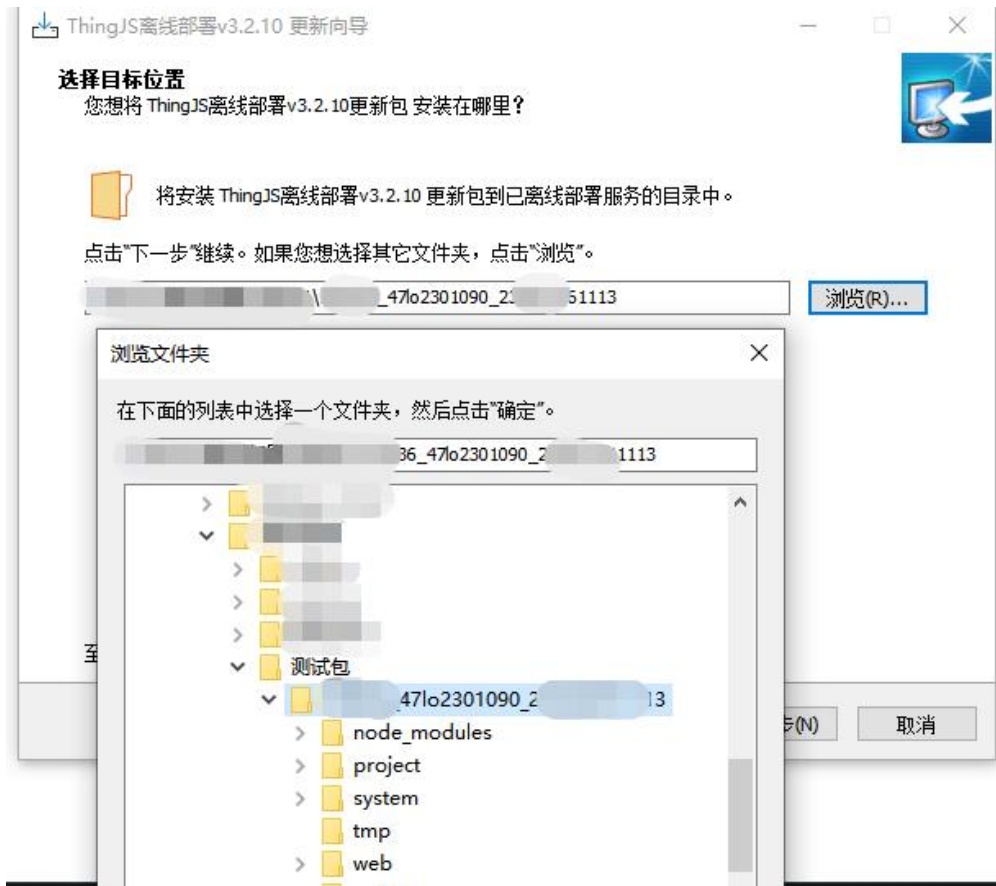
解压下载的“ThingJS 离线部署 v x.x.x 更新包 (Windows) .zip”更新包，并双击进入更新页面，默认安装语言为中文，如下图所示：



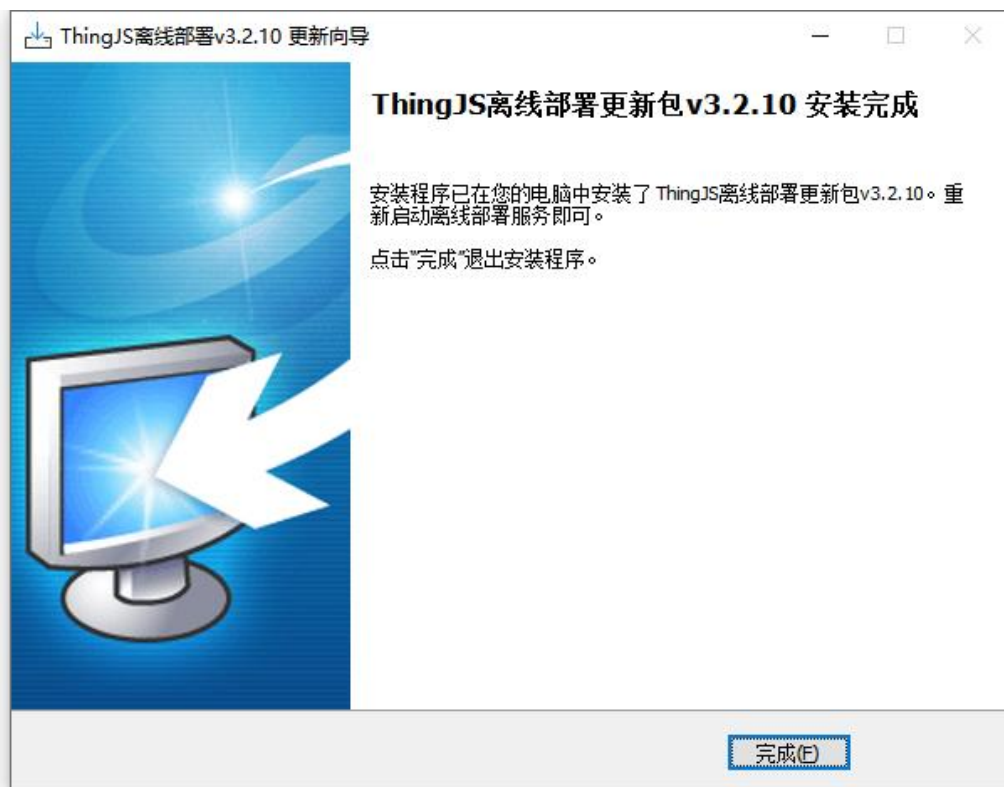
点击“确定”后，进入安装界面，按照界面操作提示进入下一步，如下图所示：



进入下一步后，提示选择更新包安装位置，需要手动点击“浏览”，选择已离线部署服务的目录，如下图所示：

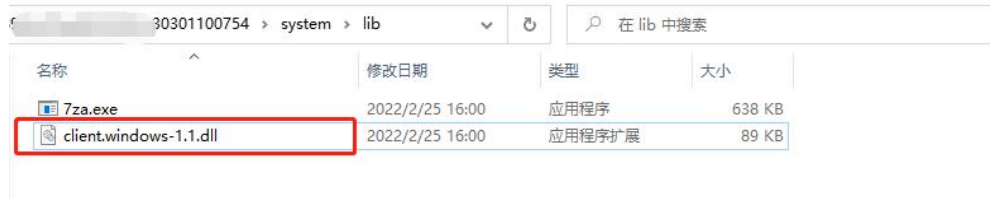


选择完安装目录后，点击“确定”进入安装步骤，直至提示更新完成，如下图所示。

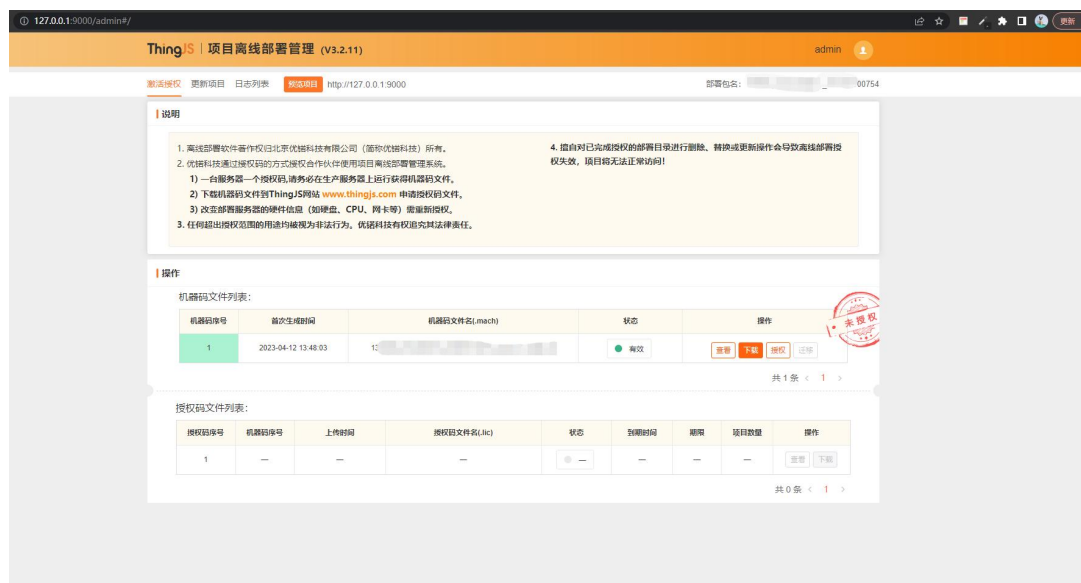


至此，已完成更新 v3.2.0 以上的部署服务；若您的早期部署版本低于（包含）v3.2.0 版本，请继续按以下步骤操作：

联系 ThingJS 商务人员获取 client.windows-1.1.dll 文件，并放至部署目录 system/lib 下；



启动离线部署服务，访问授权界面，将新机器码文件发送至对应商务人员，重新给予授权；



授权成功后，本次部署服务升级将全部完成，您可点击预览查看项目。

7.2 Linux 服务器

打开低代码在线开发-平台产品页面 (<https://www.thingjs.com/guide/offlineDeployment/>)，下载项目离线部署更新包（Linux 版），如下图所示：



