



链滴

# 在 windows 平台上构建 logseq

作者: [zxhd86](#)

原文链接: <https://ld246.com/article/1656236483260>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# 目的

最近打算学习一下函数式编程语言，挑来挑去选中了 clojure。既然要深入学习，那么肯定就得再学了入门的语法后，再找一个高级的开源项目进行学习，所以最后就挑中了 logseq，而且确实也对开源平台上少有的另一个双链笔记感到兴趣。

在本地的构建中，也遇到了不少坑，虽然这些对于专业的开发者来说可能不值一提，但我想，这一份经验对于一些配置构建依赖配置得焦头烂额的萌新还是有点帮助的，故发此文。

# 前提

- 一个能运行 wsl2 的 windows 操作系统
- 一个提供本地端口的代理软件

# 步骤

现在我来总结一下，一个 windows 下 logseq 环境的配置吧。

- 打开你的 windows 代理软件，并且配置为允许局域网接入，记住端口
- 查看自己的局域网 ip 地址，也就是 192.168.\*.\*  
这就是你之后要写的代理地址了。我的地址为 192.168.0.112
- 配置 http 代理，主要是为了 git，wsl 的 apt 貌似不认 http 代理的环境变量

- 

```
vim ~/.profile
```

```
#末尾加入加入：
```

```
#host_ip=$(cat /etc/resolv.conf |grep "nameserver" |cut -f 2 -d " ")
host_ip=192.168.0.112
export http_proxy="http://$host_ip:10808"
export https_proxy="http://$host_ip:10808"
```

- 配置 apt 代理，否则下面难做了

- 

```
sudo vim /etc/apt/apt.conf
```

```
# 加入
```

```
Acquire::http::Proxy "http://192.168.0.112:10808";
```

```
#退出并保存
```

```
:wq
```

- 安装大佬的 [透明代理](#)

- 

```
# 更新
```

```
sudo apt update
```

```
# 安装编译环境
```

```
sudo apt install gcc make
```

```
# 在 linux 子系统下执行
```

```
cd /usr/local && sudo git clone https://github.com/akirarika/wsl2proxy.git && cd wsl2proxy  
& sudo chmod +x ./wsl2proxy
```

```
vim ~/.bashrc
```

```
# 插入以下文本
```

```
# 向脚本传递两个参数，以指定你在 windows 下代理程序的协议和端口
```

```
# source /usr/local/wsl2proxy/wsl2proxy [协议] [端口]
```

```
source /usr/local/wsl2proxy/wsl2proxy http 10808
```

```
# 保存退出
```

```
:wq
```

```
# 安装依赖
```

```
installProxychains
```

- 安装 jdk、git

- 

```
sudo apt install openjdk-17-jdk git
```

- 安装 clojure

- 

```
# 进入透明代理环境
```

```
proxy bash
```

```
# 先安装 brew 包管理器
```

```
mkdir brew
```

```
cd brew
```

```
git clone https://github.com/Homebrew/install.git
```

```
cd install
```

```
./install.sh
```

```
# 按照要求走就是了，记得最后按照它的要求添加环境变量和安装依赖
```

```
echo 'eval "$(/home/linuxbrew/.linuxbrew/bin/brew shellenv)"' >> /home/zxhd/.profile
```

```
eval "$(/home/linuxbrew/.linuxbrew/bin/brew shellenv)"
```

```
sudo apt-get install build-essential
```

```
#开始安装 clojure
```

```
brew install clojure/tools/clojure
```

```
# 安装后打开一下，自动更新
```

```
clj
```

#退出

- 安装 node.js

- 

# 进入透明代理环境

proxy bash

# 安装 node.js 版本管理器

git clone https://github.com/nvm-sh/nvm.git

./nvm/install.sh

# 可以重启终端也可以直接运行下面的命令

export NVM\_DIR="\$HOME/.nvm"

[ -s "\$NVM\_DIR/nvm.sh" ] && \. "\$NVM\_DIR/nvm.sh" # This loads nvm

[ -s "\$NVM\_DIR/bash\_completion" ] && \. "\$NVM\_DIR/bash\_completion" # This loads nvm b  
sh\_completion

# 安装 node.js

# 可以用下面那个命令查看可用版本

# nvm ls-remote

nvm install v16.15.1

- clone logseq 并运行自动配置脚本

- 

# 进入透明代理环境

proxy bash

# clone logseq

git clone https://github.com/logseq/logseq

cd logseq

yarn

yarn watch

- 在 windows 上连接到编译环境

- 

# 在一些比较老的版本上可以直接用 localhost:\* 连接，新的版本

# 必须运行下面命令，得到地址后才能用

ip addr | grep eth0