



ThingJS 离线开发网络版 用户手册

(Rev.1.2.2)

北京优锆科技有限公司

2022 年 01 月 08 日

目 录

1	概述.....	3
1.1	说明.....	3
1.2	硬件配置.....	3
1.3	软件依赖.....	4
2	获取.....	5
3	安装.....	6
3.1	Windows 服务器.....	6
3.2	Linux 服务器.....	7
4	配置.....	10
4.1	修改服务启动端口.....	10
4.2	修改登录密码.....	10
5	授权.....	13
5.1	访问“离线开发网络版”管理界面.....	13
5.2	获取授权.....	14
6	开发.....	17
6.1	搭建场景.....	17
6.2	应用开发.....	20
6.3	离线部署.....	32
7	迁移.....	34
7.1	获取新机器码文件.....	34
7.2	获取迁移码文件.....	34
7.3	获取新授权码文件.....	35
7.4	完成新服务器的授权.....	35
8	常见问题 (FAQ).....	36
8.1	访问 3D 场景展示出错.....	36
8.2	Windows 环境下, start.exe 启动失败.....	36

8.3	Windows 环境下，start.exe 卡住，服务无反应	36
9	修订历史.....	37

1 概述

“ThingJS 离线开发网络版”是 ThingJS PaaS 平台 (<https://www.thingjs.com>) 推出的可在独立局域网环境运行的离线开发 ThingJS 3D 可视化（或数字孪生）项目的开发服务器产品。“ThingJS 离线开发网络版”专为企业内有较多 ThingJS 开发人员需协同完成 ThingJS 项目开发的企业打造。

本文档是说明“ThingJS 离线开发网络版”产品安装、使用、迁移等各方面技术问题的用户手册。

1.1 说明

ThingJS 3D 可视化开发平台提供在线开发、离线开发两种开发方式。其中离线开发又分离线开发 SDK 版（坐席版）和离线开发网络版。

ThingJS 离线开发 SDK 版提供一个 U 盘形式的 Key，插入特定计算机，该机就可以通过离线开发 SDK 软件包进行 ThingJS 离线开发了。离线开发 SDK 版从形式上是单坐席授权，每个 Key 对应一台计算机可进行开发。比较适合企业内仅投入 1 个或少数人员进行 ThingJS 开发的情况。而对于企业有多位 ThingJS 开发人员要投入开发，如果购买多个离线开发 SDK 版授权来支持，这在成本、便捷度两方面都不是特别合适。

ThingJS 离线开发网络版是一个可在局域网环境部署的开发服务器。它通过和开发版本管理工具 Git 结合，理论上可支持不限人数的 ThingJS 开发人员共同进行项目开发。适合于企业内有较多开发人员会参与 ThingJS 开发的情况。离线开发网络版所支持的本机开发 IDE（VSCode）+Git 方式，在开发习惯跟大部分开发人员的日常工作习惯相同，可让开发人员在开发方式上完成无缝切换。

若您希望咨询、了解、购买“ThingJS 离线开发网络版”，可到 ThingJS 网站 (<https://www.thingjs.com>) 联系网站客服人员，或致电 ThingJS 400 电话：400-666-9832。

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：不支持在 CPU 为 ARM 的服务器上部署）

1.2.2 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 软件依赖

1.3.1 服务器端

“ThingJS 离线开发网络版”基于 Git 对开发项目进行管理，因此需要在部署安装“ThingJS 离线开发网络版”的环境中安装 Git。

Git 是一个开源的分布式版本控制系统，用于敏捷高效地处理任何或小或大的项目。可在 <https://mirrors.edge.kernel.org/pub/software/scm/git/> 中下载安装包。

详细的安装步骤可参考第三章中关于 Git 安装的内容。

1.3.2 客户端

1.3.2.1 CampusBuilder 客户端

CampusBuilder 在园区级 3D 可视化场景的搭建方面，功能强大，不仅可以搭建园区场景，在建筑楼层和室内搭建方面也表现卓越。

最新版本可在 <https://store.thingjs.com/tools> 中下载。

1.3.2.2 Visual Studio Code 编辑器（推荐）

Visual Studio Code 是一款针对于编写现代 Web 和云应用的跨平台源代码编辑器，推荐使用 Visual Studio Code 进行项目开发。

最新版本可在 <https://code.visualstudio.com/> 中下载。

1.3.2.3 Git

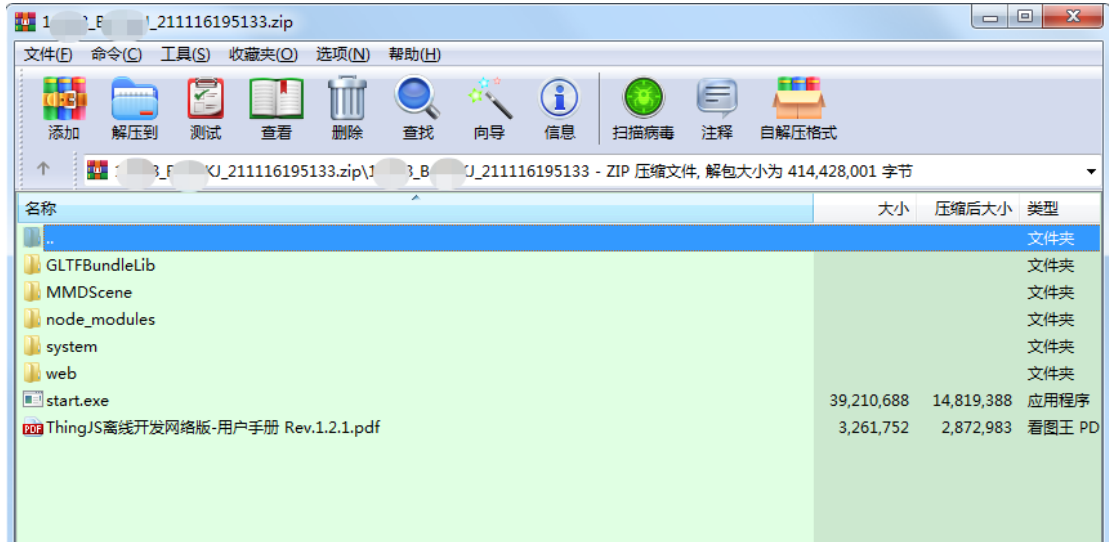
“ThingJS 离线开发网络版”基于 Git 对开发项目进行管理，因此需要在进行开发的客户端环境中安装 Git。

可在 <https://git-scm.com/download> 中下载安装包。

详细的安装步骤可参考第三章中关于 Git 安装的内容。

2 获取

同 ThingJS 商务人员购买“ThingJS 离线开发网络版”后，告知商务人员将要部署安装“ThingJS 离线开发网络版”的环境（Linux/Windows），将收到商务人员发送的对应环境的“ThingJS 离线开发网络版”压缩安装包。如下图所示：



3 安装

3.1 Windows 服务器

3.1.1 安装 Git

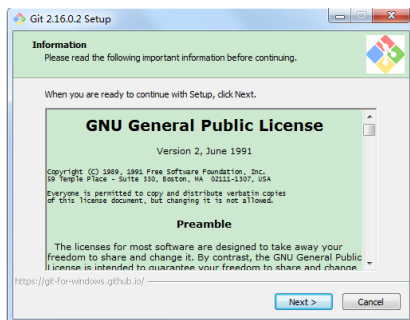
“ThingJS 离线开发网络版”基于 Git 对开发项目进行管理，因此需要在部署安装“ThingJS 离线开发网络版”的环境中安装 Git。

3.1.1.1 下载安装包

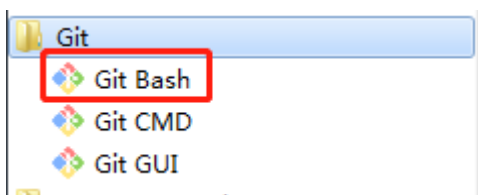
访问网址 <https://gitforwindows.org/> 下载安装包（官网速度慢，可使用国内的镜像地址进行下载 <https://npm.taobao.org/mirrors/git-for-windows/>，推荐使用稳定版本 v2.16.0.windows.2）。

3.1.1.2 安装

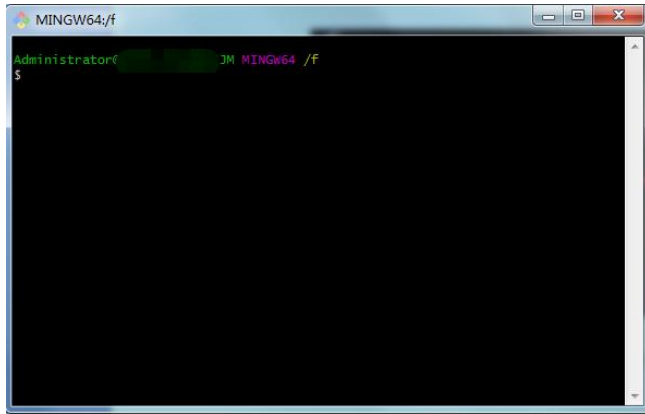
运行安装文件，根据提示内容完成安装。



安装完成后，运行“Git Bash”。

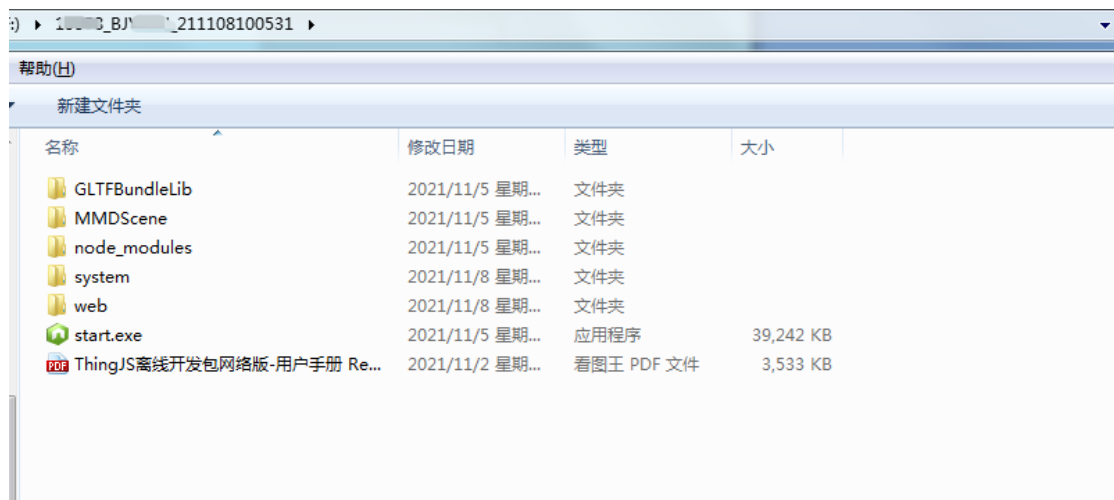


出现下图所示界面，即说明 Git 安装成功。



3.1.2 解压安装包

将获取到的 Windows 版本的“ThingJS 离线开发网络版”压缩安装包解压至需要部署的目录下。解压后目录结构如下图所示：



3.1.3 启动服务

打开“ThingJS 离线开发网络版”服务的部署目录，双击运行 start.exe 即可启动，服务默认的端口号为 9100，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

3.2.1 安装 Git

3.2.1.1 安装依赖包

安装 Linux 系统对应的依赖包，命令如下：

Centos/RedHat:


```
$ yum install curl-devel expat-devel gettext-devel \
    openssl-devel zlib-devel
```