将下载的“ThingJS 离线部署 v x.x.x 更新包 (Linux) .zip”更新包解压并放入部署的 linux 服务器中，执行安装命令即可开始安装更新包。命令为：`./xxxxxx.sh -target /原已部署目录`，如 `(./mysetup.sh --target /root/ThingJS/offline/)`

```

Creating directory /root/test/offline1
=====
                    离线部署 v3.2.11 更新包 For Linux
=====
ThingJS 离线部署 v3.2.11 更新包 Linux版&Win版同步上线。
ThingJS Server v3.2.11 update pakge will be available for Linux and Win simultaneously.

更多信息请访问 https://cdn.thingjs.com/doc/ThingJS项目离线部署包-用户手册 Rev.3.2.11.pdf
=====

正在验证离线部署 v3.2.11 更新包的完整性...
Verifying archive integrity of the ThingJS Server update pakge...

该离线部署 v3.2.11 更新包是完整的。
The ThingJS Server update pakge is fine.

> 更新前请确定 当前的离线部署版本在 v3.2.0 及以上? [Y/N]
> To confirm that your ThingJS Server version is v3.2.0 or higher ? [Y/n]
您的选择是:y
> 更新前请确定 离线部署服务 已经关闭 !!! [Y/N]
> Before updating, ensure that the ThingJS Server has been stopped!!! [Y/n]
您的选择是:y

安装中请耐心等待..... Please Waitting.....

=====
离线部署更新完成!!!
重新启动离线部署服务即可。
    
```

注：安装过程中会询问两个问题，第一个是确保原离线部署版本在 v3.2.0 及以上，若是则输入“y”即可进入下一步，第二个是确保原离线部署服务已经关闭，若是则输入“y”即可进入安装过程，直至更新安装完成。

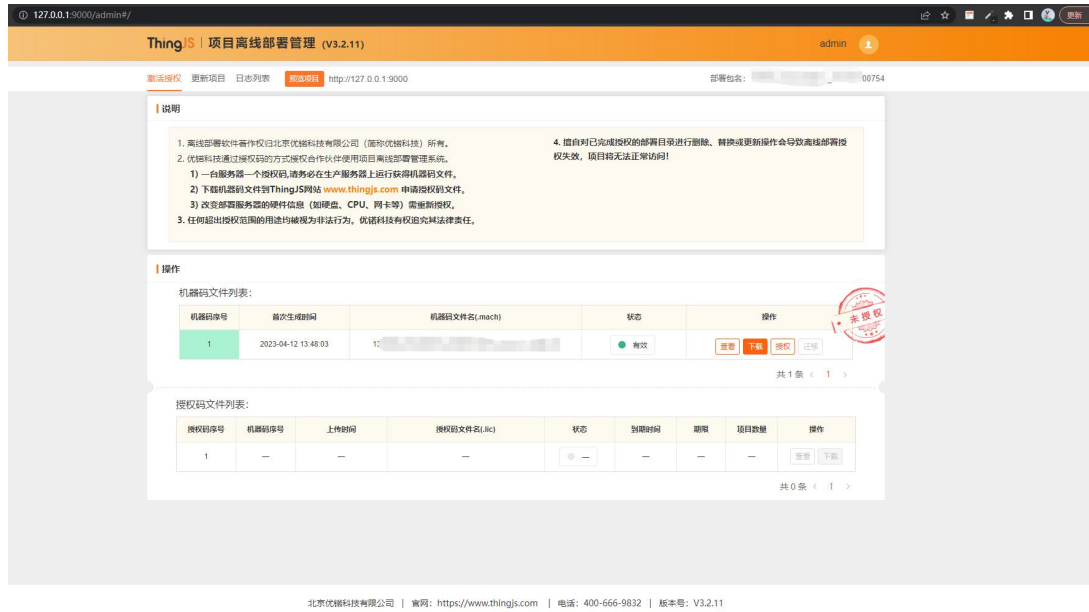
至此，已完成更新 v3.2.0 以上的部署服务；若您的早期部署版本低于（包含）v3.2.0 版本，请继续按以下步骤操作：

联系 ThingJS 商务人员获取 client.Linux-1.1.so 文件，并放至部署目录 system/lib 下；

```

[~/test/test/system/lib]# ll
10月 20 2021 7za
4月 12 13:53 client.linux-1.1.so
    
```

启动离线部署服务，访问授权界面，将新机器码文件发送至对应商务人员，重新给予授权；



授权成功后，本次部署服务升级将全部完成，您可点击预览查看项目。

8 迁移

变更部署服务器，或改变部署服务器的硬件信息（如硬盘、CPU、网卡等）需重新授权，如确需对**已永久部署**的“ThingJS 项目离线部署包”进行环境变更，请按本章节内容迁移“ThingJS 项目离线部署包”。

8.1 获取新机器码文件

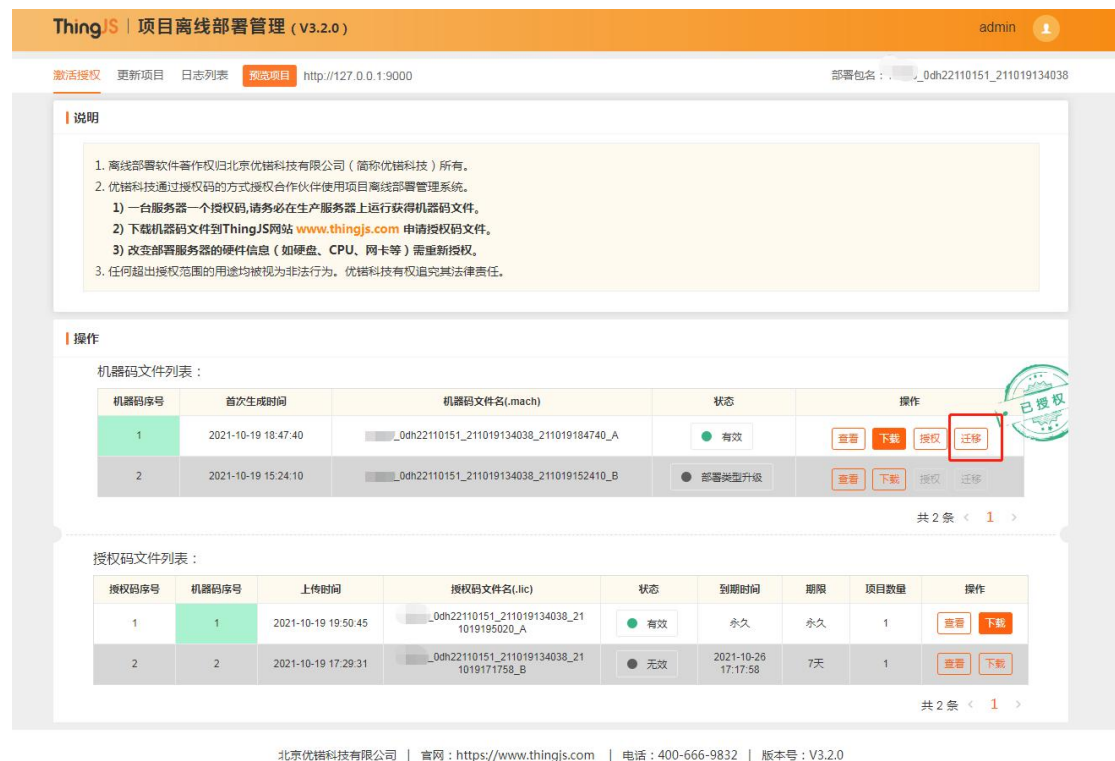
首先，获取到需要在新服务器环境中部署的离线部署包（永久部署包），该部署包可以是已永久部署的离线部署包的副本，也可以按第 2 章所述获取当前已部署项目的新的离线部署包（永久部署包）。

将获取到的离线部署包正确部署到新的服务器环境中，启动后访问部署在新服务器环境中的离线部署包的“项目离线部署管理”界面，下载当前有效的新的“机器码文件”至本地。

8.2 获取迁移码文件

返回到已永久部署的离线部署包的“项目离线部署管理”界面，点击“迁移”按钮，获取“迁移码文件”。

（注：执行迁移操作后，当前机器授权将失效，且无法再次授权激活。）



说明

1. 离线部署软件著作权归北京优诺科技有限公司（简称优诺科技）所有。
2. 优诺科技通过授权码的方式授权合作伙伴使用项目离线部署管理系统。
 - 1) 一台服务器一个授权码, 请务必在生产服务器上运行获得机器码文件。
 - 2) 下载机器码文件到ThingJS网站 www.thingjs.com 申请授权码文件。
 - 3) 改变部署服务器的硬件信息（如硬盘、CPU、网卡等）需重新授权。
3. 任何超出授权范围的用途均被视为非法行为。优诺科技有权追究其法律责任。

操作

机器码文件列表：

机器码序号	首次生成时间	机器码文件名(.mach)	状态	操作
1	2021-10-19 18:47:40	..._0dh22110151_211019134038_211019184740_A	● 有效	查看 下载 授权 迁移
2	2021-10-19 15:24:10	..._0dh22110151_211019134038_211019152410_B	● 部署类型升级	查看 下载 授权 迁移

