## Hexo 笔记

## **二、准备工作**

（不详细介绍，可参考hexo官方文档：<https://hexo.io/zh-cn/docs/>）

### **1. 安装git。**

### **2. 注册github，新建一个特殊的仓库(username.github.io)。**

### **3. 安装Node.js。**

## **三、安装hexo，并初始化**

参考官网教程：<https://hexo.io/zh-cn/index.html>

在你需要安装hexo的地方打开git命令行工具，运行一下命令（注：#开头为注释）

# 全局安装hexo# 并初始化一个文件夹Hexo作为hexo的根目录（可自定义，命令格式为`hexo init <folder>`）# 进入Hexo文件夹# npm install：Hexo 将会在指定文件夹中新建所需要的文件。# hexo server：部署在本地，会提醒可通过`http://localhost:4000`访问，`Ctrl+C`可停止

npm install hexo-cli -g

hexo init Hexocd Hexo

npm install

hexo server

### 创建新文章

<https://hexo.io/zh-cn/docs/writing>

根目录命令行输入hexo new <模板> <文章名>新建文章

其中<模板>在scaffolds中，默认情况下有三个模板post、draft、page

| **参数** | **功能** | **路径** |  |
| --- | --- | --- | --- |
| post | 新建文章 | /source/\_posts/ |  |
| draft | 新建草稿 | /source/\_drafts/ |  |
| page | 新建页面（标签页，分类页等） | /source/ |  |

## **将hexo部署到github上面，实现双分支部署**

引用master分支中README.md中的有关双分支的部署说明：

yansheng836.github.io

-- master 部署分支：主要存放生成的静态文件，用于部署到github pages

-- blog 写作分支：主要存放hexo环境配置、博客（md）

### **1. 部署master分支**

#### **1.1 指定hexo部署的方式、仓库、分支**

修改hexo根目录下的 \_config.yml

#部署到github的仓库#Deployment#Docs: https://hexo.io/docs/deployment.htmldeploy:

type: git

repository: https://github.com/yansheng836/yansheng836.github.io.git

branch: master

#### **1.2 为项目添加说明README.md**

你可以把README.md,LICENSE（许可证）等文件放在/source/目录下，该目录的文件可以上传到github根目录；之前提过hexo g是将/source//.md所有文件渲染成html文件（如果没有设置跳过渲染的话），然后放到对应的/public/目录下；而对于其他文件，直接复制到对应文件夹下。

设置跳过渲染README.md，需要修改hexo根目录下的 \_config.yml

#hexo g 会默认将所有md文件转换成（渲染成）HTML文件放到public文件夹中；加上这句话的意思就是告诉hexo的解析器，跳过渲染(skip\_render)README.md文件skip\_render:

- 'README.md'

如果不设置该属性，README.md文件会被装换成README.html文件，同理你可以通过该配置跳过渲染你不希望hexo渲染的md文件。

#### **1.3 把master分支部署到github的master分支**

如果有修改配置文件，最好都进行重新编译。

hexo clean

hexo g

hexo d

如果出现类似错误：

ERROR Deployer not found: git

需要添加git依赖：

npm install hexo-deployer-git --save

### **2. 部署blog分支**

注意：该分支名可自定义。

#### **2.1 在Hexo根目录打开git命令行，初始化一个仓库**

在hexo根目录下新建仓库，用于管理写作分支（之前的部署到github的为部署分支）

git init

git remote add origin https://github.com/yansheng836/yansheng836.github.io.git

#### **2.2 忽略一些不需要备份的文件**

修改跟目录下的.gitignore，.gitignore的原内容：

.DS\_StoreThumbs.dbdb.json\*.lognode\_modules/public/.deploy\*/

在后面增加以下内容，用来忽略原来的主题文件：

#ignored the useless themes

/themes/landscape/

#### **2.3 删除/themes/yilia/.git文件夹。**

为方便将yilia的主题配置进行同步（其实该主题已经很久没更新了……），但是之前我们使用github克隆的yilia，也就是话说：/themes/yilia/里面也有一个仓库，这涉及到git仓库下的子仓库的问题，默认情况下是不将该子仓库介入到外层仓库的，但是我们需要备份yilia主题的配置文件，所以需要删除该子仓库，即删除/themes/yilia/.git文件夹。

#### **2.4 创建blog分支。**

创建一个分支存放写作内容，方便同步到同一仓库的不同分支，先提交一次，不然会报

fatal: Not a valid object name: 'master'

添加、提交

git add .

git commit -m "init the blog"

创建分支，并同步到github的blog分支

git branch blog

git checkout blog

git push origin blog

#### **2.5 github上把blog分支设置为默认分支**

可设置该分支为默认分支Settings-->branches-->Default branch:master设置为blog-->update，这样以后仓库主页显示就是blog分支，而且拉取（或者克隆）的时候拉取的就是blog分支，而不是原来默认的master分支。

#### **2.6 实现双分支部署**

每次写完博客要同时同步到该仓库的不同分支上，以实现真正的同步(备份)。

#Writing# 同步到master分支

hexo clean

hexo generate

hexo deploy

# 手动同步到blog分支上

git add.

git commit -m "提交信息"

git push origin blog #如果设置了blog为默认分支，可以直接git push

### **五、发布到github pages**

设置master分支为主页，步骤Settings-->options-->GitHub Pages:设置master分支-->update，成功后会刷新，显示则可通过<https://yansheng836.github.io/>访问。

如果有购买域名，可设置对应域名，之前要设置域名解析（这个好像有点啰嗦不细讲），或者直接在hexo根目录的/source/文件夹下添加CNAME文件(注意：该文件没有后缀)，内容为域名：

www.yansheng.xyz

然后发布到github上

hexo clean

hexo generate

hexo deploy

成功后在setting中会显示可通过域名访问，而不是原来的路径。