

# VuePress学习

---

全局安装前我们需要Git和node这两个软件，关于怎么安装可以我之前hexo的视频教程

加入这两个都没有安装好，那么下面就不需要看了哈，栈友们

## 全局安装

---

首先我们先全局安装一下

```
npm install -g vuepress # 或者 yarn global add vuepress
```

这里等待安装完成后，我们在命令行输入 `vuepress` 或者输入 `vuepress -V` 可以看到

```
vuepress
Usage: vuepress <command> [options]
Options:
  -V, --version          output the version number
  -h, --help             output usage information
Commands:
  dev [options] [targetDir]  start development server
  build [options] [targetDir] build dir as static site
  eject [targetDir]          copy the default theme into .vuepress
Run vuepress <command> --help for detailed usage of given command
```

```
vuepress -V
0.14.8
```

## 目录结构

---

好，安装成功后，我们先来了解一下vuepress的目录结构

首先我们先新建一个项目，项目名为“**vuepress**”，然后我们进入到这个文件夹,我们在这个文件夹的空白处右键Git Bash Here，我们会看到一个黑底的命令工具，上面的全局安装也是可以在这里来安装和查看版本号的，这里不多说了，直接输入下面的命令哈

我们先输入一行命令，生成package.json文件

```
npm init -y
```

回车后，我们可以看到

```
{
  "name": "vpress",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

然后你回到vuepress文件夹内，本来是空的文件夹，现在多出来一个package.json文件

## 开始写作

---

### 新建README.md文件

在命令框输入

```
echo '# Hello VuePress!' > docs/README.md
```

输入完后，还是回到vuepress文件夹内，我们可以看到多出来一个docs的文件夹里面有一个README.md文件，打开这个文件可以看到有Hello VuePress的字样

## 运行测试一下

在项目根目录的命令行输入

```
vuepress dev docs

VuePress dev server listening at http://localhost:8080/
```

我们在浏览器打开 <http://localhost:8080/> 可以看到一个很简洁的页面

## 装饰首页

### 首页配置

我们来编辑一下docs目录下的README.md文件

```
---
home: true
heroImage: /mackxin.png
heroText: 馨客栈
tagline: 关注分享, 关注导航, 关注馨客栈
actionText: 每日更新 →
actionLink: /fuli/
features:
- title: 馨客栈导航
  details: 以 Markdown 为中心的项目结构, 以最少的配置帮助你专注于写作。
- title: 馨客栈前端导航
  details: 享受 Vue + webpack 的开发体验, 在 Markdown 中使用 Vue 组件,
- title: 馨客栈每日分享
  details: VuePress 为每个页面预渲染生成静态的 HTML, 同时在页面被加载的时候
footer: MIT Licensed | Copyright © 2018-present Mackxin
---
```

我们配置完后直接到浏览器看效果, 感觉是不是不一样了哈

### 创建配置文件

在配置之前我们先要在docs文件夹下面新建一个新的文件夹, 名字为“**.vuepress**”, 记得前面是有点的哦, 不要忘记了。然后我们进入到这

个.vuepress文件夹，再在里面新建一个config.js文件，我们的导航配置文件就是在里面配置的哈

## 配置favorite.icon

在.vuepress文件夹下新建public文件夹，把自己制作好的icon放进去

然后我们在config.js配置文件来操作

```
// .vuepress/config.js
module.exports = {
  title : 'mackxin',
  description : 'xininn',
  head : [
    ['link',{rel:'icon',href:'/mackxin.ico'}]
  ]
}
```

现在重新执行命令 `yarn docs:dev` 就可以看到效果了

## 导航配置

我们主要配置的也是config.js文件

```
module.exports = {
  title : 'mackxin',
  description : 'xininn',
  head : [
    ['link',{rel:'icon',href:'/mackxin.ico'}]
  ],
  themeConfig: {
    nav: [
      { text: '首页', link: '/' },
      { text: '馨客栈导航', link: 'http://mackxin.com/nav.htm' },
      { text: '馨客栈前端导航', link: 'http://mackxin.com/webn' },
      { text: '馨客栈每日分享', link: 'http://mackxin.com/fx.' },
      { text: '关于', link: '/about' },
      {
        text: '分享',
        items:[
          { text: '技术' , link:'/jishu/'},
          { text: '每日分享' , link:'/fx/' }
        ]
      }
    ],
  },
}
```

```

    { text: 'GitHub', link: 'https://github.com/mackxin'
    // 下拉列表显示分组
    {
      text: '学习',
      items: [
        {
          text: '前端',
          items: [
            { text: 'js', link: '/js/' },
            { text: 'css', link: '/css/' }
          ]
        },
        {
          text: '后端',
          items: [
            { text: 'php', link: '/php/' },
            { text: 'java', link: '/java/' },
          ]
        },
      ],
    ]
  },
  sidebarDepth: 2,
  lastUpdated: 'Last Updated'
}
}

```

这里我们要新建about、js、css、php、java、jishu、fx等七个文件夹，我们来看看新建完的目录结构哈