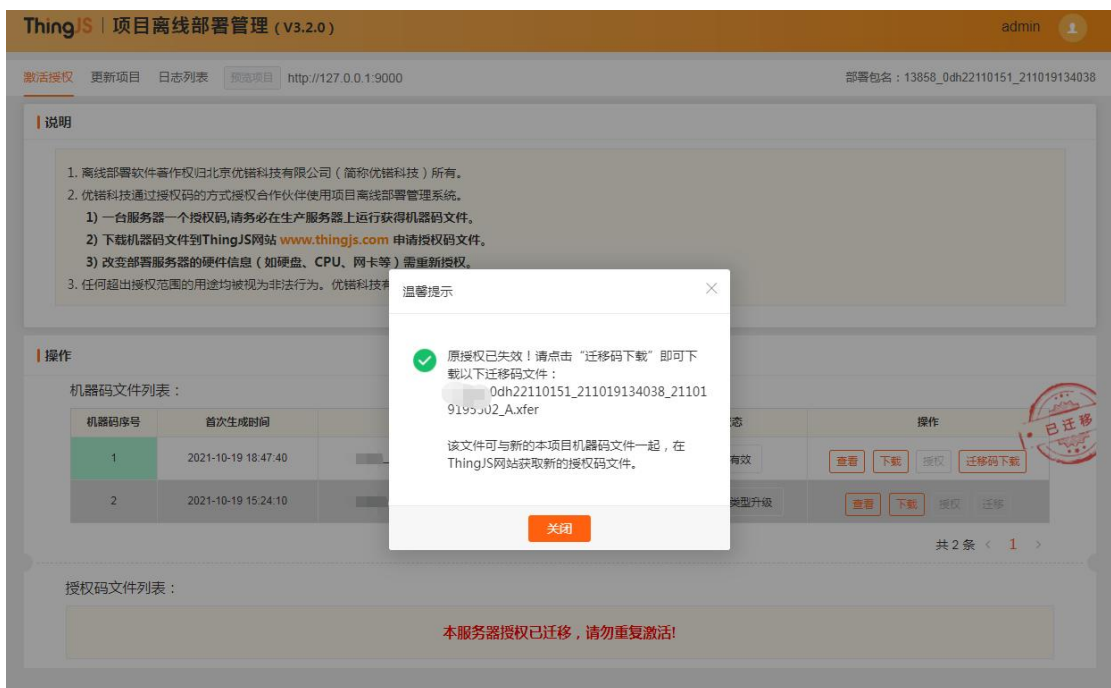
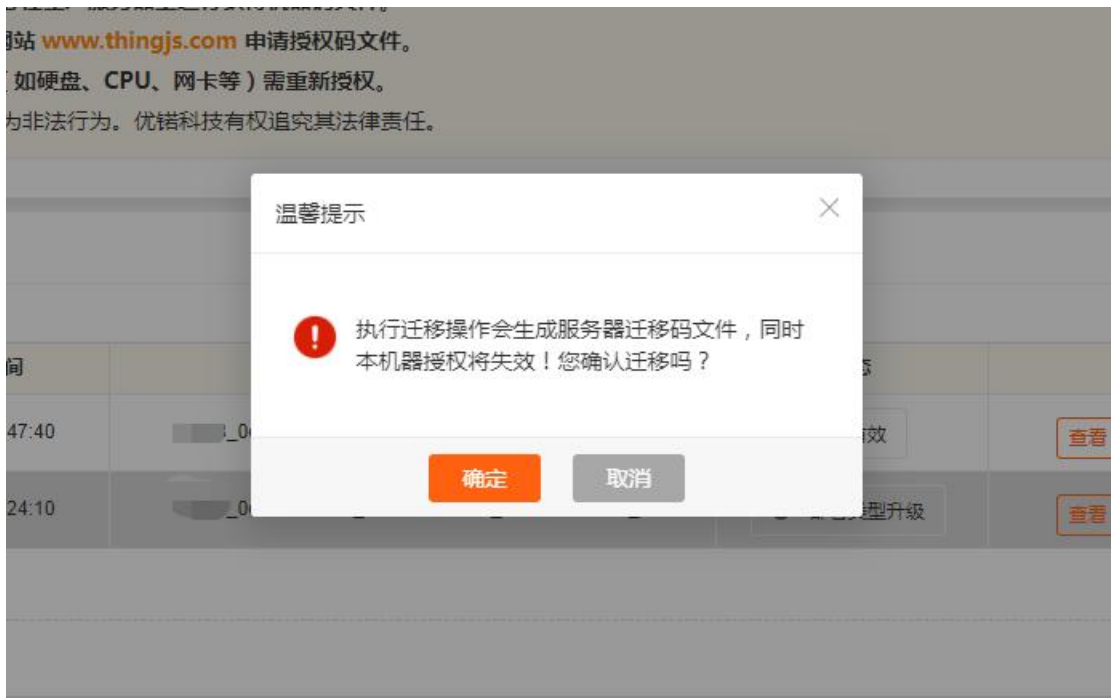
共 2 条 < 1 >

授权码文件列表：

授权码序号	机器码序号	上传时间	授权码文件名(.lic)	状态	到期时间	期限	项目数量	操作
1	1	2021-10-19 19:50:45	..._0dh22110151_211019134038_211019195020_A	● 有效	永久	永久	1	查看 下载
2	2	2021-10-19 17:29:31	..._0dh22110151_211019134038_211019171758_B	● 无效	2021-10-26 17:17:58	7天	1	查看 下载

共 2 条 < 1 >

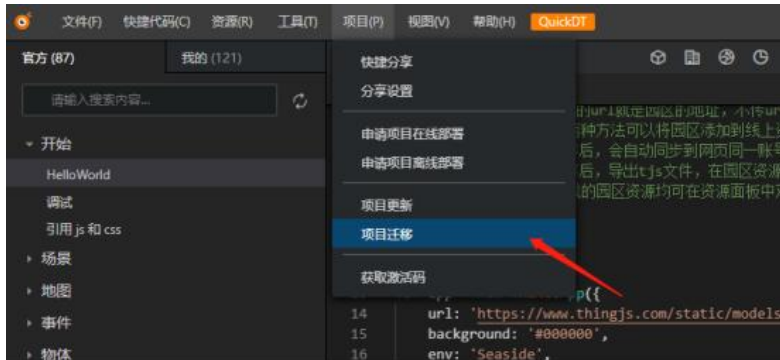
北京优诺科技有限公司 | 官网: <https://www.thingjs.com> | 电话: 400-666-9832 | 版本号: V3.2.0