Debian/Ubuntu :

```
$ apt-get install libcurl4-gnutls-dev libexpat1-dev gettext \
    libz-dev libssl-dev
```

3.2.1.2 下载源码包

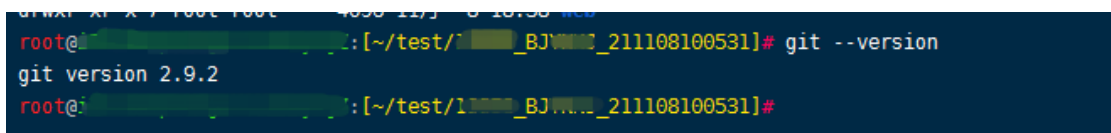
访问网址 <https://git-scm.com/download> 下载源码包。

3.2.1.3 安装

解压并安装下载的源码包，命令如下：

```
$ tar -zxf git-x.x.x.tar.gz
$ cd git-x.x.x
$ make prefix=/usr/local all
$ sudo make prefix=/usr/local install
```

安装完成，查看 Git 版本号，如下图所示，即说明 Git 安装成功。



```
root@BJ20000211108100531:~/test/1_BJ20000211108100531# git --version
git version 2.9.2
root@BJ20000211108100531:~/test/1_BJ20000211108100531#
```

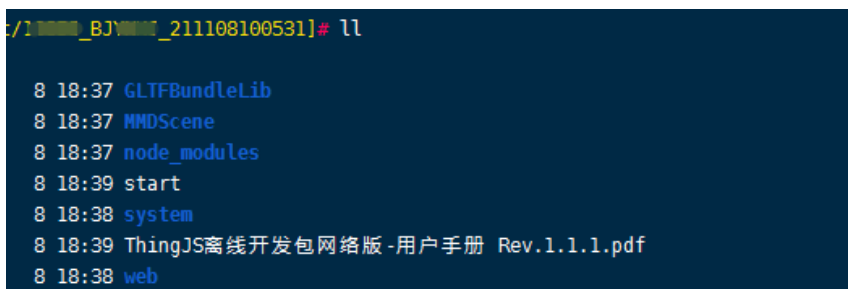
3.2.1.4 配置

成功安装 Git 后，依次执行以下命令，存储凭证。

```
$ git config --global credential.helper store
$ git config --global user.name "username"
$ git config --global user.email "email"
```

3.2.2 解压安装包

将获取到的 Linux 版本的“ThingJS 离线开发网络版”压缩安装包解压至需要部署的目录下。解压后目录结构如下图所示：



```
t/1_BJ20000211108100531# ll
8 18:37 GLTFBundleLib
8 18:37 MMDScene
8 18:37 node_modules
8 18:39 start
8 18:38 system
8 18:39 ThingJS离线开发包网络版-用户手册 Rev.1.1.1.pdf
8 18:38 web
```

3.2.3 启动服务

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

```
[~/test/1[REDACTED]_BJ[REDACTED]_108100531]# chmod u+x start
[~/test/1[REDACTED]_BJ[REDACTED]_108100531]# ll

96 11月  8 18:37 GLTFBundleLib
96 11月  8 18:37 MMDScene
96 11月  8 18:37 node_modules
14 11月  8 18:39 start
96 11月  8 18:38 system
38 11月  8 18:39 ThingJS离线开发包网络版-用户手册 Rev.1.1.1.pdf
96 11月  8 18:38 web
```

执行 `start` 启动服务，服务默认的端口号为 9100，如需修改请参考第 4.1 节的内容。



```

ThingJS Development Server v1.1.1 listening at 0.0.0.0:9100

status : unauthorized (50108)
尚未授权，请先到离线开发网络版管理后台进行授权！

离线开发网络版管理后台: http://192.168.1.20:9100/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 -napt | 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。

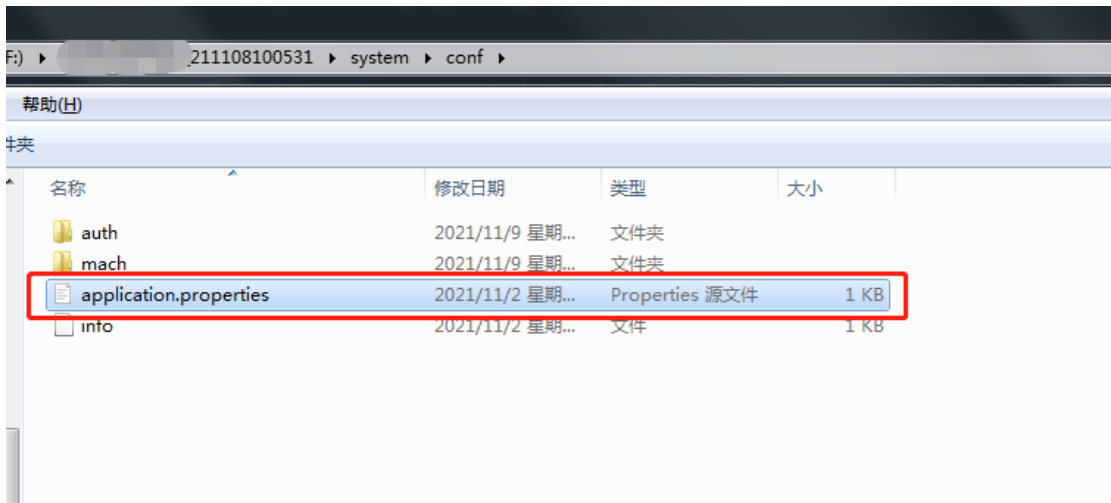
```
h22101280]# kill -9 21858
```

4 配置

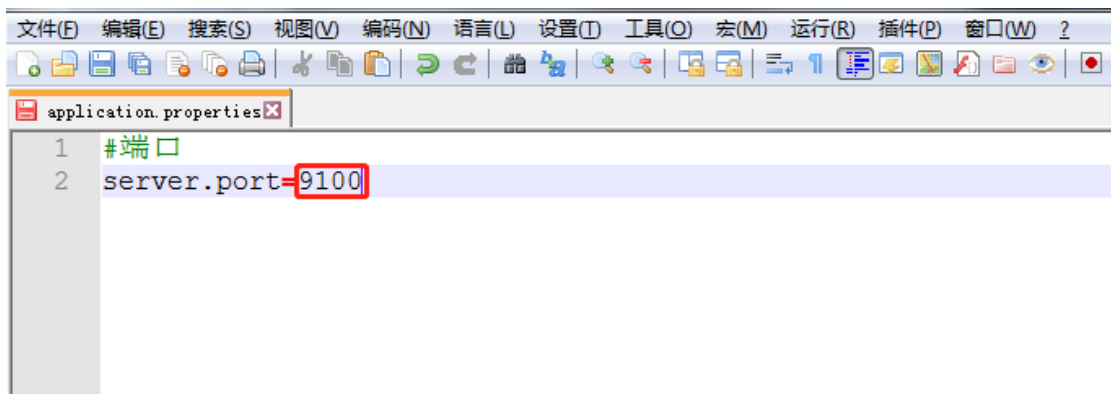
本章介绍了几种常见的“ThingJS 离线开发网络版”支持的配置方法，请勿自行更改未说明的其他文件内容，以免发生不可预期的问题。

4.1 修改服务启动端口

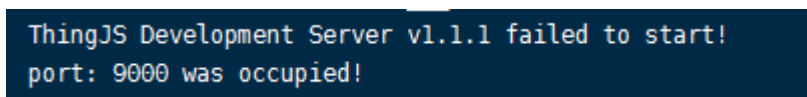
“ThingJS 离线开发网络版”服务默认端口号为 9100，可以通过配置服务目录中 system\conf 目录下的 application.properties 文件来修改端口号。



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服务，如下图所示：



4.2 修改登录密码

为确保“ThingJS 离线开发网络版”中项目的安全，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“离线开发网络版”管理界面后，如图所示，点击“修改密码”，

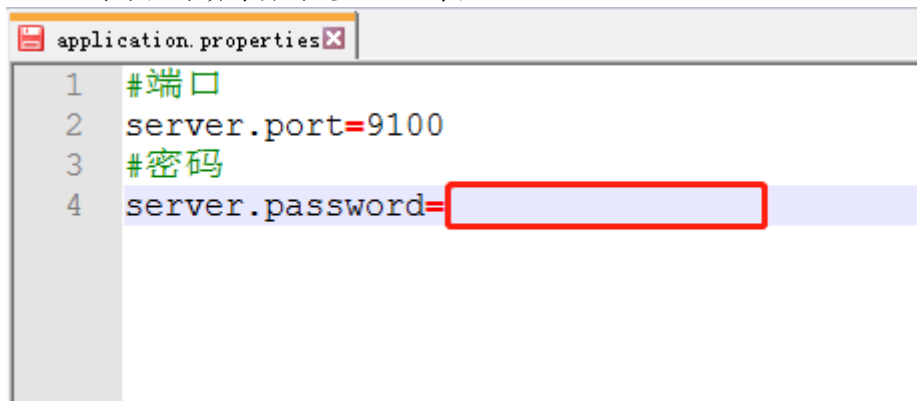


在弹出的对话框中输入需要修改的新密码，即可完成登录密码的修改。



密码将以“密文”的形式保存在 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。)



The image shows a code editor window titled "application.properties". The content of the file is as follows:

```
1 #端口
2 server.port=9100
3 #密码
4 server.password=
```

The text "server.password=" is highlighted in blue, and a red rectangular box is drawn around the empty space following the equals sign, indicating where a password should be entered.

5 授权

5.1 访问“离线开发网络版”管理界面

根据服务启动信息，访问“离线开发网络版”管理界面（<http://ip:port/admin>）。

