

# 如何优雅的使用Github Action服务来将Hexo部署到Github Pages



## 参考文章

[Bilibili视频教程-9分钟零成本搭建自动化部署个人博客\(Hexo + Github Action + Page\)](#)

[Hexo官方文档](#)

[利用 GitHub Action 自动部署 Hexo 博客](#)

[Hexo主题-Icarus快速上手](#)

## 前提条件

当前PC环境中Node和Git。版本可以参考Hexo文档。

文章中出现的yourusername为Github用户名，your-repo为仓库名。

## 1. 初始化Hexo

安装脚手架，初始化hexo，这会新建blog文件夹，进入后安装依赖。

```
1 npm install -g hexo-cli
2 hexo init blog
3 cd blog
4 npm install
```

## 2. 初始化仓库

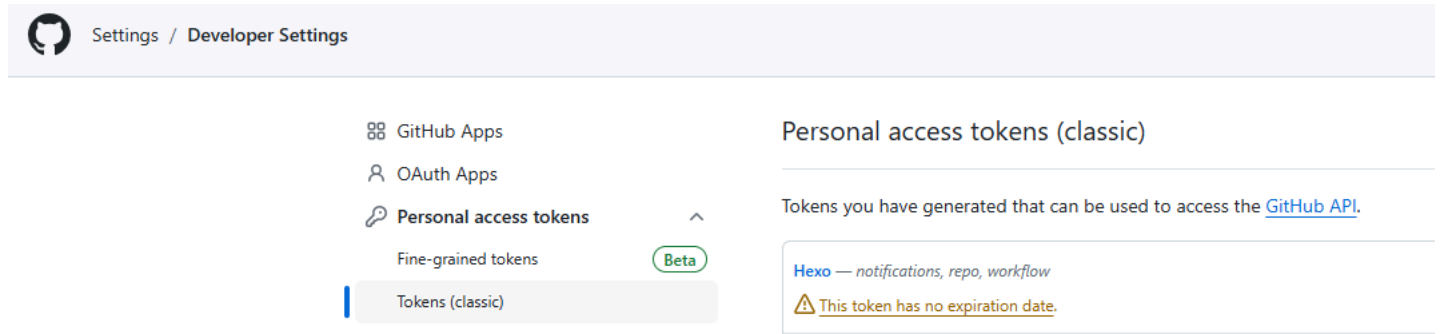
可以选择利用VSCode等软件直接对项目开源到github仓库。

也可以手动去github创建一个空仓库，然后手动在命令行中推送。

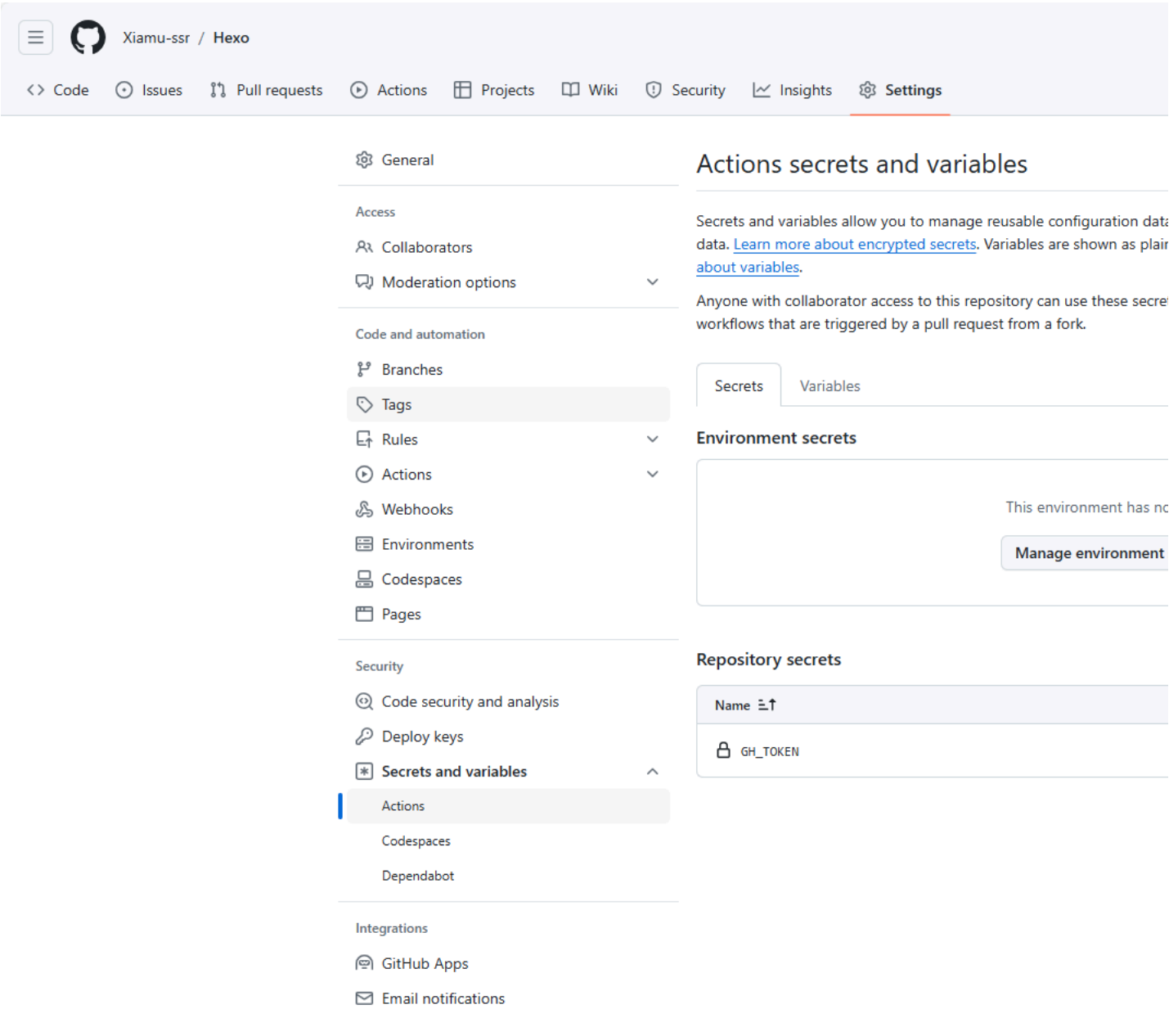
```
1 git init
2 git remote add origin https://github.com/yourusername/your-repo.git
3 git add .
4 git commit -m "Initial commit"
5 git push -u origin main
```