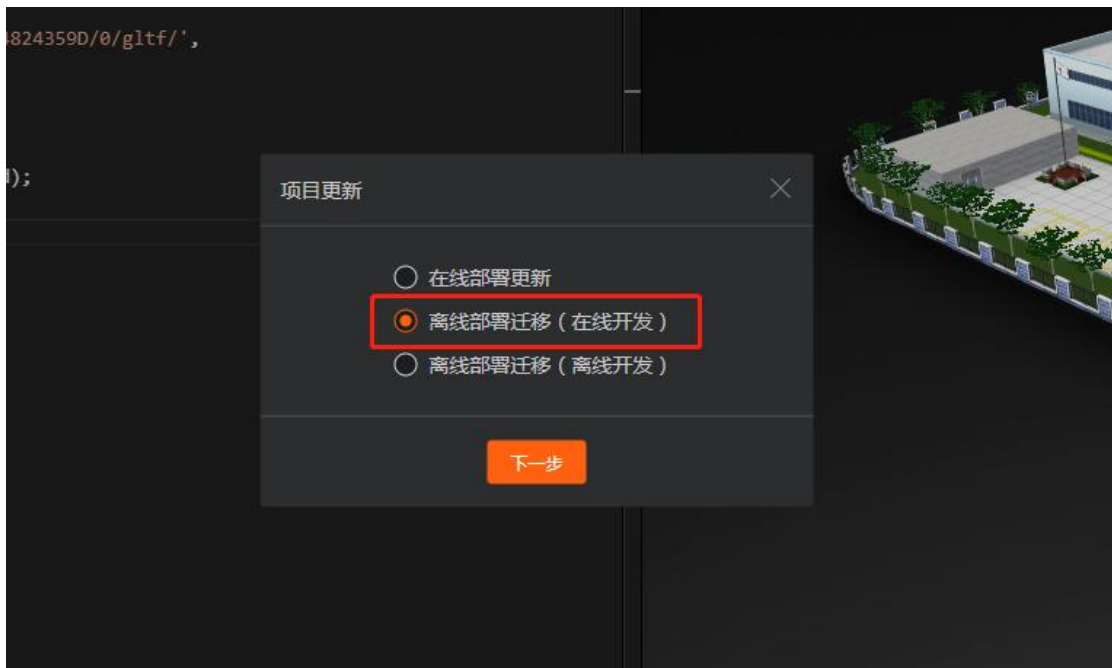
将生成的“迁移码文件”下载至本地。

8.3 获取新授权码文件

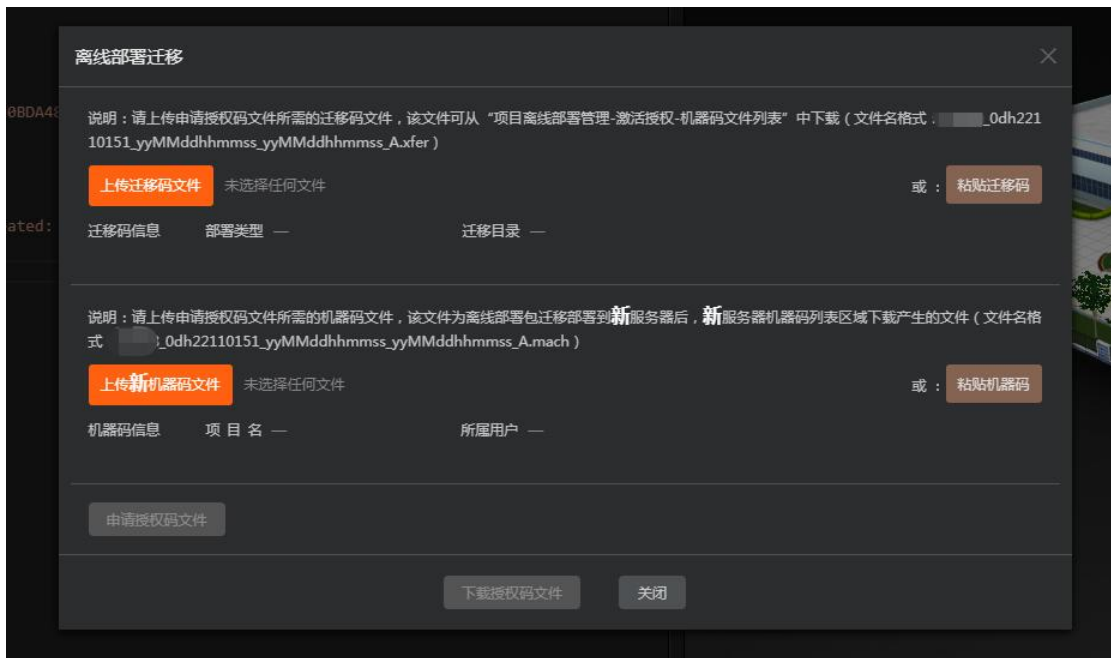
访问“ThingJS 在线开发平台”，打开已部署项目，点击“项目—项目迁移”。



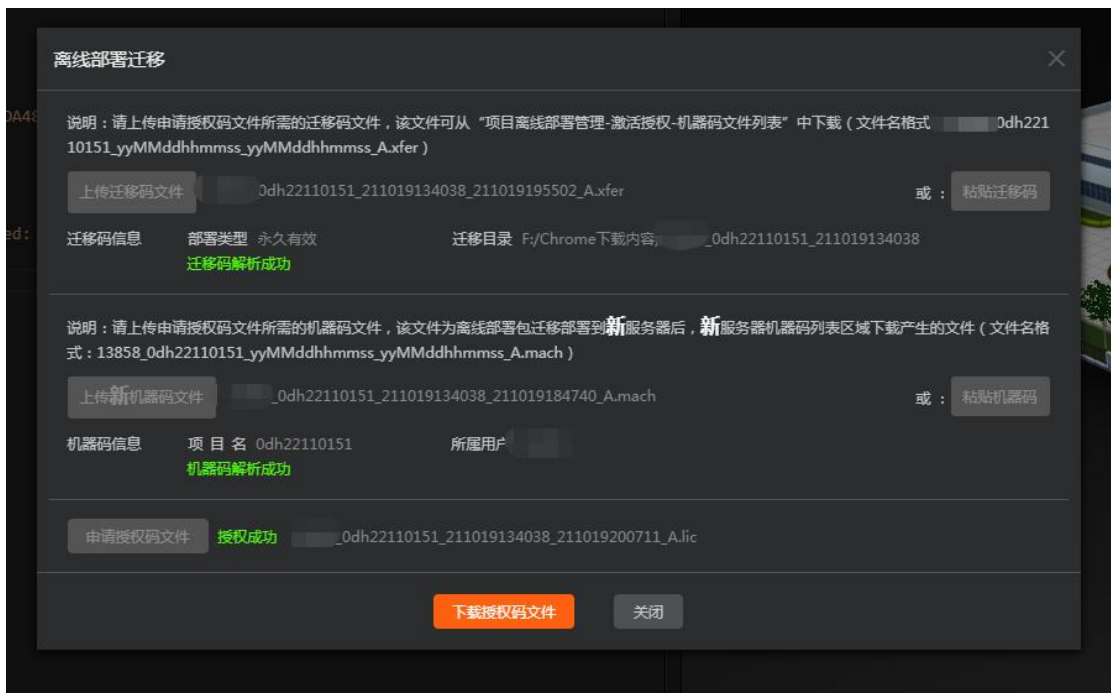
选择“离线部署迁移（在线开发）”。



在弹出的“离线部署迁移”面板中，从上到下依次上传 7.2 节获取的“迁移码文件”，和 7.1 节获取的“新机器码文件”。



成功上传该项目对应的正确的“迁移码文件”和“新机器码文件”后，即可申请“新机器码文件”对应的新的“授权码文件”，如下图所示：



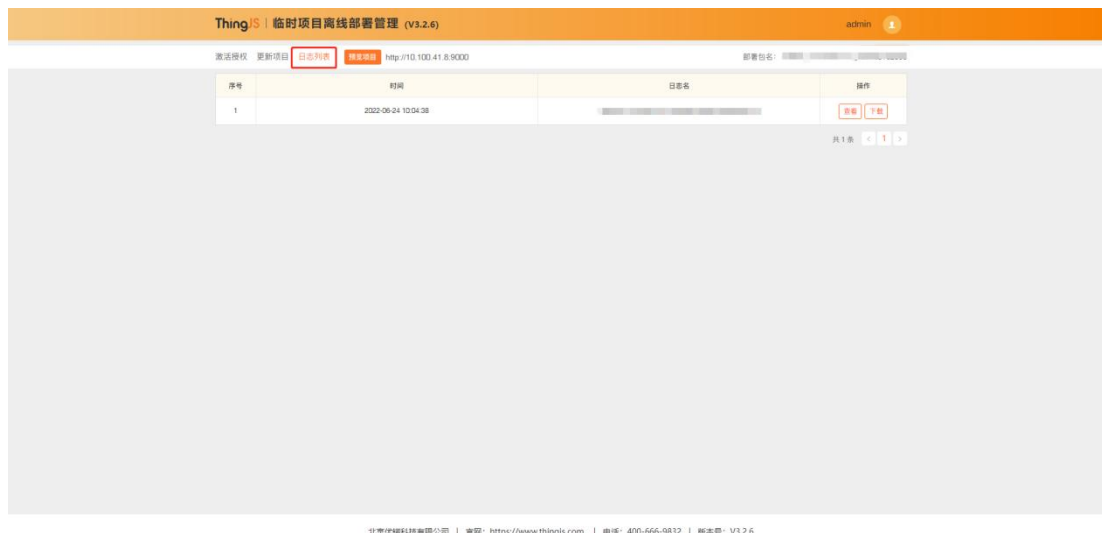
8.4 完成新服务器的授权

访问部署在新服务器环境中的离线部署包的“项目离线部署管理”界面，在当前有效的机器码文件一栏中选择“授权”按钮，上传 7.3 节中获取的新的对应的授权码文件，即可完成新服务器的授权。