```
ThingJS Development Server v1.1.1 listening at 0.0.0.0 9000

status : authorized
已授权，可到离线开发网络版管理后台进行后续离线开发操作！

离线开发网络版管理后台：http://172.16.2.22:9000/admin
```

访问“离线开发网络版”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”（为确保项目的安全，请在首次登录后尽快修改密码，参考第4.2节内容），登录界面如下图所示：



登录成功后即可进入到“离线开发网络版”管理界面，如下图所示：



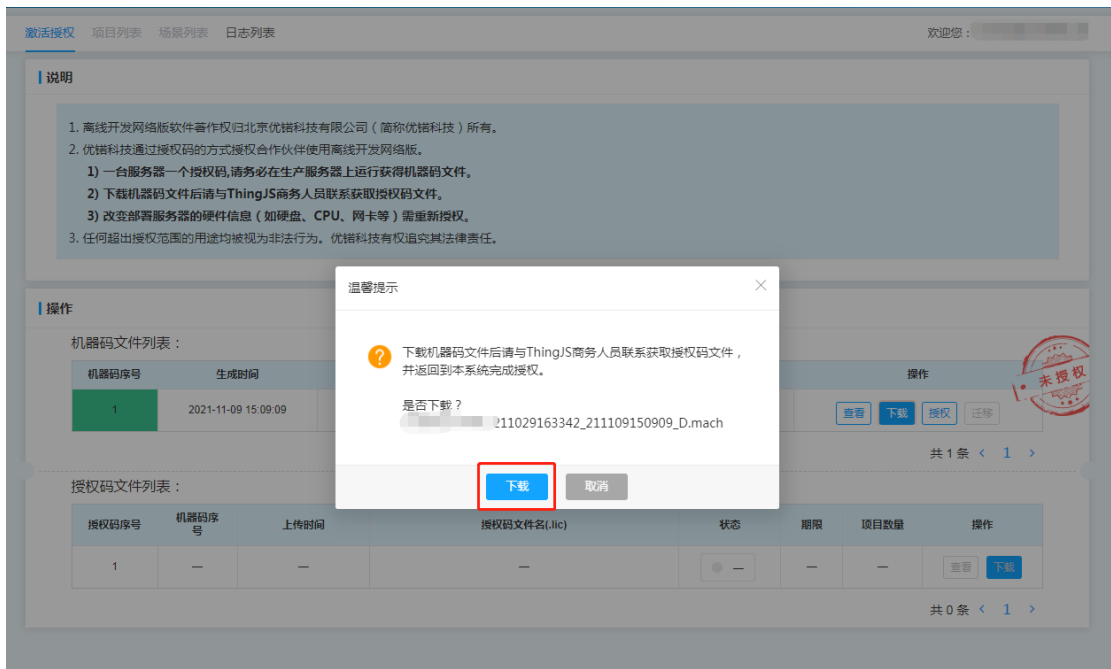
5.2 获取授权

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 申请授权码文件

将 5.2.1 节获取的机器码文件发送给 ThingJS 商务人员，获取对应的授权码文件。

5.2.3 完成授权

返回到“离线开发网络版”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，即可完成授权。



ThingJS | 离线开发网络版 v1.1.1
admin

系统概况 | 角色列表 | 场景列表 | 日志列表

说明

- 离线开发网络版软件著作权归北京优炫科技有限公司（简称优炫科技）所有。
- 优炫科技通过授权的方式授权合作伙伴使用离线开发网络版。
 - 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。
 - 下载机器码文件后请与ThingJS商务人员联系获取授权码文件。
 - 改变部署服务器的硬件信息（如硬盘、CPU、网卡等）需重新授权。
- 任何超出授权范围的使用均被视为非法行为，优炫科技有权追究法律责任。

操作

机器码文件列表：

机器码序号	生成时间	机器码文件名(.mach)	状态	操作
1	2021-11-09 15:09:00	11029163342_211109150909_D	有效	更新 下载 授权 注销

共 1 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	期限	项目数量	操作
1	1	2021-11-09 15:13:04	11029163342_211109151255_D	有效			更新 下载

共 1 条 < 1 >

北京优炫科技有限公司 | 官网: <https://www.thingjs.com> | 电话: 400-666-9832 | 版本号: v1.1.1

6 开发

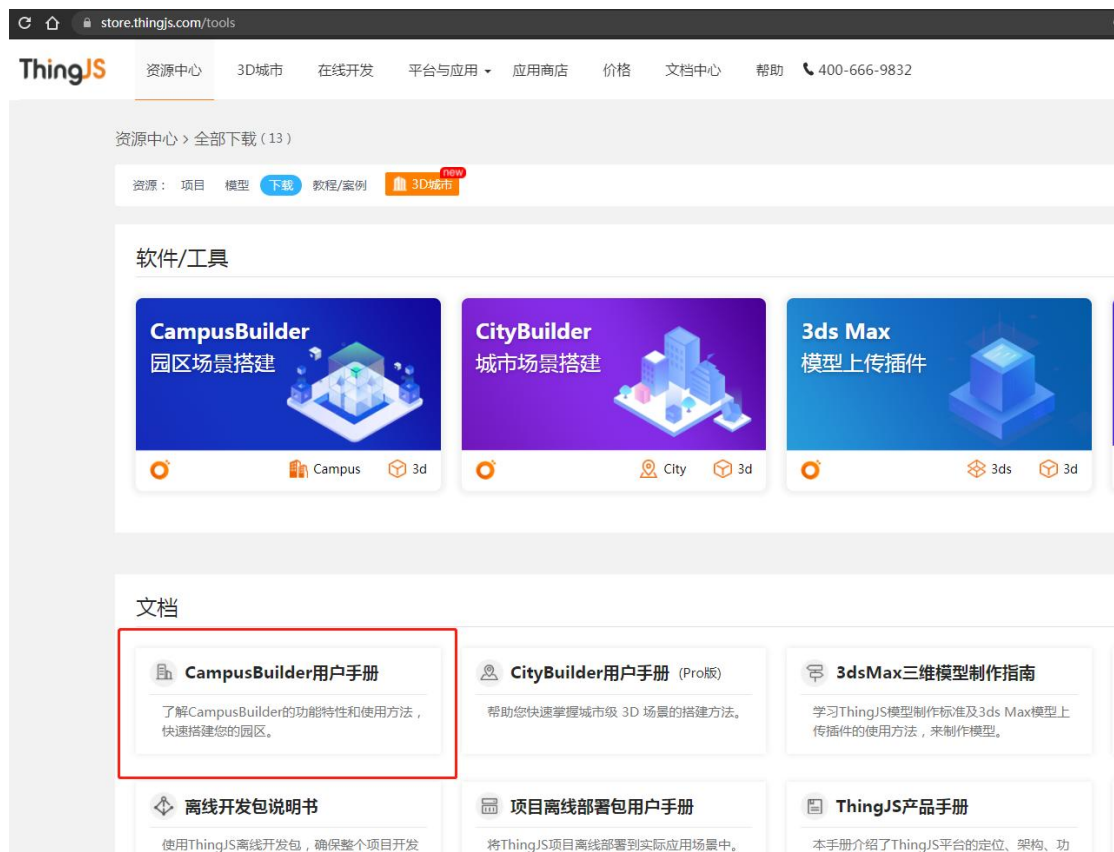
本章将详细的介绍如何通过“ThingJS 离线开发网络版”进行项目开发，包括搭建场景、开发应用，到最后发布项目。

6.1 搭建场景

6.1.1 安装“CampusBuilder”

CampusBuilder 在园区级 3D 可视化场景的搭建方面，功能强大，不仅可以搭建园区场景，在建筑楼层和室内搭建方面也表现卓越。

可在网站 <https://store.thingjs.com/tools> 中下载最新版本，并查看“CampusBuilder 用户手册”进行后续的安装使用。



6.1.2 搭建场景

