

Docsify介绍

docsify 是一个动态生成文档网站的工具。不同于 GitBook、Hexo 的地方是它不会生成将 `.md` 转成 `.html` 文件，所有转换工作都是在运行时进行。

这将非常实用，如果只是需要快速的搭建一个小型的文档网站，或者不想因为生成的一堆 `.html` 文件“污染” commit 记录，只需要创建一个 `index.html` 就可以开始写文档而且直接[部署在 GitHub Pages](#)。

[中文文档](#)

[更多插件](#)

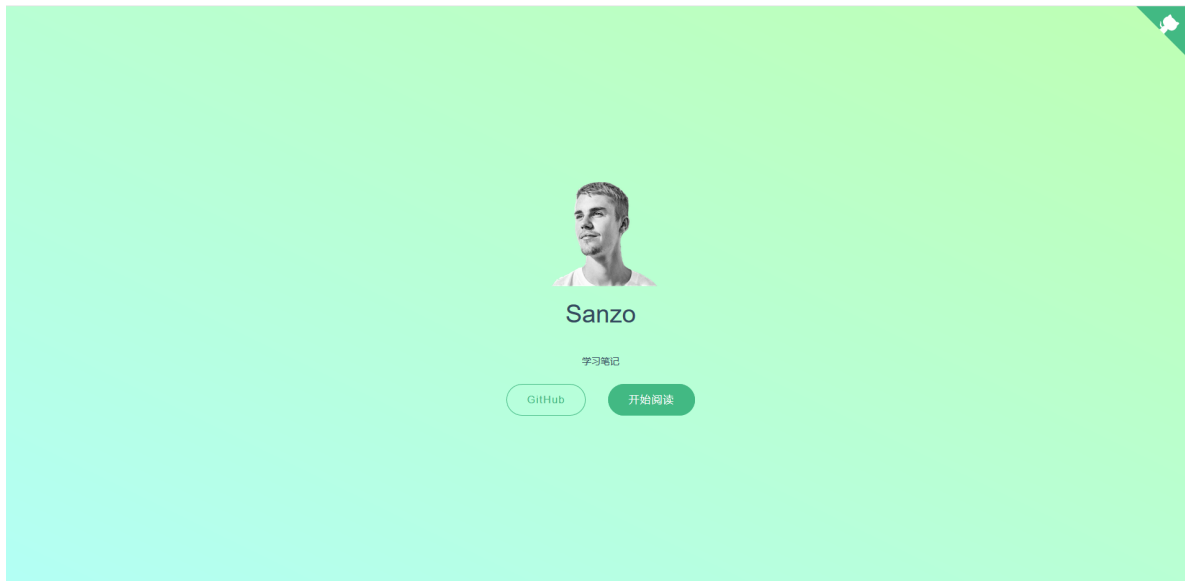
页面展示

可以参考我个人搭建的docsify。

项目地址：github.com/Sanzona/Note。

预览地址：sanzo.top

欢迎页面



首页

搜索

Sanzo

C++Linux 命令Linux 系统编程Linux 系统设置树莓派Windows算法与数据结构学习日记翻墙

记录下自己的学习笔记

笔记主要包括: C++, Linux, 树莓派, 算法和数据结构, 数据库, 操作系统...

0 条评论未登录用户

说点什么

支持 Markdown 语法使用 GitHub 登录预览

来做第一个留言的人吧!

正文

搜索

Sanzo

返回机器学习NumPy笔记Pandas笔记Matplotlib笔记线性代数计算机网络OSI 七层模型TCP/IP四层模型数据封装以太网帧ARP数据包IP数据包UDP数据包TCP数据包滑动窗口拥塞控制三次握手四次挥手time_waitclose_wait连接队列

目录

OSI 七层模型

OSI参考模型

TCP/IP模型

7 应用层

6 表示层

5 会话层

4 传输层

3 网络层

2 数据链路层

1 物理层

应用层

传输层

网络层

网络接口层

1. 物理层: 主要定义物理设备标准, 如网络的接口类型, 光纤的接口类型, 各种传输介质的传输速率等。它的主要作用是传输比特流 (就是由1、0转化电流强弱来进行传输, 到达目的地后再转化为1、0, 也就是我们常说的数模转换与模数转换), 这一层的数据叫做比特。

2. 数据链路层: 定义了如何让格式化数据以帧为单位进行传输, 以及如何让控制对物理介质的访问。这一层通常还提供错误检测和纠正, 以确保数据的可靠传输。如: 串口通信中使用的115200, 8, N, 1

3. 网络层: 在位于不同地理位置的网络中的两个主机系统之间提供连接和路径选择。Internet的发展使得从世界各站点访问信息的用户数大大增加, 而网络层正是管理这种连接的层。