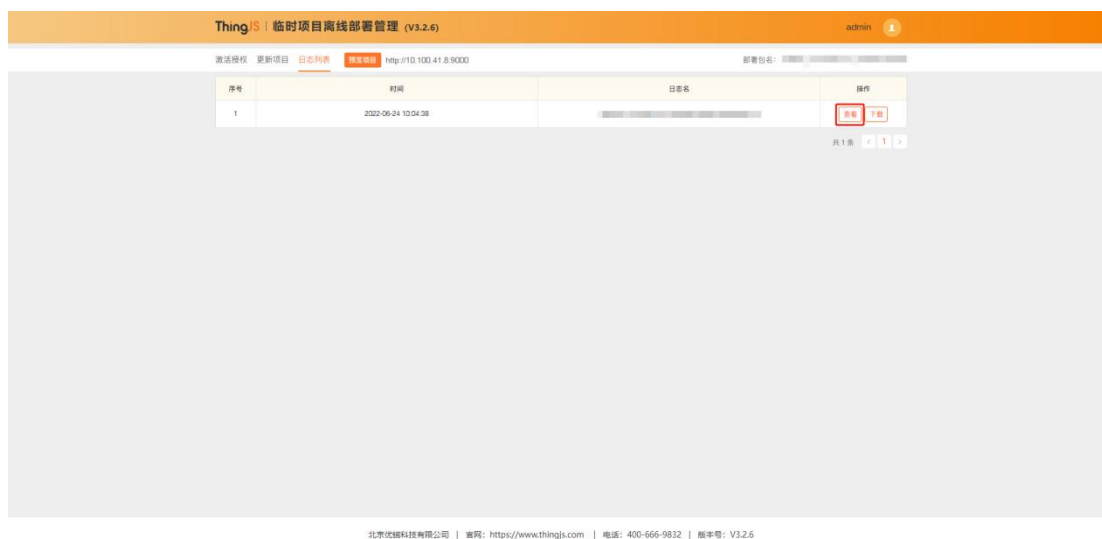
9 日志

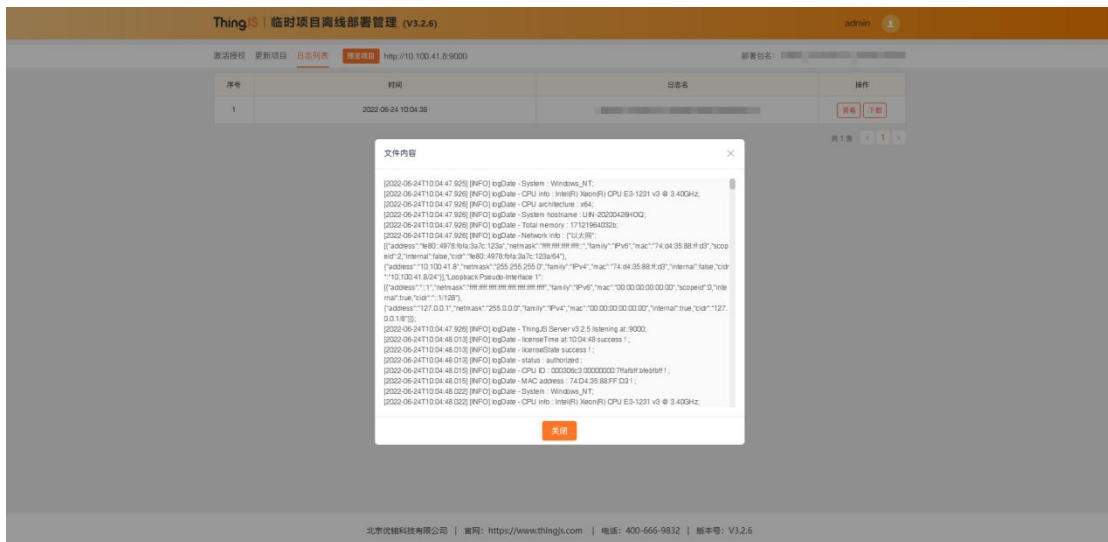
9.1 查看日志

访问“项目离线部署管理”页面，点击“日志列表”页签，如下图所示：



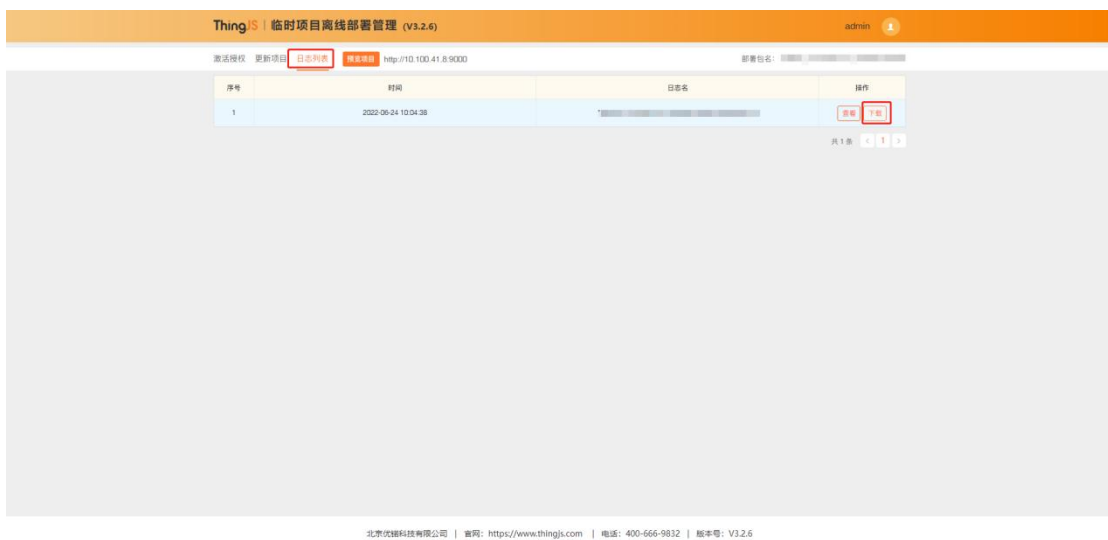
选择所需日期的日志文件，点击“查看”即可查看详细日志内容，如下图所示：



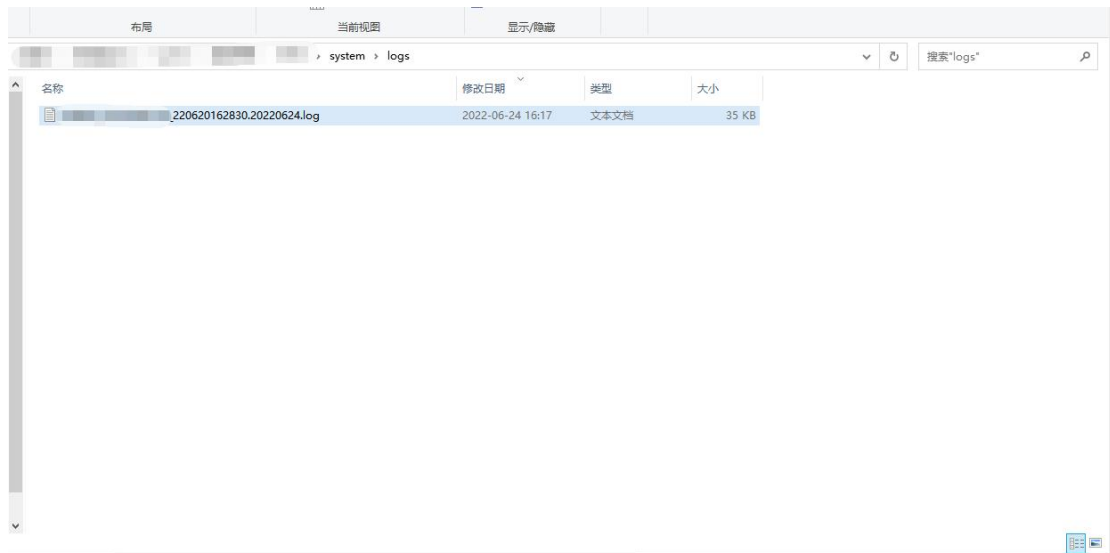


9.2 下载日志

访问“项目离线部署管理”页面，点击“日志列表”页签，选择所需日期的日志文件，点击“下载”即可获取日志文件，如下图所示：



日志文件也可在部署目录“/system/logs/”文件夹中获取，如下图所示：



10 常见问题 (FAQ)

10.1 在线开发正常运行项目，离线部署后访问 127.0.0.1:9000，3D 场景展示出错

答：用于部署的网络服务器一般不强调显示，故显卡较差，甚至无独立显卡。而 3D 场景显示对显卡要求较高，直接在服务器上 127.0.0.1 访问，服务器显卡不足以支持 3D 场景渲染展示，往往场景会显示出错。故可用有独立显卡的客户机浏览器访问该服务器场景，来验证 3D 场景显示是否正常。

10.2 Windows 环境下，start.exe 启动失败

答：常见的 start.exe 启动失败、发生闪退的情况是，当前 Windows 环境中离线部署服务的端口号已被其他服务占用，请检查端口占用情况，或为离线部署服务配置其他可使用的端口号。

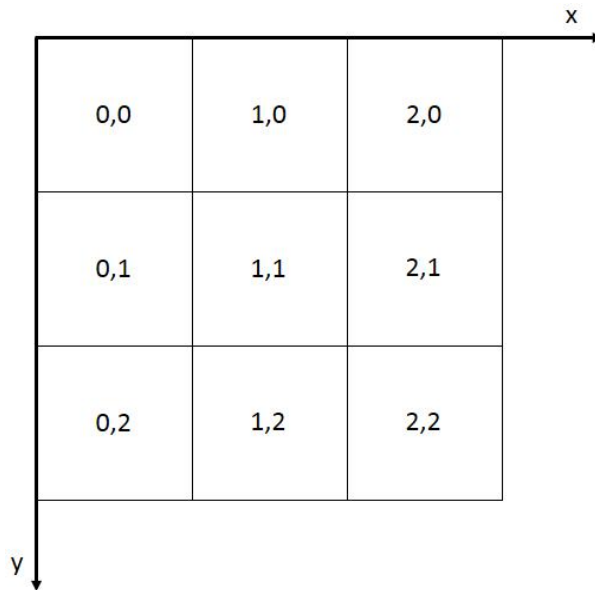
10.3 Windows 环境下，start.exe 卡住，服务无反应

答：这种情况一般是鼠标不小心点击了服务程序窗口，导致服务程序暂停，需手动于服务程序窗口中敲击回车来解决。

10.4 地图瓦片和离线前不一致

答：原因主要是地图瓦片资源未离线打包，打包时默认带了谷歌影像前 5 级的瓦片，所以我们需要参考第 4.3 节内容替换地图瓦片服务。关于支持的地图瓦片服务(ZXY 标准瓦片)格式说明如下：将地图全幅显示时的图片从左上角开始，

往下和往右进行切割，切割的大小默认为 256*256 像素，左上角的格网行号为 0，列号为 0，往下和往右依次递增。如下图所示：



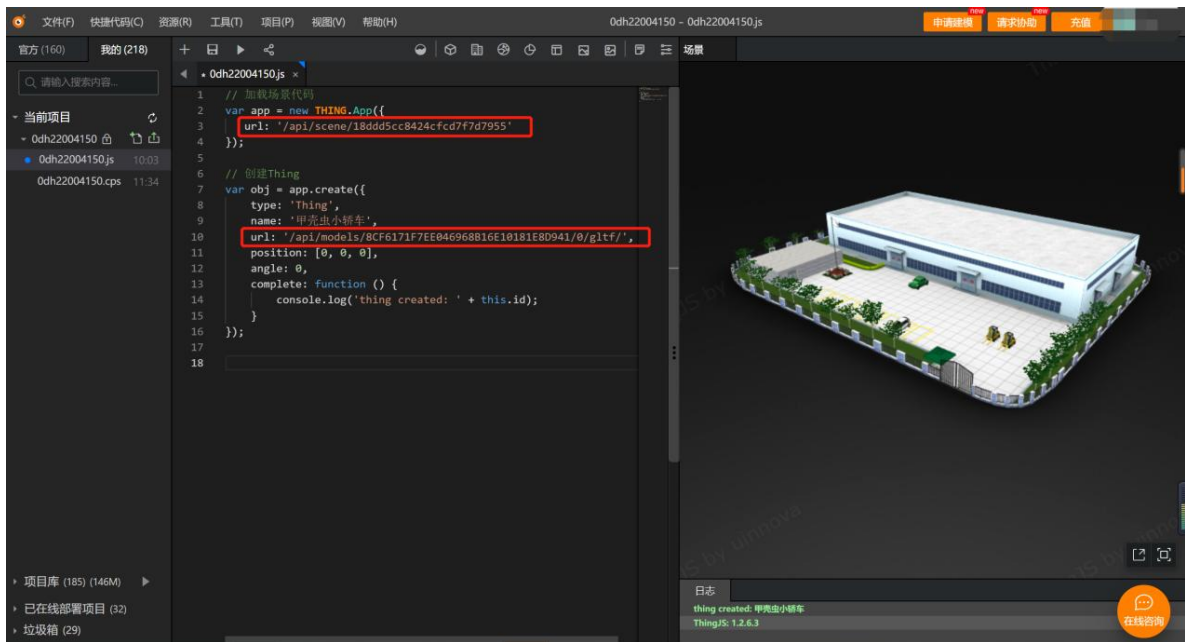
目前主流互联网地图基本都使用该格式地图瓦片，ArcGIS Server、SuperMap iServer 发布的地图瓦片服务也是 ZXY 标准瓦片。