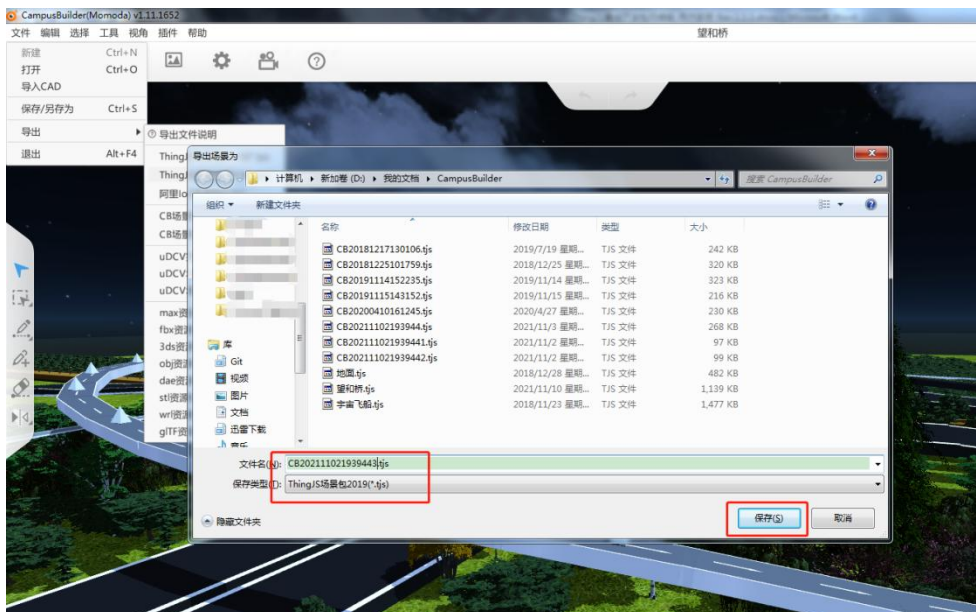
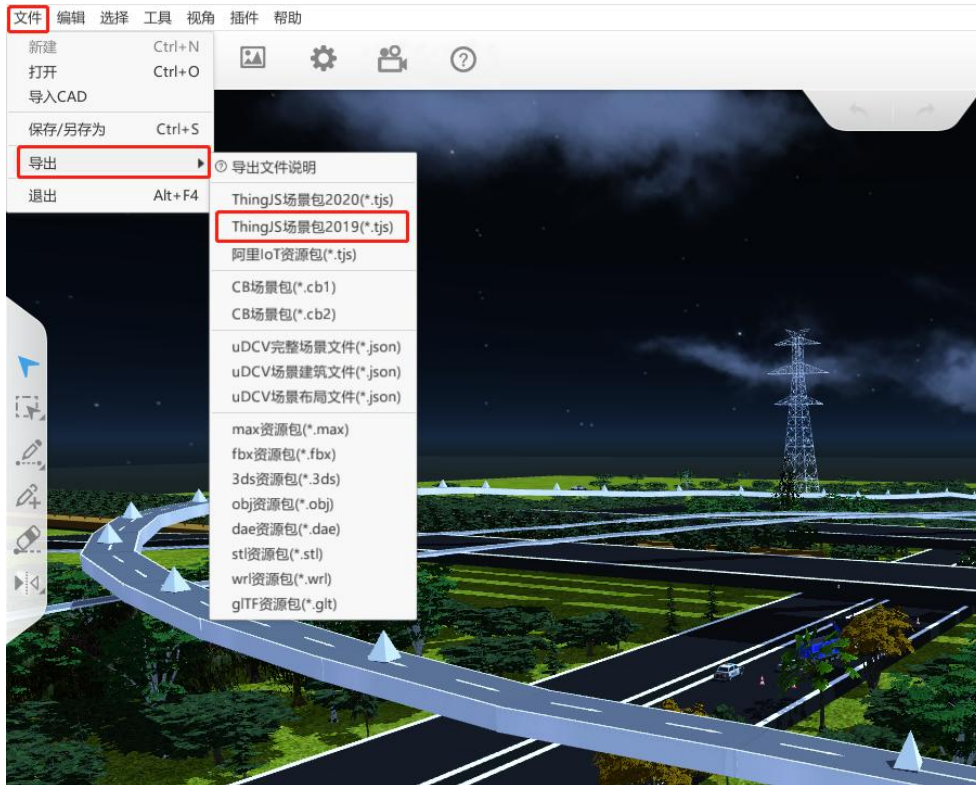
可在网站 <https://store.thingjs.com/tools> 中查看“CampusBuilder 用户手册”学习搭建场景。

6.1.3 上传场景

为后续使用场景进行应用开发，需将“CampusBuilder”中搭建好的场景上传到“ThingJS 离线开发网络版”中。

6.1.3.1 导出场景

在“CampusBuilder”中搭建好场景后，点击“文件-导出-ThingJS 场景包 2019(*.tjs)”，如下图所示。



6.1.3.2 上传场景

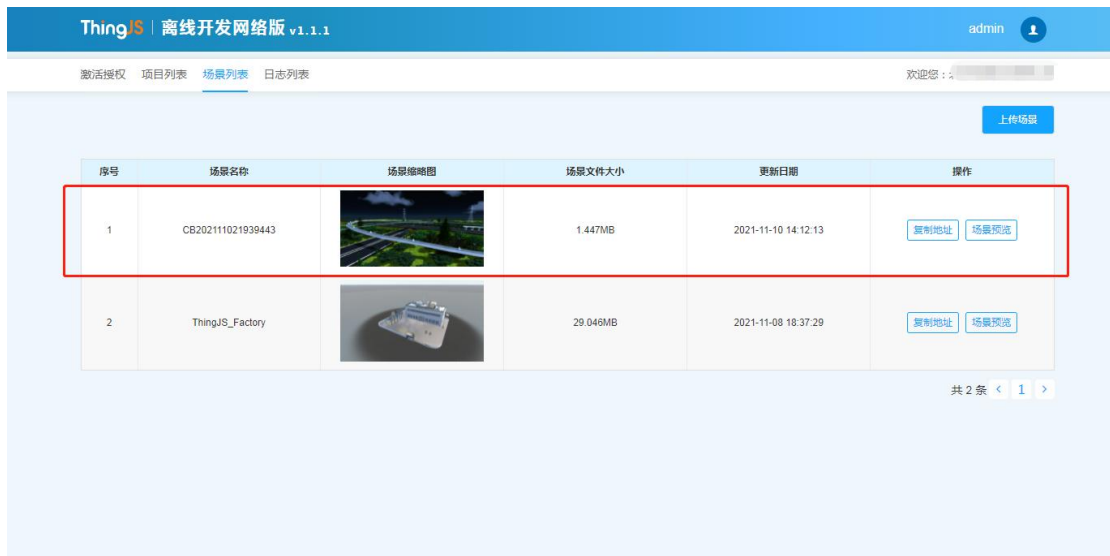
访问“ThingJS 离线开发网络版”管理界面，切换到“场景列表”页签，如下图所示：



点击“上传场景”按钮，



上传 6.1.3.1 节中导出的“ThingJS 场景包 2019(*.tjs)”，上传成功后，场景列表如下图所示：



上传成功后，可对该场景进行“复制地址”和“场景预览”操作。

6.2 应用开发

6.2.1 安装 Git

“ThingJS 离线开发网络版”基于 Git 对开发项目进行管理，因此需要在进行开发的客户端环境中安装 Git。

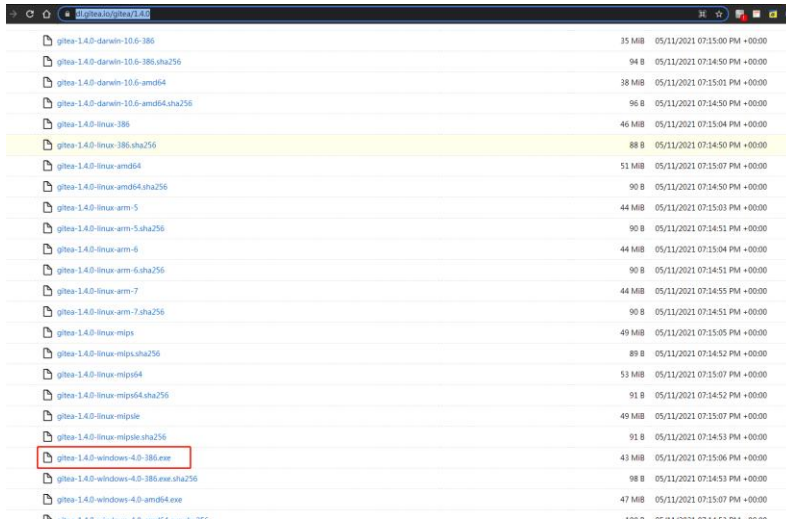
详细的安装步骤可参考第三章中关于 Git 安装的内容。

6.2.2 搭建 Git 服务器

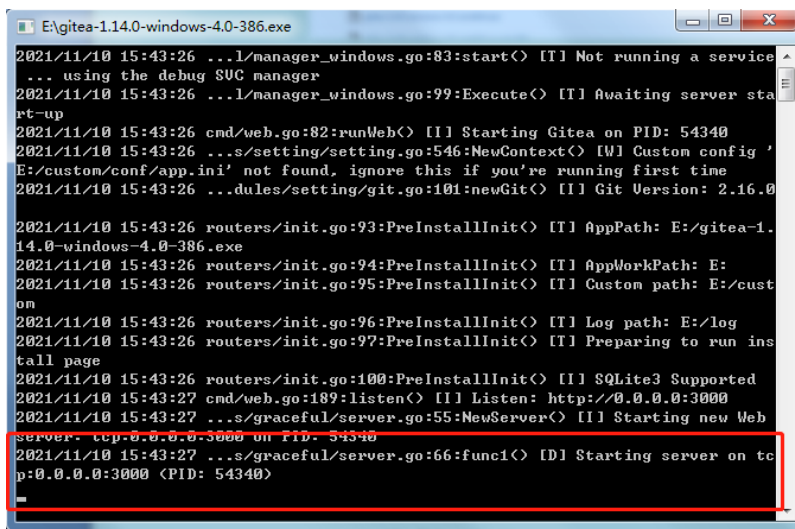
互联网上有许多免费托管开源代码的远程仓库，例如 GitHub，如果项目需要保密，也可以自行搭建 Git 服务器用于代码托管。

Git 服务器的程序有很多，比如 GoGs、Gitea、Gitblit、Gitlab 等等，都是较为成熟的 Git 服务器程序，这里仅对 Gitea 的安装进行简单的介绍，如需要搭建其它 Git 服务器，可自行查阅相关资料进行安装搭建。

访问 <https://dl.gitea.io/gitea/1.4.0> 下载对应系统的安装包，这里示例安装 Windows 版本，如下图所示，



下载完成，双击运行安装程序。



根据提示，访问服务（默认端口号为 3000），如下图所示，

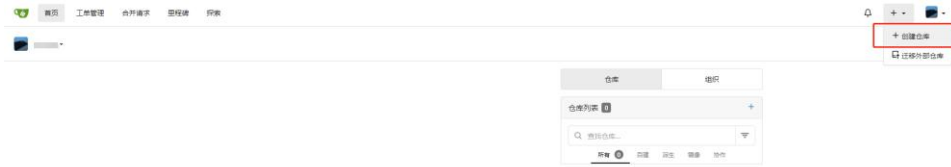


根据页面内容填写初始配置，填写完成后，点击“立即安装”即可。

更多关于 Gitea 的使用说明，请查阅文档 <https://docs.gitea.io/zh-cn/>。

6.2.3 创建仓库

以 Gitea 为例，如下图所示，点击“创建仓库”，



按需填写仓库配置内容，

创建仓库

仓库包含所有项目文件，包括修订历史。已经在别处有了吗？ [迁移代码库](#)

所有者 * uinnova

由于最大存储库限制，一些组织可能不会显示在下拉列表中。

仓库名称 * HelloWorld

好的存储库名称使用简短、深刻和独特的关键字。

可见性 符合仓库为私有
只有组织所有人或拥有权利的组织成员才能看到。

仓库描述 输入简要描述 (可选)

模板 选择模板

工单标签 选择一个工单标签集

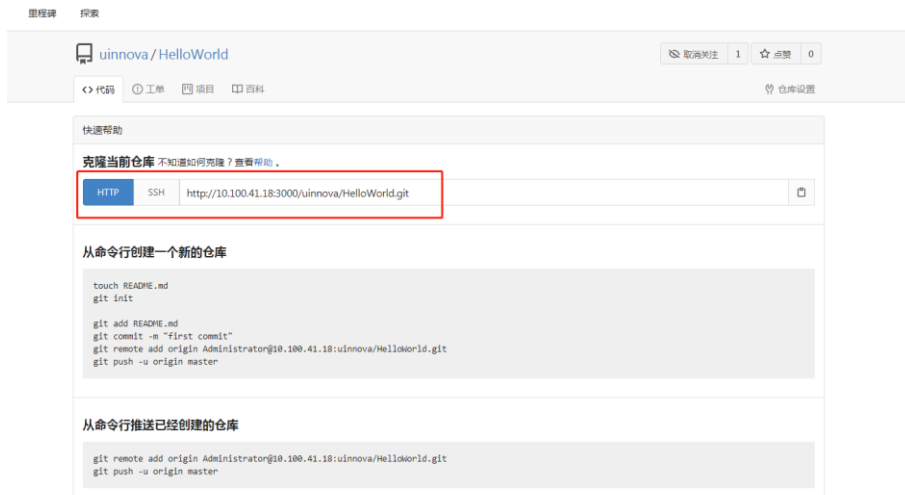
.gitignore 选择 .gitignore 模板。