4. 传输层: 定义了一些传输数据的协议和端口号 (WWW端口80等), 如: TCP (传输控制协议, 传输效率高, 可靠性强, 用于传输可靠性要求高, 数据量大的数据), UDP (用户数据报协议)

评论

2 条评论未登录用户

说点什么

支持 Markdown 语法使用 GitHub 登录预览

Sanzona 发表于 5 个月前

hello.

wanice 发表于 14 天前

tql

黑夜模式



环境搭建

```
# 将项目克隆到本地
git clone git@github.com:Sanzona/Note.git
```

欢迎页面

修改 `_coverpage.md` 文件，替换自己的github链接和头像。



目录结构

docsify的目录结构支持子目录。

设置父目录

将自己的目录放到post目录里：

> sanzona.github.io > post >			
名称	修改日期	类型	大小
C++	2020/9/23 23:10	文件夹	
Linux命令	2020/9/23 23:10	文件夹	
Linux系统编程	2020/9/23 23:10	文件夹	
Linux系统设置	2020/9/23 23:10	文件夹	
Windows	2020/9/23 23:10	文件夹	
翻墙	2020/9/23 23:11	文件夹	
树莓派	2020/9/23 23:11	文件夹	
算法与数据结构	2020/9/23 23:11	文件夹	
学习笔记	2020/10/5 20:17	文件夹	

同时修改对应的 `_sidebar.md` 文件。



```
1 <!-- * [简介](README.md) -->
2 * [C++](post/C++/)
3 * [Linux 命令](post/Linux命令/)
4 * [Linux 系统编程](post/Linux系统编程/)
5 * [Linux 系统设置](post/Linux系统设置/)
6 * [树莓派](post/树莓派/)
7 * [Windows](post/Windows/)
8 * [算法与数据结构](post/算法与数据结构/)
9 * [学习笔记](post/学习笔记/)
10 * [翻墙](post/翻墙/)
11
```

添加新文章

将对应的markdown文件添加到对应的目录中，然后修改子目录中的 `_sidebar.md` 文件。

修改预览

在项目目录地址使用cmd，执行 `docsify serve .` 命令，可以在浏览器实时预览修改。

浏览器访问 `localhost:3000`。

```
管理员: C:\Windows\System32\cmd.exe - docsify serve .
Microsoft Windows [版本 10.0.19041.508]
(c) 2020 Microsoft Corporation. 保留所有权利。
C:\Users\na\Desktop\sanzona.github.io>docsify serve .
Serving C:\Users\na\Desktop\sanzona.github.io now.
Listening at http://localhost:3000
```

提交修改

最后通过git命令，将修改同步到仓库中即可。

域名绑定

另外可以将Github Pages绑定个人域名。

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

✓ Your site is published at <http://sanzo.top/>

Source

Your GitHub Pages site is currently being built from the master branch. [Learn more.](#)

Branch: master

/ (root)

Save

Custom domain

Custom domains allow you to serve your site from a domain other than sanzona.github.io. [Learn more.](#)

sanzo.top

Save

☐ Enforce HTTPS

HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traffic to your site. When HTTPS is enforced, your site will only be served over HTTPS. [Learn more.](#)

tabs插件

官方文档

语法1

可以放代码块。

**** cpp ****

```
cout << "hello world!" << endl;
```

**** python ****

```
print("hello world!")
```

```
<!-- tabs:start -->

#### ** cpp **

```cpp
cout << "hello world!" << endl;
```

#### ** python **

```python
print("hello world!")
```

<!-- tabs:end -->
```

语法2

不能放代码块。

Hello!

Bonjour!

Ciao!

```
<!-- tabs:start -->

<!-- tab:English -->

Hello!

<!-- tab:French -->

Bonjour!

<!-- tab:Italian -->

Ciao!

<!-- tabs:end -->
```

主题

Active State

Tab Markdown

```
#### **Active State**
```

**** python ****

```
print("hello world!")
```

pdf插件

[项目地址](#)

```
```pdf
path-to-the-pdf-file
```
```

```
post/Others/pdf/docsify.pdf
```

文件嵌入

同时好像只能嵌入一个。

```
# 以codeb lock形式嵌入md文件
[filename](path/test.md ':include :type=code')

# 嵌入网页
[sanzo.top](https://sanzo.top ':include :type=iframe width=100% height=400px')

# 嵌入视频
[001](img/test.mp4 ':include :type=vedio')
[bilibili](//player.bilibili.com/player.html?
aid=582621141&bvid=BV1x64y1u7t9&cid=172456678&page=1 ':include :type=vedio')
```

[bilibili](#)