### 3. 创建Token

在个人设置中新增一个Personal access tokens。至少要包含repo权限，然后记住token。  
这个token是给Github Action用的，Github Action会把Hexo编译部署到gh-pages分支。



随后在存放Hexo代码的仓库里把这个Token新增进去，名称为GH\_TOKEN(随意，后面需要一致)。



4. 修改\_config.yml

在\_config.yml中修改deploy字段。指示Hexo在deploy时的推送地址。

```
1 deploy:
2   type: git
3   repo: https://github.com/yourusername/your-repo.git
4   branch: gh-pages
```

5. 配置Github Action workflow

在 .github 文件夹下新增workflows文件夹，然后新增deploy.yml文件，内容如下。

里面有个node-version要和你本地的node一致。

步骤大致意思就是使用ubuntu-latest作为基础环境，然后安装各种依赖，随后hexo generate生成博客网站静态文件夹，把这个文件夹推送到同一仓库的gh-pages分支。

```
1 name: Deploy Hexo to GitHub Pages
2
3 on:
4   push:
5     branches:
6       - main # 当推送到 main 分支时触发
7
8 jobs:
```

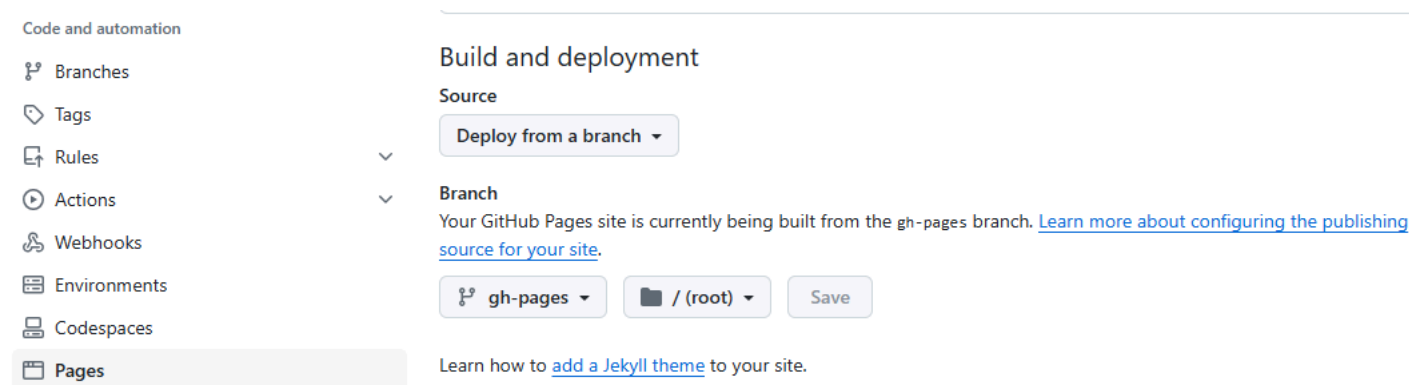
```
9   build:
10     runs-on: ubuntu-latest
11
12     steps:
13       - name: Checkout repository
14         uses: actions/checkout@v2
15         with:
16           submodules: false # 禁用子模块检查
17
18       - name: Setup Node.js
19         uses: actions/setup-node@v2
20         with:
21           node-version: '18'
22
23       - name: Install Dependencies
24         run: npm install
25
26       - name: Install Hexo Git Deployer
27         run: |
28           npm install hexo-deployer-git --save
29           npm install hexo-cli -g
30
31       - name: Clean and Generate Static Files
32         run: |
33           hexo clean
34           hexo generate
35
36       - name: Configure Git
37         run: |
38           git config --global user.name 'github-actions[bot]'
39           git config --global user.email 'github-actions[bot]@users.noreply.github.com'
40
41       - name: Deploy to GitHub Pages
42         env:
43           GH_TOKEN: ${ secrets.GH_TOKEN }
44         run: |
45           cd public/
46           git init
47           git add -A