从常见语言的模板列表中选择忽略跟踪的文件。默认情况下，由开发或构建工具生成的特殊文件都包含在 .gitignore 中。

授权许可 选择授权许可文件。

许可证指明了其他人可以或不可以用你的代码做什么。 不确定哪一个适合你的话

成功创建仓库后，即可得到一个仓库地址，如下图所示：



6.2.4 安装 Visual Studio Code

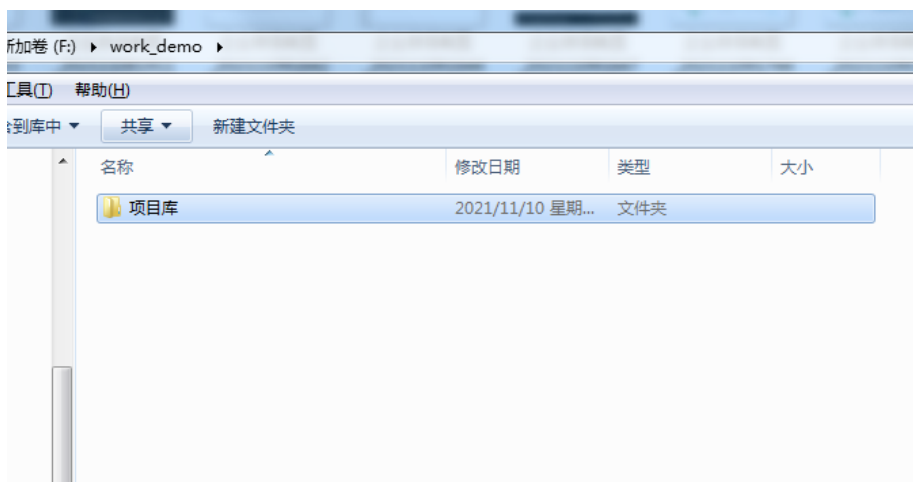
Visual Studio Code 是一款针对于编写现代 Web 和云应用的跨平台源代码编辑器，推荐使用 Visual Studio Code 进行项目开发。

最新版本可在 <https://code.visualstudio.com/> 中下载。

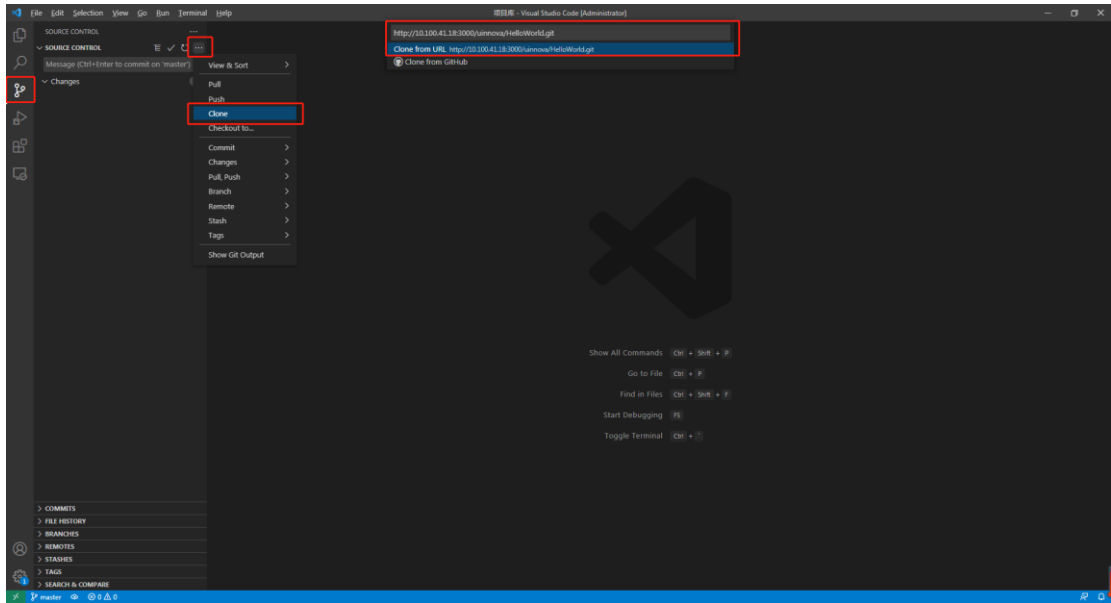
6.2.5 本地开发

6.2.5.1 克隆仓库

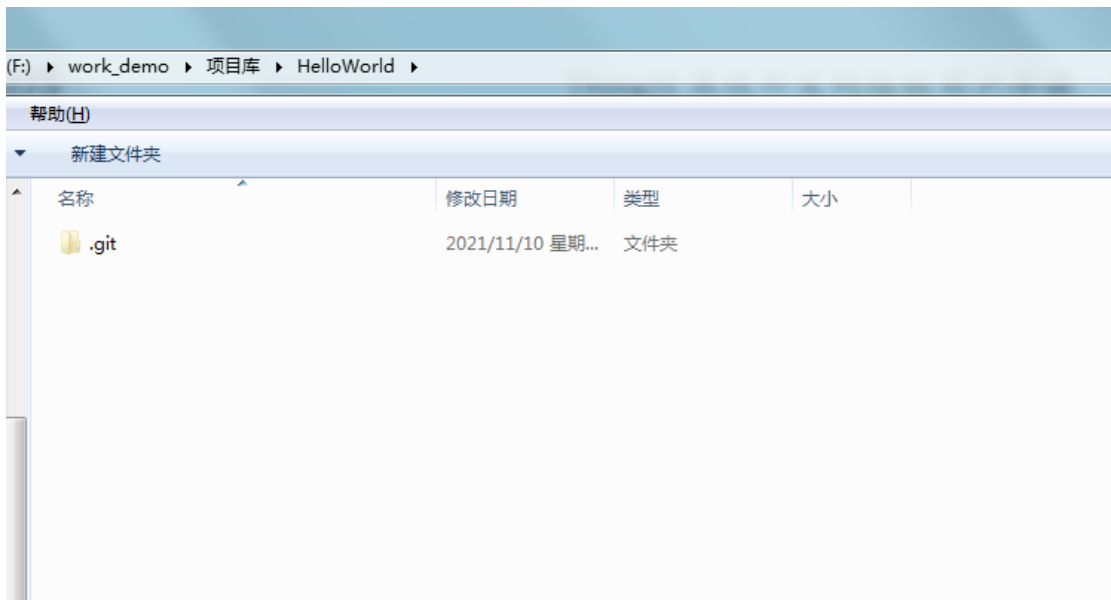
在本地开发环境中的工作目录下，创建一个名为“项目库”的文件夹，如下图所示：



使用 Visual Studio Code 打开项目库文件夹，将 6.2.3 节中得到的仓库地址，克隆 (clone) 到项目库目录下 (如有其它 Git 版本控制系统，例如 TortoiseGit，也可直接进行仓库克隆操作，本处以 Visual Studio Code 进行示例)，如下图所示：



仓库克隆成功后，目录结构如下图所示：

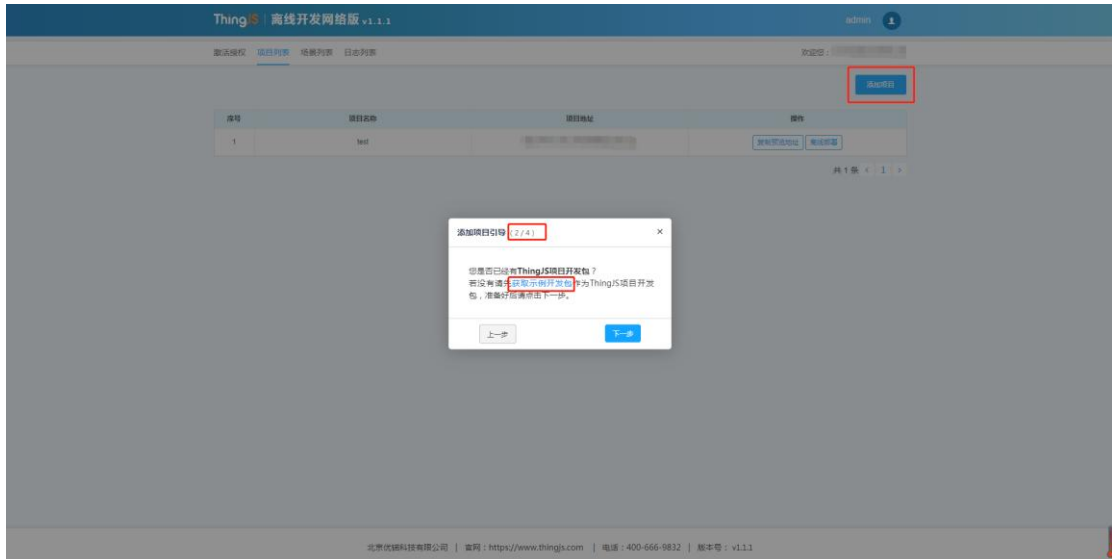


6.2.5.2 应用开发

进行应用开发时，可通过“ThingJS 离线开发网络版”提供的“示例开发包”快速创建项目，也可以从“ThingJS 在线开发平台”获取一个已有项目的“离线开发包”继续进行开发。

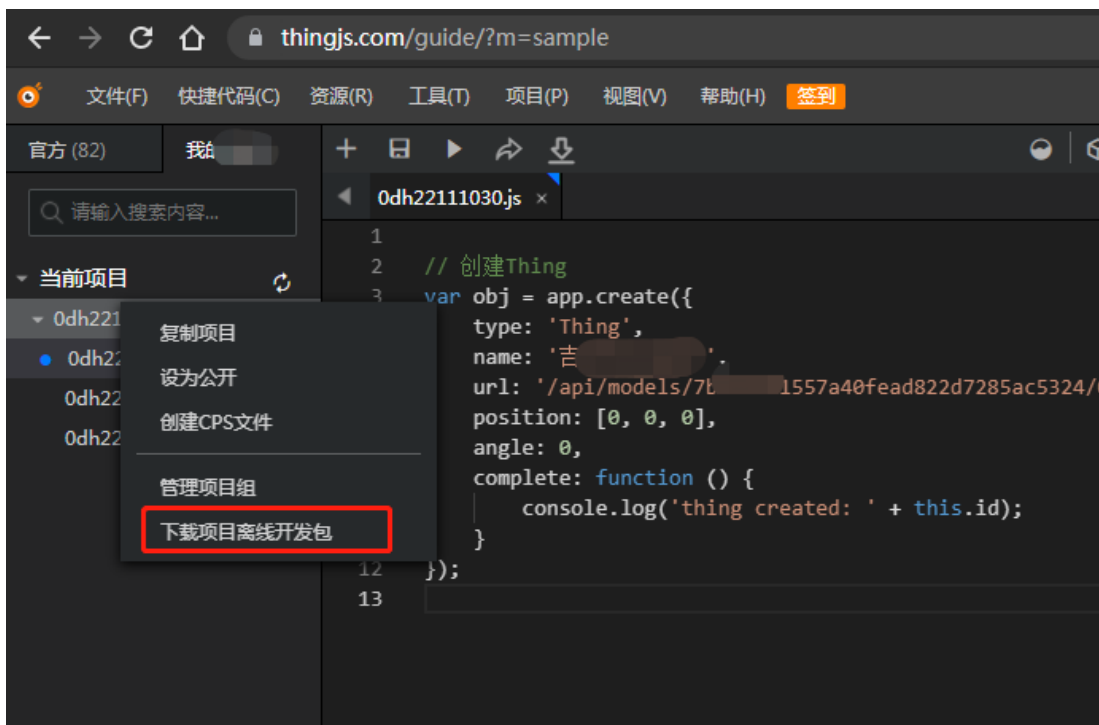
(1) 获取“示例开发包”

访问“ThingJS 离线开发网络版”，切换至“项目列表”页签，点击“添加项目”按钮，在“添加项目引导”中获取“示例开发包”，如下图所示：

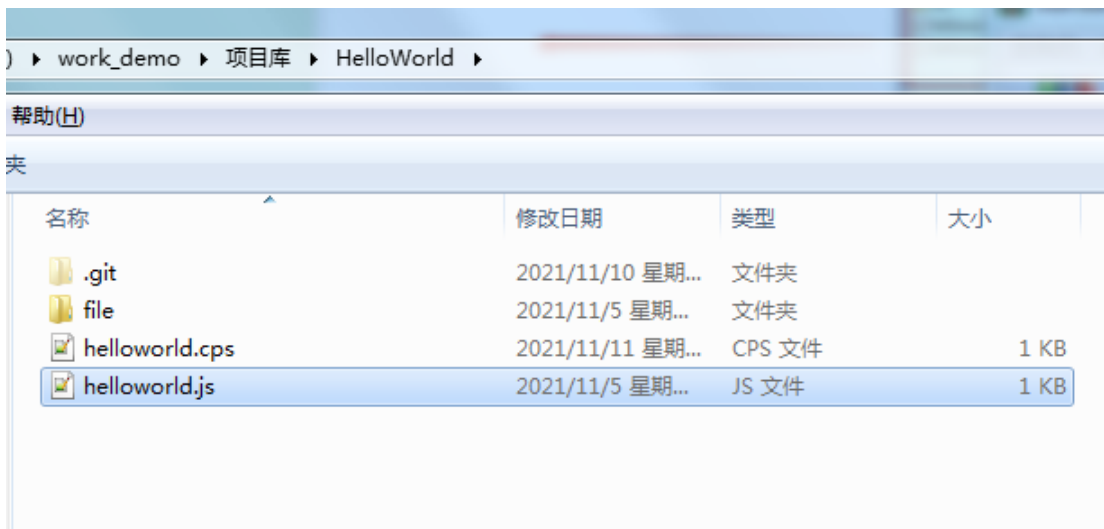
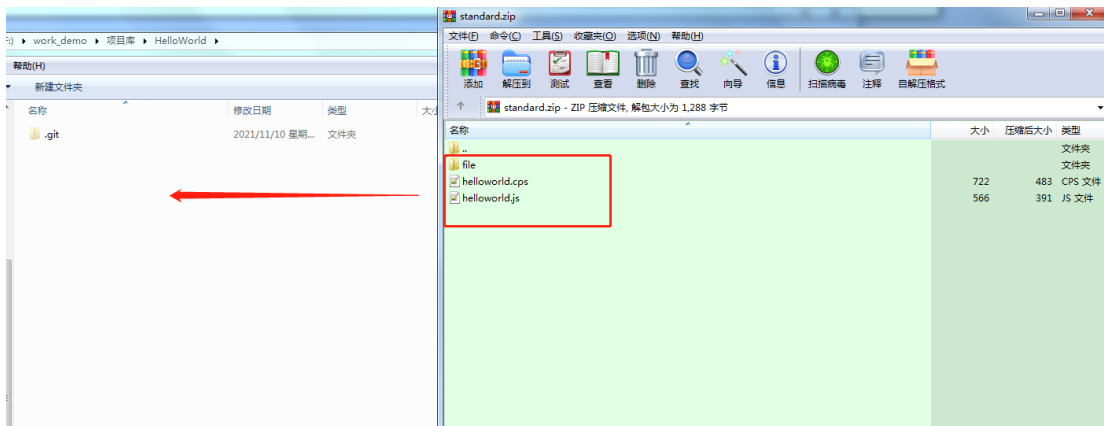


(2) 获取“离线开发包”

在“ThingJS 在线开发平台”中，打开一个已有项目，右键该项目，点击“下载项目离线开发包”，如下图所示：



将获取到的“开发包”中的文件移动到 6.2.5.1 节中克隆好仓库的目录下，此处使用“示例开发包”，如下图所示：



可以在此“开发包”基础上继续进行开发。

6.2.5.3 引用资源

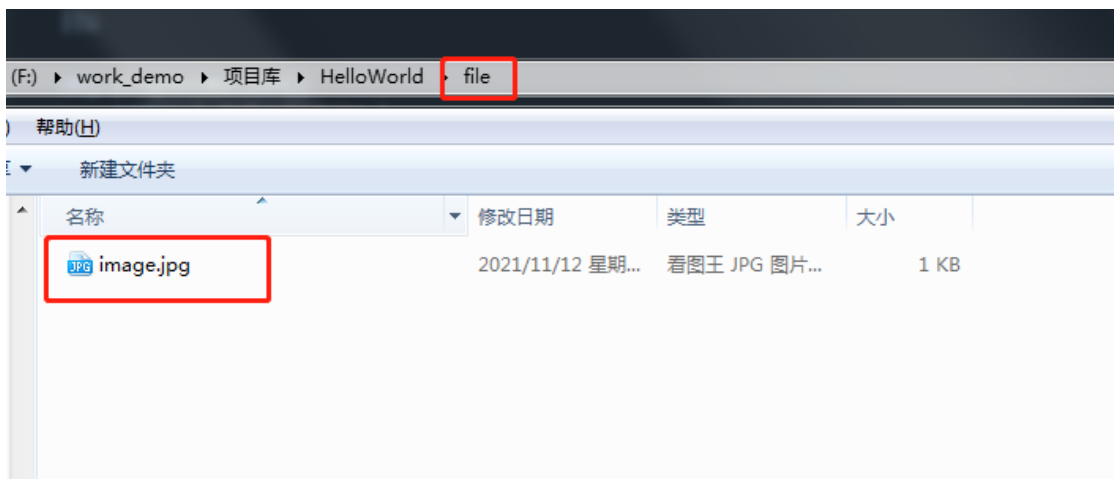
- (1) 场景地址为第 6.1.3.2 节中上传场景后所获取到的场景地址；
- (2) 如果想用代码创建模型，可以去 CampusBuilder 客户端右侧模型列表找到要加载的模型 id，例如模型 id 为“7bfb3321557a40fead822d7285ac5324”，则该模型地址为：“/api/models/7bfb3321557a40fead822d7285ac5324/0/gltf/”，其中的 7bfb3321557a40fead822d7285ac5324 为模型 id，如下图所示。