10.5 场景、模型未离线打包

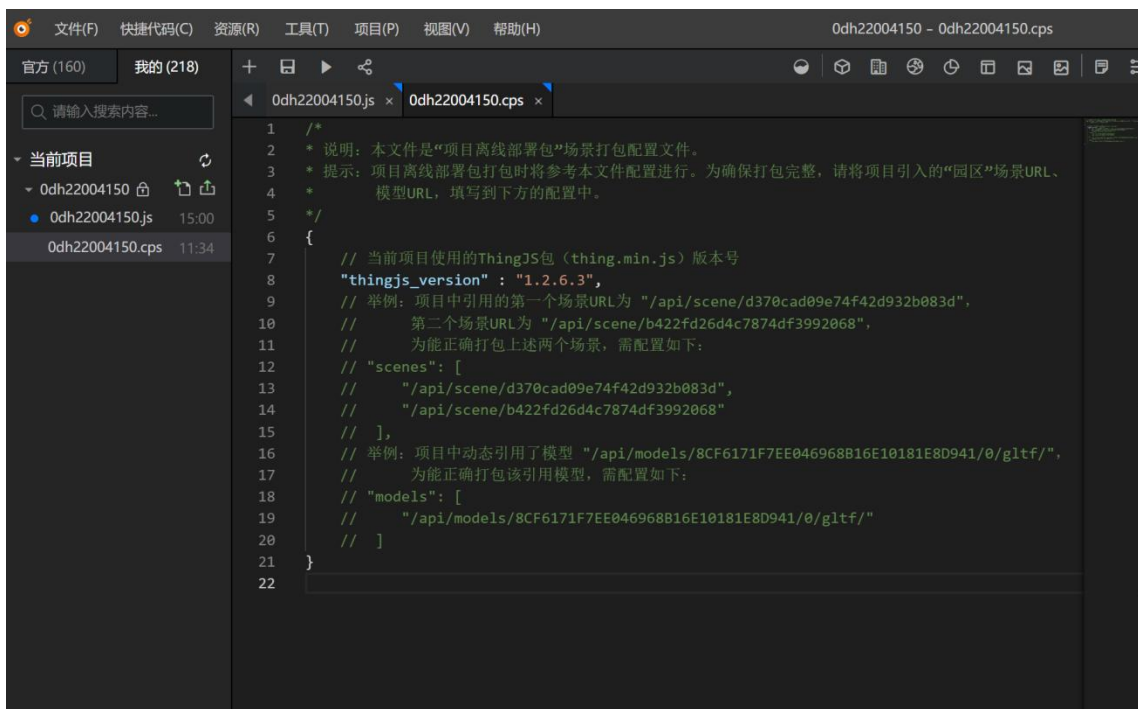
答：一般情况，离线程序会自动打包项目所引用场景和模型，因为开发人员程序写法多样，离线程序很难一一识别，所以加了 cps 配置文件，辅助打包。如遇项目引用场景或模型未离线打包的情况，需在在线开发中自行配置项目对应的 cps 文件并重新打包。

cps 文件配置方法如下：

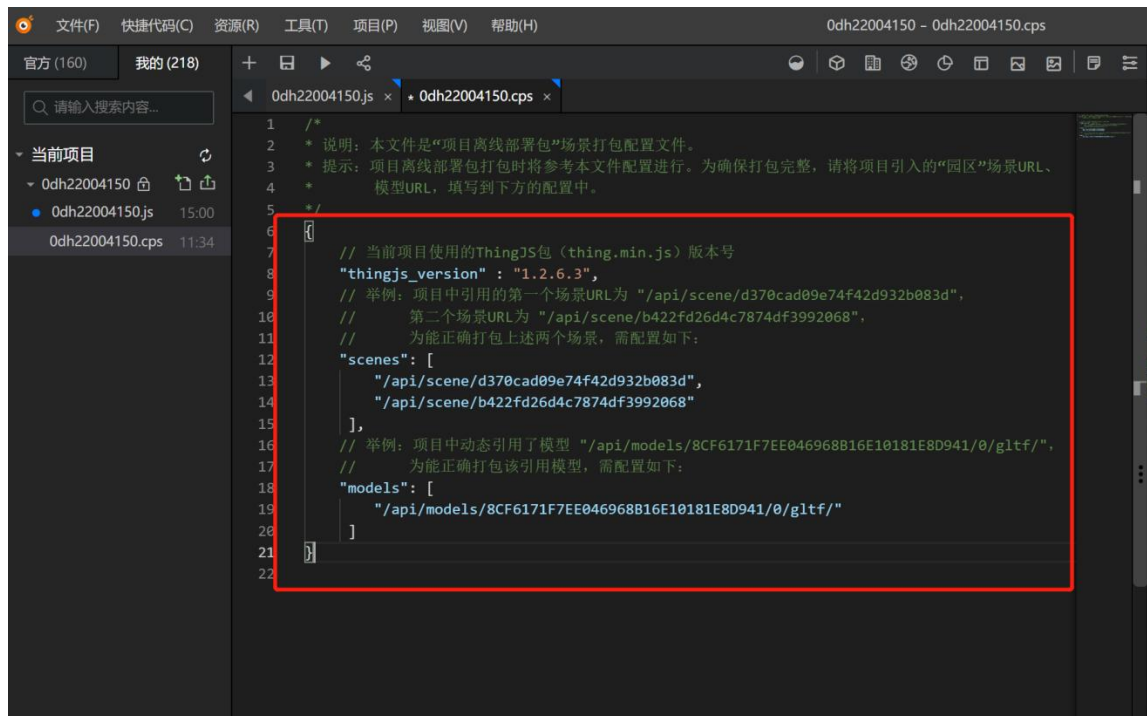
例如，在线开发中，需要进行离线部署的项目如下图所示：



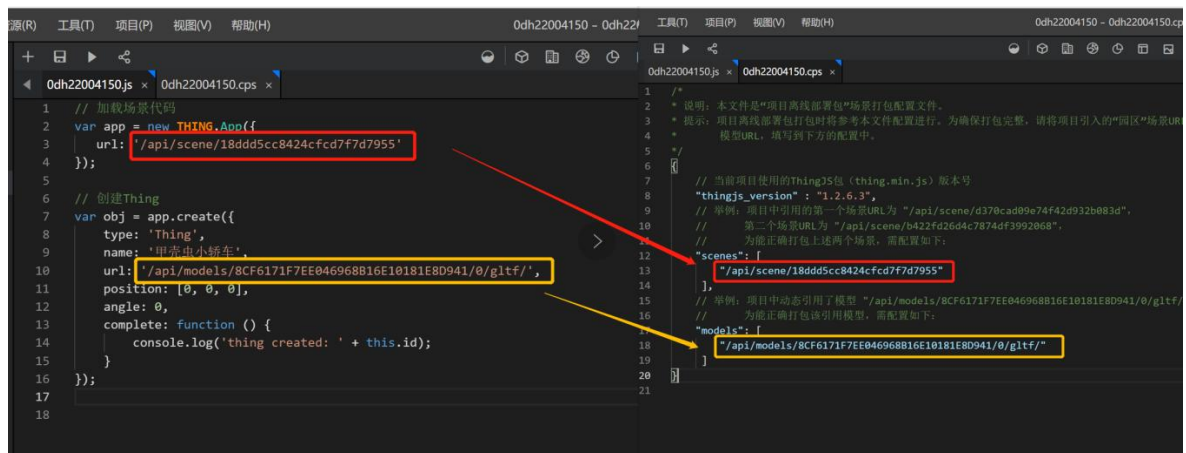
项目中引用了一个场景 ('/api/scene/18ddd5cc8424cfd7f7d7955'), 和一个模型 ('/api/models/8CF6171F7EE046968B16E10181E8D941/0/gltf/'). 点击打开当前项目下该项目对应的 cps 文件, 初始 cps 文件内容如下图所示:



将 cps 文件下面的配置示例注释打开。



将需要离线的场景 URL 填入 scenes 配置项中，模型 URL 填入 models 配置项中。

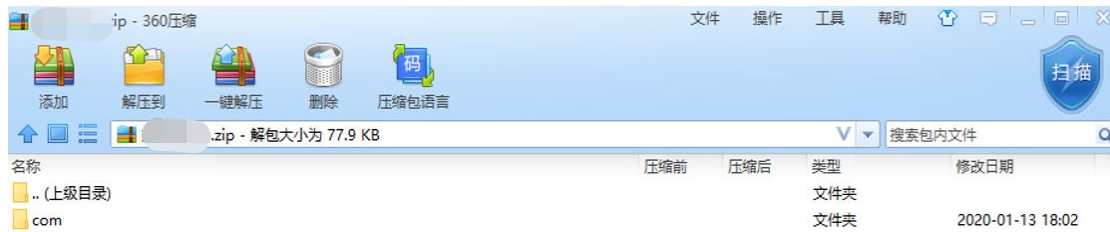


保存后重新下载离线部署包。

场景文件将打包至离线部署包的/web/scene/目录下，模型文件将打包至离线部署包的/web/models/目录下。

10.6360 解压离线部署包有问题

答：360 压缩软件解压离线部署包会出现如下情况：



请使用 7-Zip 或其他支持 7z 格式的压缩软件进行解压。

10.7 倾斜摄影数据访问不到，控制台报错 404

答：两种解决方案。

方案一：将 web/lib/tileset.json 该文件中的特殊字符 ‘+’ 进行转义处理，使用编辑器打开该文件之后，全局替换 ‘+’ 字符变成 ‘%2B’ 字符。修改之后，保存即可。重新刷新项目之后既可访问。