48           git commit -m "Create by workflows"
49           git remote add origin https://${ secrets.GH_TOKEN }@github.com:yourusername/your-repo.git
50           git push origin HEAD:gh-pages -f
```

## 6. 推送验证

把刚才更新的所有文件都推送一遍，github就会触发工作流，然后去网站看工作流运转的如何。等一切运转完毕，就会发现仓库多出一个gh-pages分支。

## 7. 配置Github Pages

在仓库settings中配置page来源为gh-pages分支即可。等待网站部署完毕，就可以看了。网站链接可以在settings的GitHub Pages看到，也可以去action里看到。



## 8. 修改Hexo主题样式

以一个比较热门的主题为演示示例，参考地址<https://github.com/ppoffice/hexo-theme-icarus>。

若要使用NPM将Icarus安装为Node包，在你的Hexo站点根目录运行如下命令：

```
1 npm install -S hexo-theme-icarus hexo-renderer-inferno
```

接下来，使用hexo命令修改主题为Icarus:

```
1 hexo config theme icarus
```

会发现多出一个\_config.icarus.yml文件。这是Icarus主题的配置文件。

最后推送到仓库，等待部署后，就可以了。

## 10. 添加文章

你可以执行下列命令来创建一篇新文章或者新的页面。

```
1 $ hexo new [layout] <title>
```

您可以在命令中指定文章的布局（layout），默认为 post，可以通过修改 \_config.yml 中的 default\_layout 参数来指定默认布局。

文章添加编辑后，现在只需要推送到仓库，那么github不仅会保存你的Hexo个人博客源码，还会自动更新个人博客静态页面到gh-pages，由此触发github-page功能来更新你的个人博客网站。

## 遇到了一些问题和方案

### 1. 网站没有样式问题

在网站打开F12，发现css等样式资源无法加载，仔细查看报错原因和请求地址，发现并不是当前仓库。

缺少仓库地址，所以把请求地址复制一份，并在后面添加上仓库名即可，这需要修改\_config.yml中修改url字段。yourusername似乎为小写。

推送后等待工作流执行，查看结果。

```
1 url: https://yourusername.github.io/your-repo
2 root: /your-repo/
```

### 2. 图片不显示

在\_config.yml中设置

```
1 post_asset_folder: true
```

意思是每个md博文会单独配套一个同名文件夹，用来存放图片。形如

```
1 source/_posts/
2 |--- my-new-post.md
3 |--- my-new-post/
4 |   |--- example.jpg
```

hexo提供三种语法

```
1 {% asset_path slug %}
2 {% asset_img slug [title] %}
3 {% asset_link slug [title] %}
```

那么在md中可以这样引用图片

```
1 {% asset_img example.jpg This is an example image %}
```

这样一来，部署的时候图片就不会不显示了，但是有个新的问题，我在本地编辑md的时候无法预览图片怎么办。

建议用VSCode下载插件vscode-hexo和Hexo Utils，随后在左边栏目就可以看到新Hexo Utils的新菜单，只要你的VSCode当前打开的文件夹是hexo的根目录，那么插件就会自动识别到，当你对md文件使用“侧边预览”时，图片就正常显示了。