```
var app = new THING.App({
  // 引用场景
  url: '/api/scene/20210616095248956300069', // 场景地址
  skyBox: 'BlueSky', // 天空盒
  resourceLibraryUrl: './'
});

app.on('load', function () {
  // 创建模型
  let obj = app.create({
    type: 'Thing',
    name: '宇航员',
    url: '/api/models/7bfb3321557a40fead822d7285ac5324/0/gltf/',
    position: [0, 0, 0],
    angle: 45
  });

  obj.playAnimation({
    name: '_defaultAnim_',
    loopType: THING.LoopType.Repeat,
  });
});
```


- (3) 地图离线开发组件引用地址为“/source/uearth.min.js”。
- (4) 项目中其他的文件资源引用地址需使用相对地址进行加载，例如：需要引用项目中/file 目录下的文件 image.jpg，目录结构如下图所示：

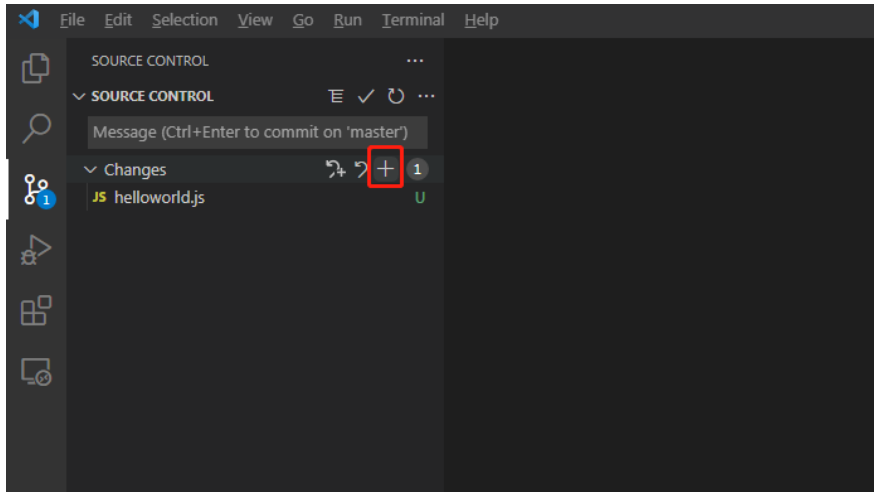


则引用地址为“./file/image.jpg”，如下图所示：

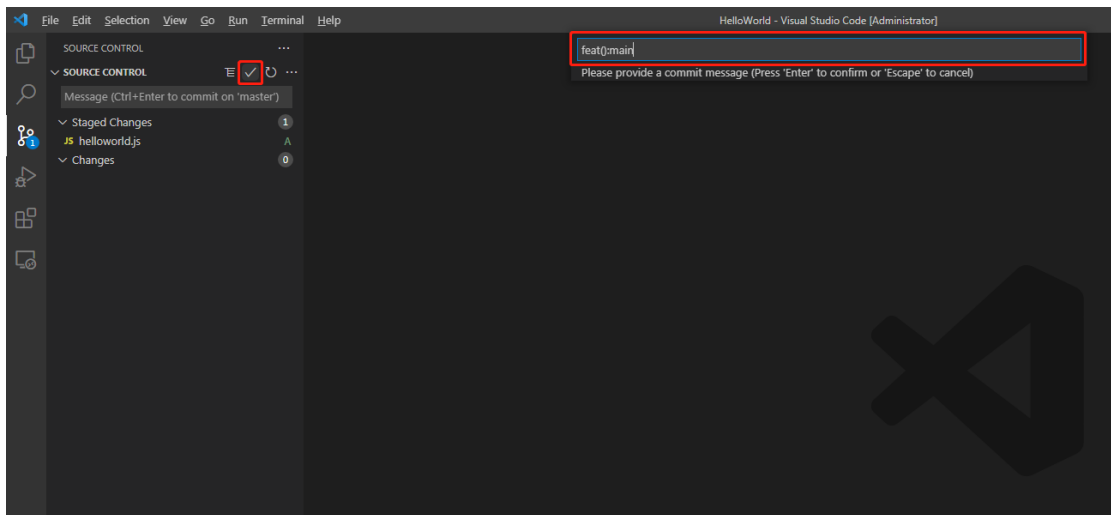
```
23
24 new THING.widget.Button('设置背景图片', function () {
25   app.background = './file/image.jpg';
26 })
27
```

6.2.5.4 推送项目

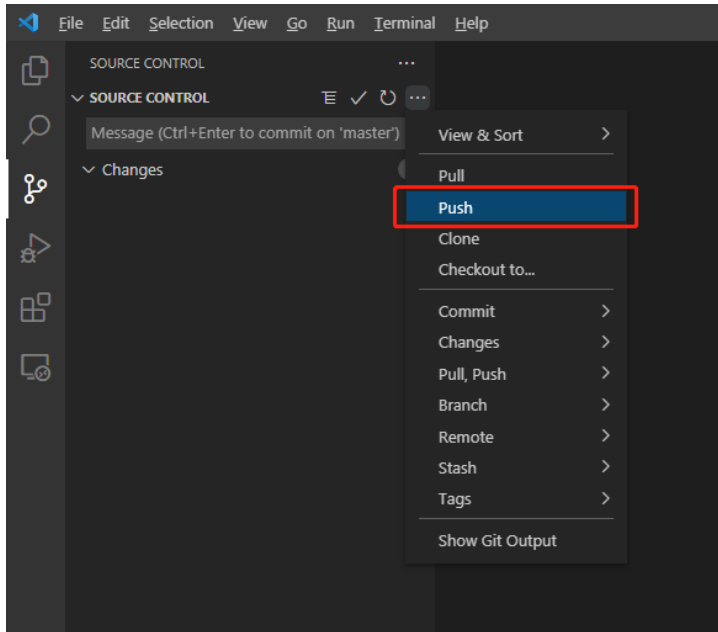
将项目添加到克隆好的仓库目录中后，Visual Studio Code 的 Git 版本控制面板中将展示新添加的文件，点击  将文件添加到 Git 中，如下图所示：



添加后点击 ，将更改内容提交（commit）到 Git 中，如下图所示：



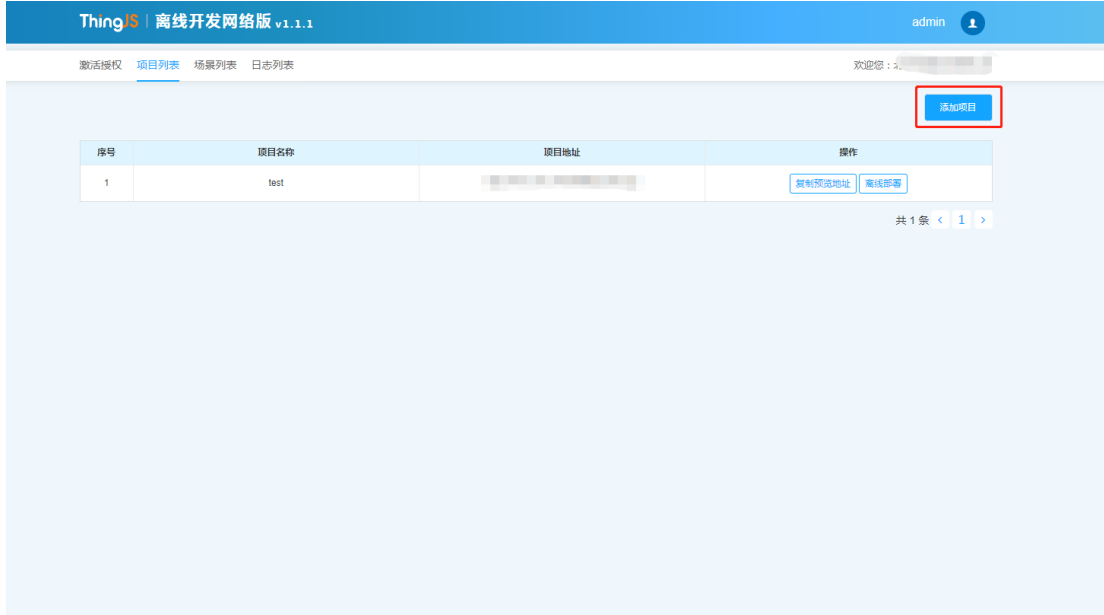
提交完成后，选择“推送”，如下图所示：



注：为能预览到最新的项目，每次都需将修改内容推送到仓库中。

6.2.6 添加项目

项目推送到仓库中后，访问“ThingJS 离线开发网络版”管理界面，切换到“项目列表”页签，点击“添加项目”按钮，如下图所示：



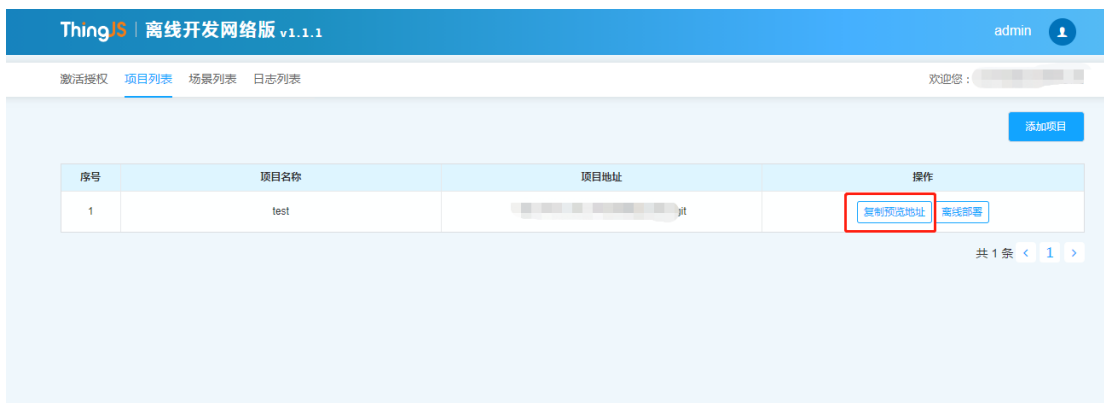