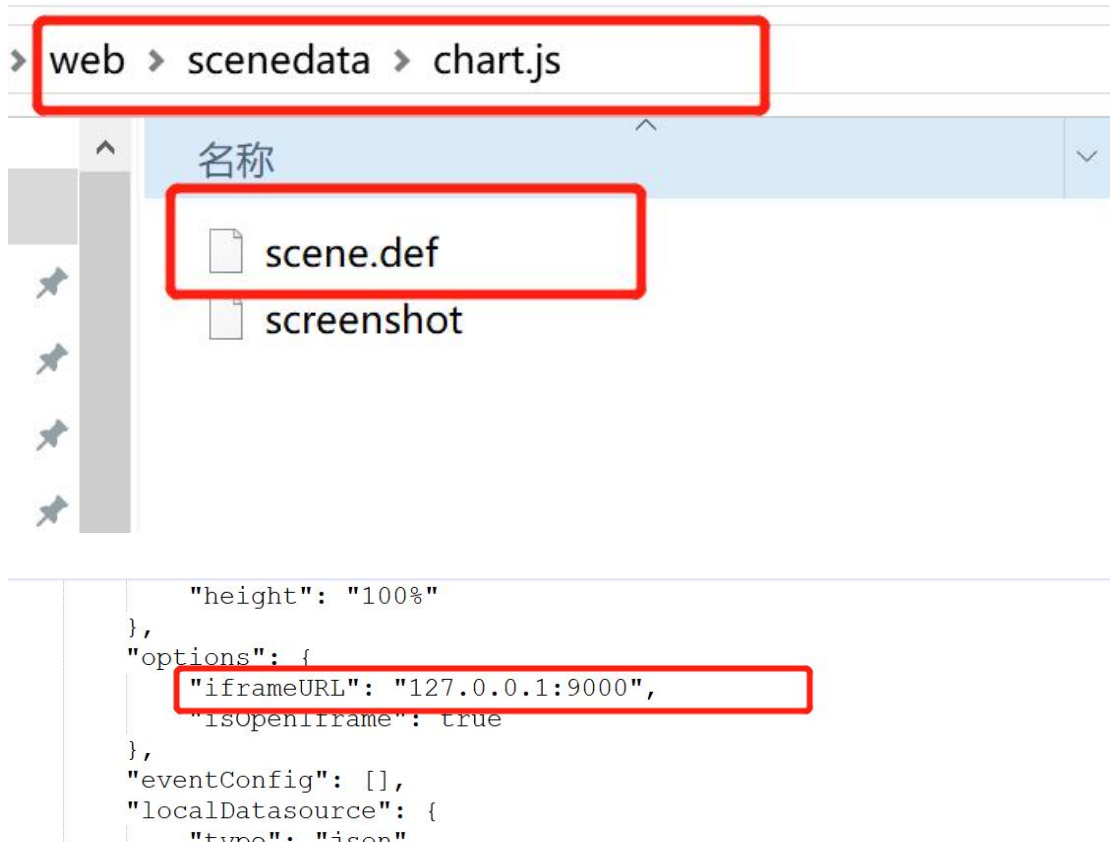
方案二：单独启动一个服务，来访问倾斜摄影的数据，推荐使用 node.js 启动服务。首先在命令行中安装 http-server 服务，可以输入 `npm install http-server -g` 进行安装。安装完成之后进入 web\lib 目录下，输入命令 `http-server --cors -p 8088`，来启动服务。启动之后修改脚本中倾斜摄影地址为 `http://{本机ip}:8088/tileset.json` 即可。

10.8 ChartBuilder 项目离线部署后看不到图表内容

答：需要在访问端口后拼/chart 来查看 ChartBuilder 项目，例如：
<http://127.0.0.1:9000/chart>（服务地址:服务端口/chart）

10.9 ChartBuilder 项目离线部署后显示 127.0.0.1 拒绝了连接请求

答：ChartBuilder 项目离线部署后，默认为 3D 内容配置的是本地 9000 端口，如使用 ip+自定义端口的方式访问，需要自行修改 scene.def 文件中的 iframeURL 为使用的 ip+端口。如下图所示：

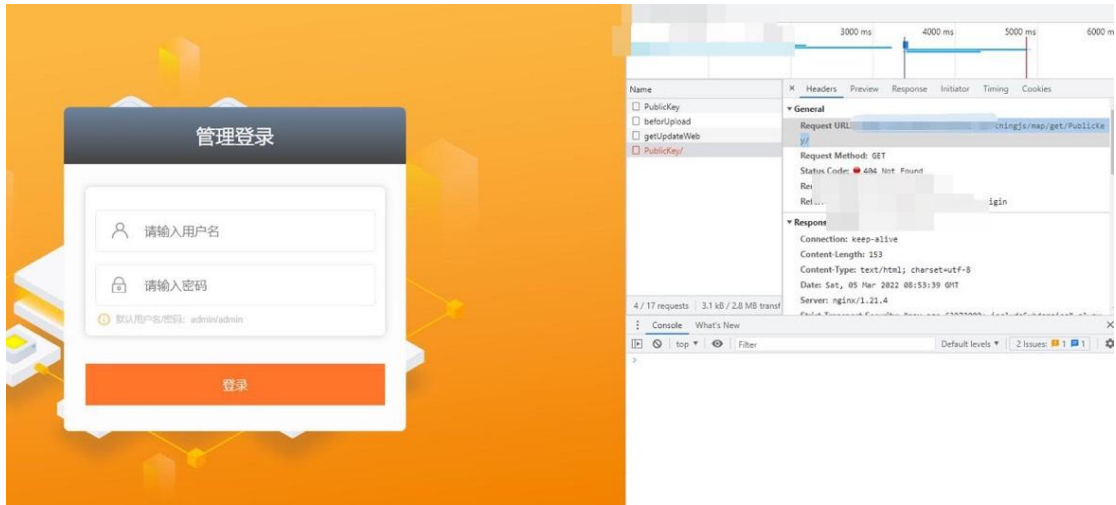


10.10 ThingJS 项目离线部署包 V3.2.0 版本与旧版本相比，有什么区别

答：ThingJS 项目离线部署包 V3.2.0 版本与旧版本相比，区别如下：

- (1) 增加了“项目离线部署管理”界面；
- (2) 强调了“临时部署包”和“永久部署包”两个概念，具体请查看本文档的第 1.3 节内容；
- (3) 原“机器码”、“授权码”全部升级为“文件”的形式；
- (4) 新增“登录”功能；
- (5) 新增“更新项目”功能；
- (6) 完善“日志系统”，如遇到部署问题请下载日志文件发送给平台客服人员；
- (7) 完善“错误码信息”，如遇到授权失败问题，请先查看本文档第 9 章的“错误码对照表”。

10.11 更新项目后，控制台接口 “/get/PublicKey” 报 404



答：是后台程序未更新，导致接口 “/get/PublicKey” 没有请求到，请手动将离线部署包中的 start 程序替换为最新版本并重启。

10.12 服务配置在二级路由下，控制台报错 404

答：两种解决方案。

方案一：Nginx 配置如下两个路径。注：bigMap 为需要的二级路由名

```
server {
    listen      80;
    server_name localhost;

    #charset koi8-r;

    #access_log logs/host.access.log main;

    location /bigMap/{
        proxy_pass http://192.168.10.94:9000/;
        proxy_pass_header Server;
    }
    location / {
        proxy_pass http://192.168.10.94:9000/;
        proxy_pass_header Server;
    }

    #error_page 404              /404.html;

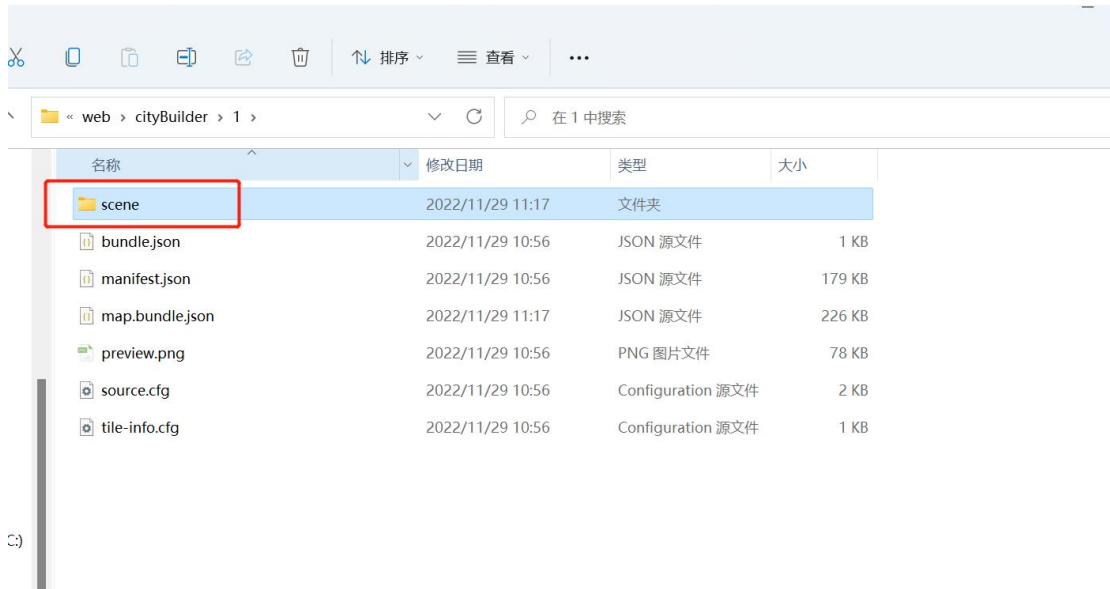
    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   html;
    }
}
```

方案二：手动将项目中引用到的绝对路径改为相对路径。

10.13 森城市携带场景离线部署后场景加载不到该如何配置

答：森城市携带场景离线部署后配置操作步骤：

打开/web/cityBuilder 中查看对应地图场景目录下是否有 scene 目录，若没有 scene 目录，需到在线开发重新进行离线部署打包。如下图所示：



打开/web/cityBuilder 中对应地图场景目录下的 source.cfg 文件，找到配置场景的地方：



将 source.cfg 文件中的场景路径改为相对路径，如图所示：

```

    },
    "scenes": [
      {
        "name": "scene",
        "angle": 180,
        "id": 1033547,
        "url": "/cityBuilder/1/scene/1033547/",
        "lonlat": [116.39497,
        39.903316],
        "height": 1.0
      },
      {
        "name": "未命名园区",
        "angle": 180,
        "id": 1033548,
        "url": "/cityBuilder/1/scene/1033547/",
        "lonlat": [116.39497,
        39.903316],
        "height": 1.0
      }
    ],
    "terrainUrl": ""
  }
}
    
```