```

├─docs      // docs目录下
│  ├─.vuepress  //文件夹的名字
│  │  └─ public  //文件夹的名字
│  │     └─logo.png
│  │  └─ config.js  //配置文件
│  └─js      //文件夹的名字
│     └─README.md  //js首页文件
├─css      //文件夹的名字
│  └─README.md  //css首页文件
├─php      //文件夹的名字
│  └─README.md  //php首页文件
├─java     //文件夹的名字
│  └─README.md  //java首页文件
├─jishu    //文件夹的名字
│  └─README.md  //技术首页文件
├─fx       //文件夹的名字

```

```
|    |—README.md    //分享首页文件
|  about.md    //关于页面
|  README.md    //首页配置
```

## 侧边栏配置

主要配置的也是config.js文件

```
// .vuepress/config.js
module.exports = {
  themeConfig: {
    sidebar: {
      '/fx/': [
        '',
        'fx1',
        'fx2'
      ],
      '/js/': [
        '',
        'js1',
        'js2'
      ],
      '/css/': [
        '',
        'css1',
        'css2'
      ],
      '/php/': [
        '',
        'php1',
        'php2'
      ],
      '/java/': [
        '/java/', // JAVA文件夹的README.md 不是下拉框形式
        {
          title: 'java标题',
          children: [
            '/java/java1', // 以docs为根目录来查找文件
            '/java/java2' // 以docs为根目录来查找文件
          ]
        }
      ]
    }
  }
}

// 下面的是首页显示侧边栏的，不需要首页显示的话可以删掉代码
// '/': [
//   '', // * / */
//   'contact', /* /contact.html */
```

```
        //      'about'      /* /about.html */
        // ]
    }
}
}
```

## 部署

这里暂时只讲部署到GitHub Pages的哈

首先我们在package.json文件里面添加代码，如下

```
{
  "scripts": {
    "docs:build": "vuepress build docs",
    "d": "bash deploy.sh"
  }
}
```

然后我们在vuepress文件夹下新建一个名为deploy.sh的文件，跟docs文件夹同级的哈，不要放错地方了

如果你想发布到 <https://github.io> ,记得吧文件里面的 `<USERNAME>` 改成你的github的用户名哦，不然无效的哈

```
#!/usr/bin/env sh

# 确保脚本抛出遇到的错误
set -e

# 生成静态文件
npm run docs:build

# 进入生成的文件夹
cd docs/.vuepress/dist

# 如果是发布到自定义域名
# echo 'www.example.com' > CNAME

git init
git add -A
git commit -m 'deploy'

# 如果发布到 https://<USERNAME>.github.io
```

```
git push -f git@github.com:<USERNAME>/<USERNAME>.github.io.git m

# 如果发布到 https://<USERNAME>.github.io/<REPO>
# git push -f git@github.com:<USERNAME>/<REPO>.git master:gh-pag

cd -
```

如果你想发布到 <https://github.io/> , 记得吧文件里面的 `<USERNAME>` 改成你的github的用户名哦, 还有就是把你的 `<REPO>` 改成你新建的仓库的名字哦

这里要注意一下哦

如果你想发布到 <https://github.io/>

那么你需要到config.js文件里面配置一下哦

```
base: '/vblog/'
```

```
#!/usr/bin/env sh

# 确保脚本抛出遇到的错误
set -e

# 生成静态文件
npm run docs:build

# 进入生成的文件夹
cd docs/.vuepress/dist

# 如果是发布到自定义域名
# echo 'www.example.com' > CNAME

git init
git add -A
git commit -m 'deploy'

# 如果发布到 https://<USERNAME>.github.io
# git push -f git@github.com:<USERNAME>/<USERNAME>.github.io.git

# 如果发布到 https://<USERNAME>.github.io/<REPO>
git push -f git@github.com:<USERNAME>/<REPO>.git master:gh-pages

cd -
```

弄好以后我们定位在 **vuepress** 的空白处, 右键git bash here



## 输入命令

```
npm run d
```

如果你看到的是下面的这样，那就说明你成功了哈

```
> vpress@1.0.0 d H:\blog\vpress
> bash deploy.sh

> vpress@1.0.0 docs:build H:\blog\vpress
> vuepress build docs

  WAIT  Extracting site metadata...
[15:53:56] Compiling Client
[15:53:56] Compiling Server
[15:54:35] Compiled Server in 39s
[15:54:36] Compiled Client in 40s
  WAIT  Rendering static HTML...

  DONE  Success! Generated static files in docs\.vuepress\dist.
  .....
  .....
  这里省略好多英文
  .....
  .....
    create mode 100644 php/php1.html
    create mode 100644 php/php2.html
Enumerating objects: 56, done.
Counting objects: 100% (56/56), done.
Delta compression using up to 2 threads
Compressing objects: 100% (53/53), done.
Writing objects: 100% (56/56), 83.58 KiB | 950.00 KiB/s, done.
Total 56 (delta 30), reused 0 (delta 0)
remote: Resolving deltas: 100% (30/30), done.
To github.com:mackxin/vblog.git
* [new branch]      master -> gh-pages
```

现在你打开你的网站，我的就是 <https://mackxin.github.io/vblog/>