根据“添加项目引导”提示，填写第 6.2.5.1 节中项目的“仓库地址”，如下图所示：



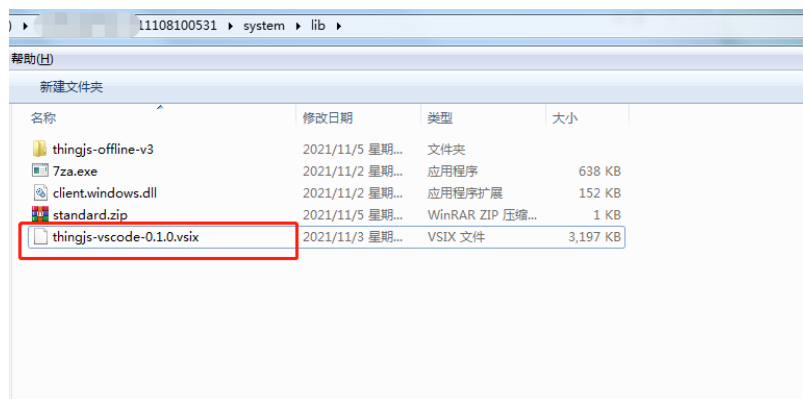
点击“创建”按钮，即可完成项目的添加，添加成功后，可对项目进行预览和离线部署操作。

6.2.7 预览项目

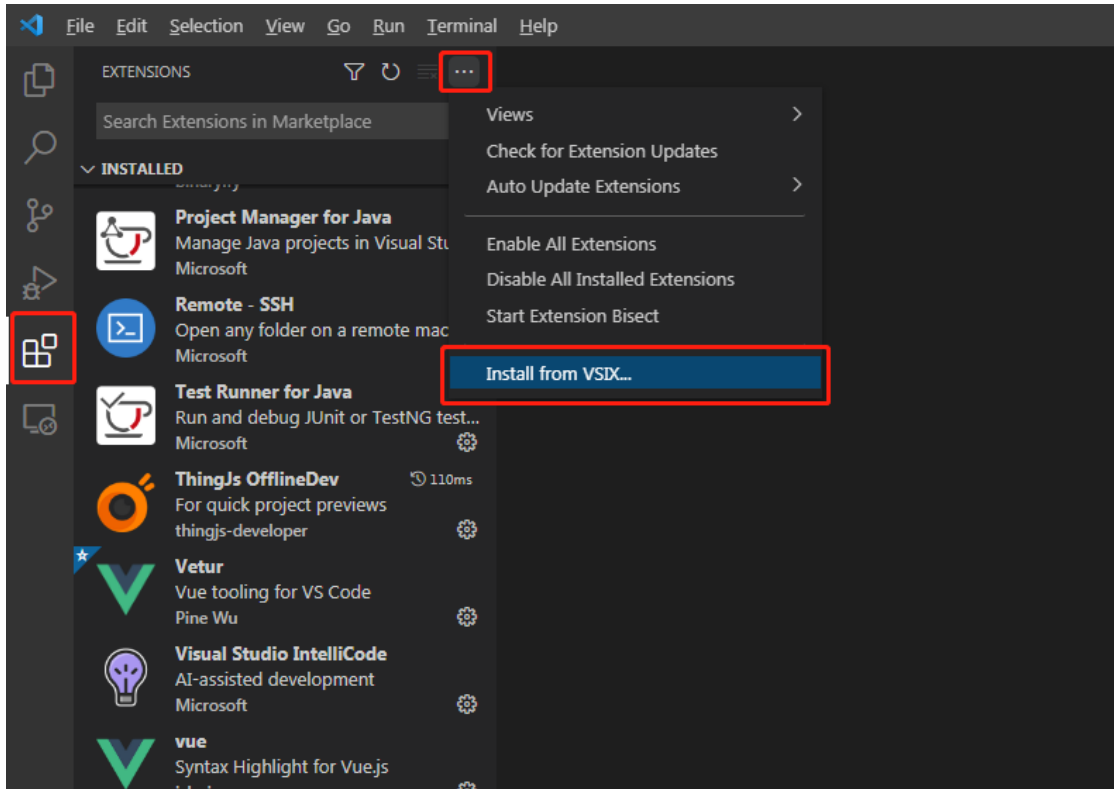
(1) 直接“复制预览地址”进行项目预览，如下图所示：



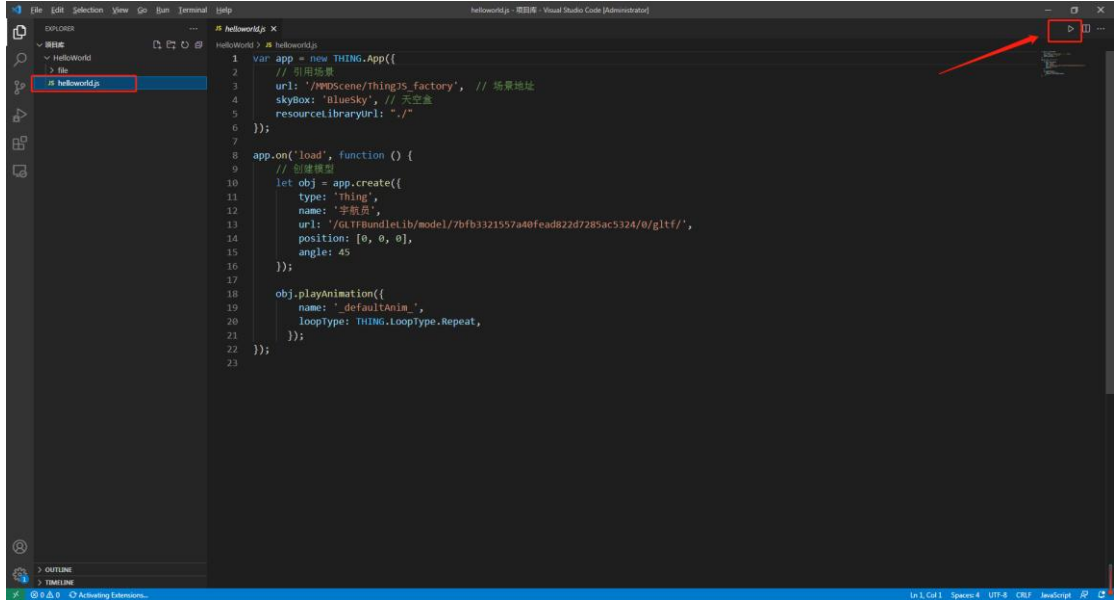
(2) 在开发环境中，为 Visual Studio Code 添加插件“thingjs-vscode-0.1.0.vsix”（注：Visual Studio Code 版本需在 1.50.0 以上），该插件可在“服务安装包”中“system/lib”目录下获取，



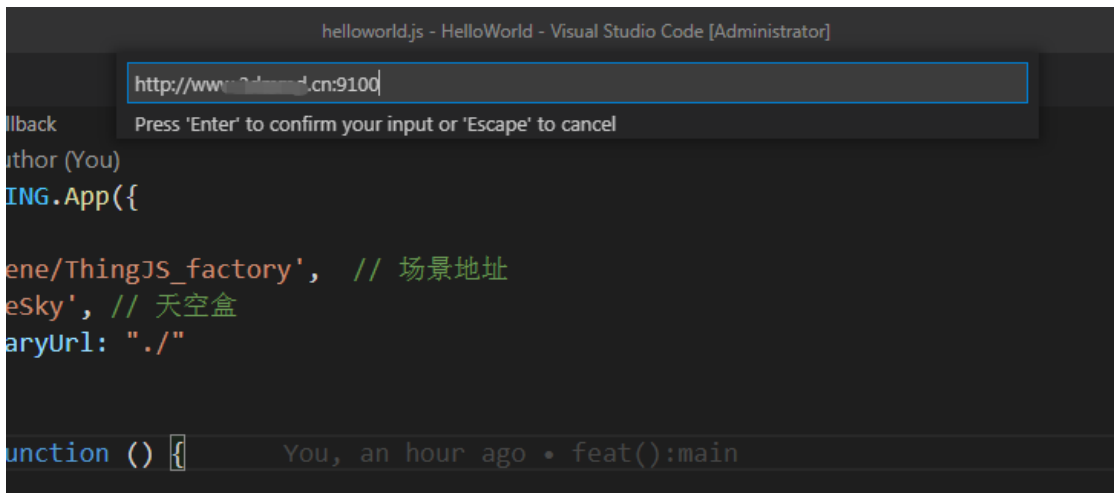
在 Visual Studio Code 中添加插件，如下图所示：



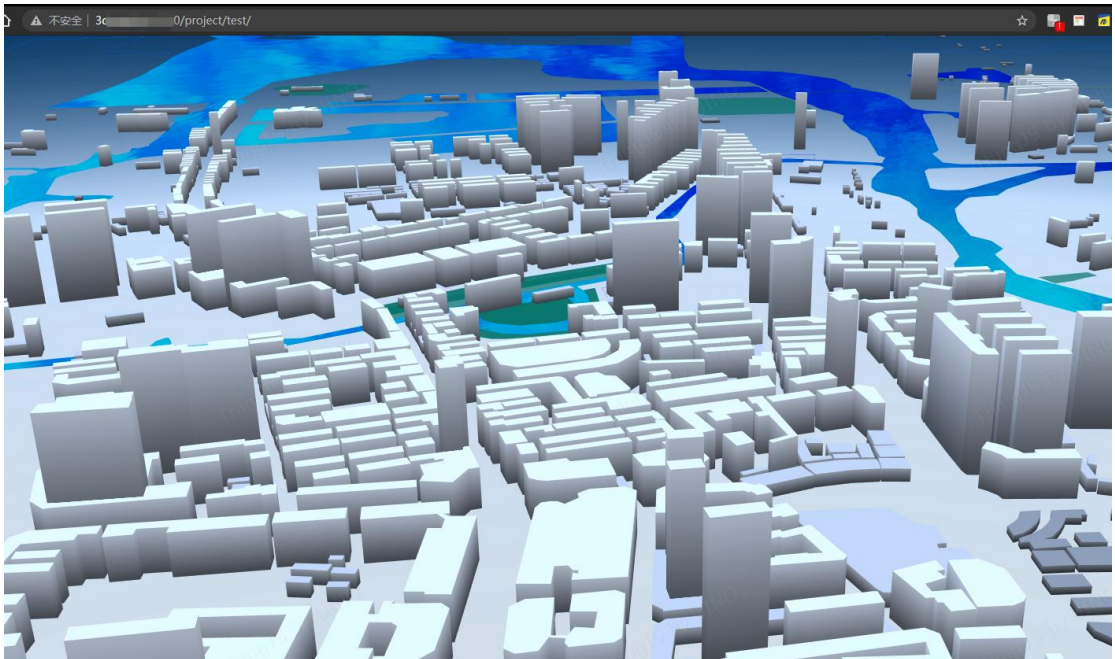
添加成功后，返回到项目中，点击右上方的“项目预览”按钮，如下图所示：



在弹出的提示框中填写“ThingJS 离线开发网络版”服务地址，



确定后即可默认跳转到浏览器进行“项目预览”。

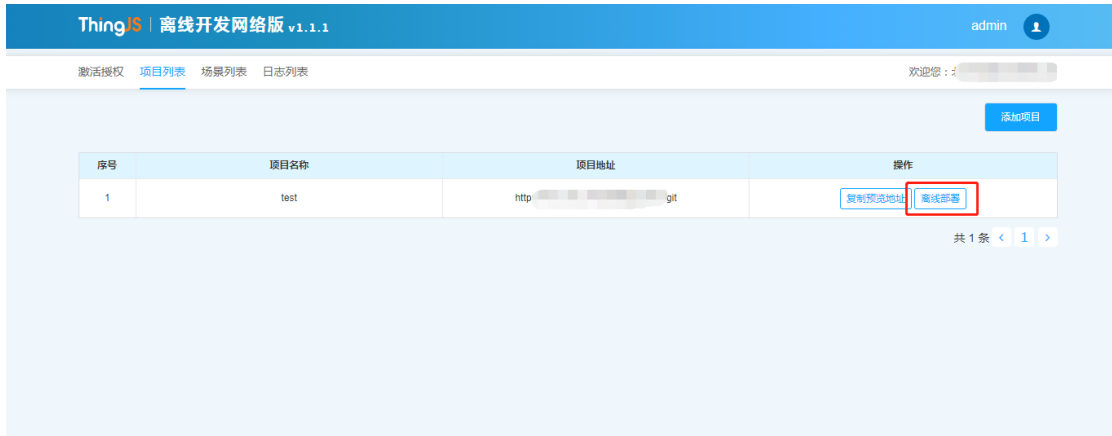


每次修改推送成功后，刷新预览链接即可看到最新的项目内容。

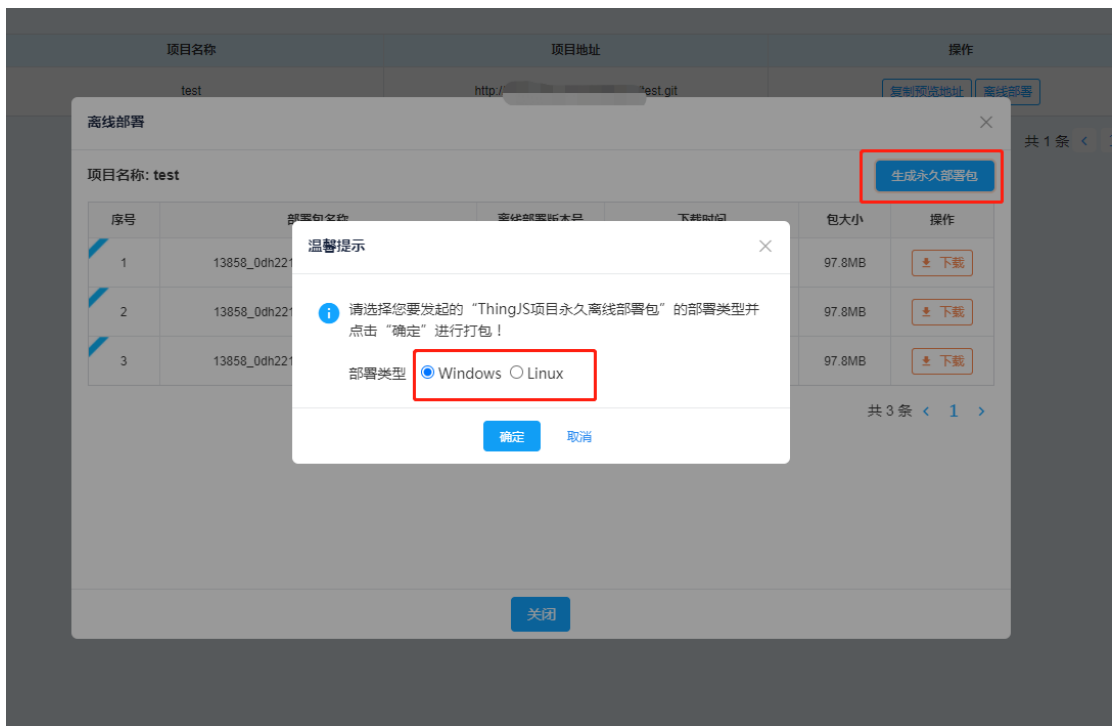
6.3 离线部署

项目开发完成后，可对项目进行离线部署。

访问“ThingJS 离线开发网络版”管理界面，切换到“项目列表”页签，对要发布的项目点击“离线部署”按钮，如下图所示：



点击“生成永久部署包”按钮，并选择“部署类型”，等待部署包生成即可。



关于离线部署包的使用，请到 <https://store.thingjs.com/tools> 查看“项目离线部署包用户手册”进行后续操作。

7 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“ThingJS 离线开发网络版”进行环境变更，请按本章节内容迁移。

7.1 获取新机器码文件

将“ThingJS 离线开发网络版”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“ThingJS 离线开发网络版”管理界面，下载当前有效的新的“机器码文件”至本地。

7.2 获取迁移码文件