打开加载地图代码块的文件，添加如下配置：

```

var app = new THING.App();
app.background = [0, 0, 0];
THING.Utils.dynamicLoad(['./source/uearth.min.js'], function () {
  app.create({
    type: 'Map',
    url: "./cityBuilder/1/map.bundle.json",
    resourceConfig: {
      resourcePrefix: './',
      loadDataFirst: true,
      isProxima: true,
      changeCampusLevel: false,
      externalConfigUrl: './cityBuilder/1/source.cfg'
    },
    complete: function (event) {
      console.log(event.object.userLayers.length);
    }
  });
});
    
```

10.14 加载森拓扑资源时，有资源报 404 该如何解决

```

加载完成
GET http://... 404 (Not Found)
GET http://... 404 (Not Found)
    
```

答：在初始化拓扑图时，添加 resourceRootPath 字段，配置拓扑图静态资源的根路径即可。如下图所示：

```
// 创建APP对象
var app = new THING.App();
// 引用插件组件脚本
THING.Utills.dynamicLoad(['./static/js/thing.diagram.min.js'], function () {
  // 初始化插件场景
  const graph = new THING.DIAGRAM.Graph({
    container: 'div2d', // 容器元素 id
    url: './diagrams/88f886350afffc3b/topo.json', // 拓扑场景资源路径
    resourceRootPath: './diagrams/88f886350afffc3b/'
  });
  // 视图加载完成
  graph.on('load', () => {
    console.log('加载完成');
  });
});
})
```

10.15 修改配置文件不生效如何解决

答：修改配置文件中的内容不生效，如端口号修改为 9001，但是无法通过 9001 端口访问，仍需通过 9000 端口访问。

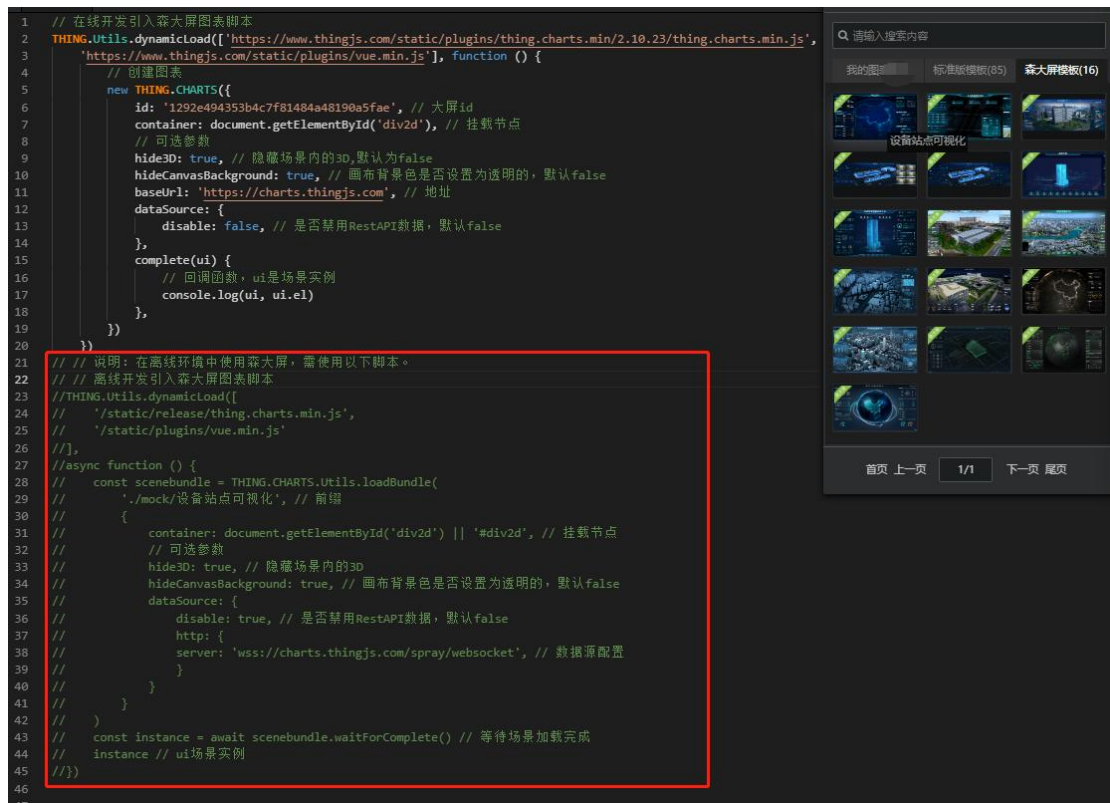
解决办法如下：

- 1、修改配置文件后需要重新启动离线部署服务；
- 2、配置文件中修改配置项时，“=”前不可有空格。



10.16 森大屏资源加载不出来

答：查看离线部署包 web 目录下是否有 mock 资源，如无则检查代码块是否是在主文件中编写，需在主文件中添加森大屏加载代码块重新进行打包；如有 mock 资源仍无法加载，则查看代码块是否切换为离线加载版本，如下图所示：



10.17 天空盒资源仍请求线上地址

答：在加载场景时，添加配置项 `resourceLibraryUrl: "./"`，如下图所示：



10.18 森大屏资源离线加载时，数据对接错误

答：离线部署包中默认不具备森大屏接口服务请求功能，离线环境下若要使用数据对接服务，需要按以下步骤进行操作：

获取 `dataSource-server` 数据对接服务（软件依赖 JRE V1.8~V1.10）：

<https://charts.thingjs.com/s-static/external/thing.charts.min/demo/dataSource-server.zip>

将 `dataSource-server` 数据对接服务 zip 包解压至部署环境中，启动服务，

windows 环境中启动 `dataSource-server/bin/start.bat`，linux 环境中启动

`dataSource-server/bin/start.sh`。

数据对接服务默认端口为 7000，可在 dataSource-server/conf/application.yml 文件中进行配置。

修改离线部署项目中引用森大屏的配置，将数据源更改为离线数据对接服务地址（ws://ip:port/spray/websocket），如下图所示：

```
// 说明：在离线环境中使用森大屏，需使用以下脚本。
// 离线开发引入森大屏图表脚本
THING.Utills.dynamicLoad([
  '/static/release/thing.charts.min.js',
  '/static/plugins/vue.min.js'
]),
async function () {
  const scenebundle = THING.CHARTS.Utills.loadBundle(
    './mock/设备站点可视化', // 前缀
    {
      container: document.getElementById('div2d') || '#div2d', // 挂载节点
      // 可选参数
      hide3D: true, // 隐藏场景内的3D
      hideCanvasBackground: true, // 画布背景色是否设置为透明的，默认false
      dataSource: {
        disable: true, // 是否禁用RestAPI数据，默认false
        http: {
          server: 'ws://ip:port/spray/websocket', // 数据源配置
        }
      }
    }
  )
  const instance = await scenebundle.waitForComplete() // 等待场景加载完成
  instance // ui场景实例
})
```

11 错误码对照表

错误码	说明	解决方法
10001	服务异常	查看日志文件或寻找平台客服人员协助
50101	授权码已过期	申请新的授权码文件
50103	当前服务器授权已迁移	授权已迁移，当前服务不可重复激活
50104	当前目录未授权	将当前服务更改至授权目录
50105	离线项目数量超出授权	删除超出的项目或申请对应项目数量的授权码文件
50106	授权码解析错误	检查授权码文件内容
50107	机器匹配失败，当前部署机器未授权	检查部署机器信息
50108	未找到匹配授权码文件	申请授权码文件
50109	授权目录读取失败	检查部署包内容变动
50110	授权项目信息读取失败	检查部署包内容变动
50111	授权码文件格式错误	检查授权码文件格式
50112	授权码文件名与机器码文件不匹配	检查授权码文件命名
50113	授权码文件过大	检查文件内容
50114	授权码文件命名格式错误	检查授权码文件命名，请勿自行更改文件命名
50115	授权码文件已失效	申请新的授权码文件
50116	授权码类型不匹配（永久部署包）	为永久部署包申请永久授权码文件
50117	授权码类型不匹配（临时部署包）	为临时部署包申请临时授权码文件

12 修订历史

版本	日期	更新内容描述
Rev. 2.0.0	2018/09/01	初版离线部署包使用说明。
Rev. 2.2.0	2020/08/27	新增迁移离线部署包功能。
Rev. 3.0.0	2021/02/01	离线部署服务形式更新升级。
Rev. 3.2.0	2021/10/21	新增离线部署管理内容； 新增离线部署更新功能的说明。
Rev. 3.2.1	2022/01/07	优化离线部署服务接口。
Rev. 3.2.2	2022/02/22	优化管理接口； 增加服务请求配置； 修复引用版本。
Rev. 3.2.2	2022/03/16	新增常见问题（FAQ）10.11。
Rev. 3.2.3	2022/04/28	优化管理界面； 新增常见问题（FAQ）10.12。
Rev.3.2.4	2022/06/13	修复服务性能问题； 增加多地图打包支持。
Rev.3.2.5	2022/06/20	增加森城市打包支持。
Rev.3.2.6	2022/06/30	增加日志获取说明； 修复 ChartBuilder 项目预览； 升级终端信息。
Rev.3.2.7	2022/07/07	修改 jQuery 版本。
Rev.3.2.8	2022/07/28	修复引用文件问题； 更新离线部署管理页面提示。
Rev.3.2.9	2022/12/06	支持在线开发 initThingJsTip 方法； 修复 obj 模型访问失败问题。 新增常见问题（FAQ）10.13。
Rev.3.2.10	2023/01/10	新增 4.8 节。
Rev.3.2.11	2023/03/07	新增 4.9 节 新增 7 节。 新增常见问题（FAQ）10.14。 新增常见问题（FAQ）10.15。
Rev.3.2.12	2023/04/19	新增接口校验
Rev.3.2.13	2023/05/08	修复更新接口