返回到原已部署的“ThingJS 离线开发网络版”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。）

ThingJS | 离线开发网络版 v1.1.1 admin

激活授权 项目列表 场景列表 日志列表 欢迎您: [用户名]

说明

- 离线开发网络版软件著作权归北京优锴科技有限公司（简称优锴科技）所有。
- 优锴科技通过授权码的方式授权合作伙伴使用离线开发网络版。
 - 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。
 - 下载机器码文件后请与ThingJS商务人员联系获取授权码文件。
 - 改变部署服务器的硬件信息（如硬盘、CPU、网卡等）需重新授权。
- 任何超出授权范围的用途均被视为非法行为。优锴科技有权追究其法律责任。

操作

机器码文件列表:

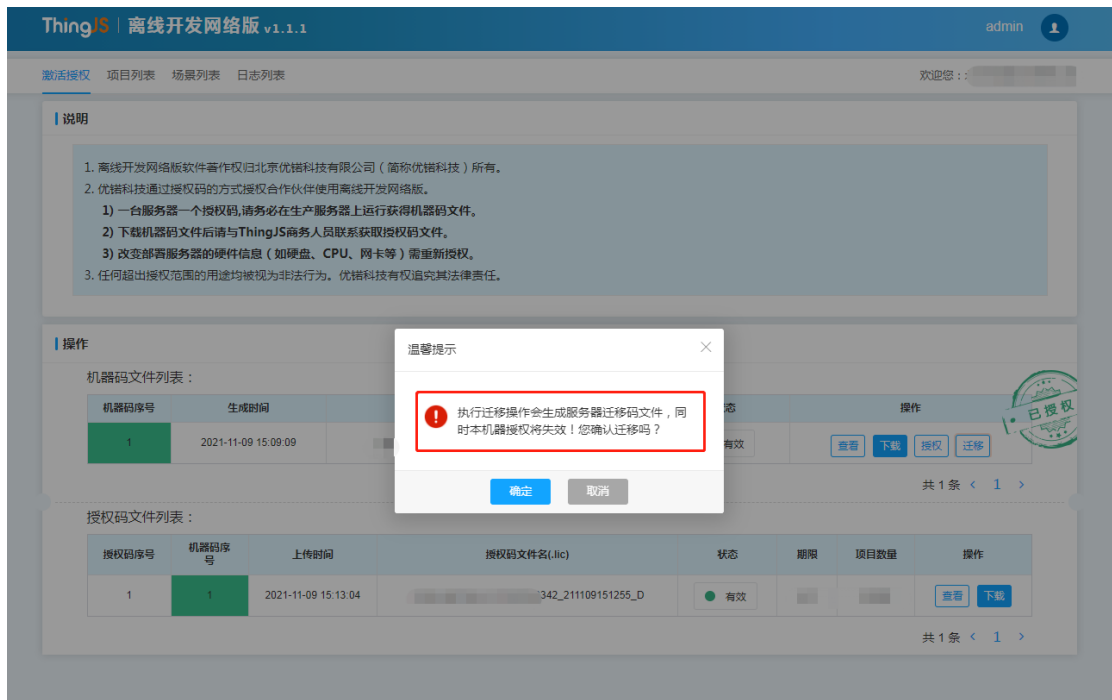
机器码序号	生成时间	机器码文件名(.mach)	状态	操作
1	2021-11-09 15:09:09	[模糊]_211029163342_211109150809_D	有效	查看 下载 授权 迁移

共 1 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	期限	项目数量	操作
1	1	2021-11-09 15:13:04	[模糊]_211029163342_211109151255_D	有效			查看 下载

共 1 条 < 1 >



将生成的“迁移码文件”下载至本地。

7.3 获取新授权码文件

将 7.1 节获取的“新机器码文件”和“迁移码文件”发送给 ThingJS 商务人员，获取“新授权码文件”。

7.4 完成新服务器的授权

访问部署在新服务器环境中的“ThingJS 离线开发网络版”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。

8 常见问题 (FAQ)

8.1 访问 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。

8.2 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，或为服务配置其他可使用的端口号。

8.3 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。

9 修订历史

版本	日期	更新内容描述
Rev. 1.1.1	2021/11/09	ThingJS 离线开发网络版功能说明。
Rev. 1.2.1	2021/11/26	更新场景、模型引用路径。
Rev. 1.2.2	2022/01/08	优化离线开发网络版服务接口。