

使用 nvm

如果 GitHub 无法进入, 可以参考这篇文章: [解决 Github 无法登录的问题](#)

- Windows 修改 `C:\Windows\System32\drivers\etc`
- MAC 修改 `\etc\hosts`

```
1 140.82.113.4 github.com
2 199.232.69.194 github.global.ssl.fastly.net
```

使用 nvm 好处: 快速切换或更新 node 版本, 并能保持系统干净

安装 nvm

- Mac 安装官网安装步骤操作即可: [nvm Github](#)

```
1 curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh
  | bash
2
3 # 修改 ~/.zshrc、~/.profile、~/.bashrc
4 export NVM_DIR="$([ -z "${XDG_CONFIG_HOME-}" ] && printf %s
  "${HOME}/.nvm" || printf %s "${XDG_CONFIG_HOME}/nvm")"
5 [ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh"
```

或者直接在 GitHub 下载

```
1 git clone https://github.com/creationix/nvm.git ~/.nvm
2 # 在 ~/.zshrc、~/.profile、~/.bashrc 添加以下命令
3 echo "source ~/.nvm/nvm.sh" >> ~/.bashrc
```

- Windows 直接下载安装包: [nvm-windows/releases](#)

切换淘宝源

- 直接修改 nvm 安装路径中的 `settings.txt`

```
1 node_mirror: https://npm.taobao.org/mirrors/node/
2 npm_mirror: https://npm.taobao.org/mirrors/npm/
```

- 或直接在命令行工具执行如下命令 (查看 `setting.txt` 文件看是否修改成功)

```
1 nvm node_mirror https://npm.taobao.org/mirrors/node/
2 nvm npm_mirror https://npm.taobao.org/mirrors/npm/
```

nvm 常用命令

- Node 以往版本

安装前, 可以去官网查一下都有哪些版本

虽然也可以通过 `nvm list available` 去查看, 不过显示的是不完整 (`partial`) 的列表

```

1 # 查看安装 node 版本(nvm list)
2 nvm ls
3
4 # 安装指定 node 版本
5 nvm install ...
6
7 # 卸载指定 node 版本
8 nvm install ...
9
10 # 使用对应的 node 版本 (命令行工具需要是管理员)
11 nvm use ...

```

更改 npm 配置

- 切换淘宝源

```

1 npm config set registry https://registry.npm.taobao.org
2 # 查看下载镜像源
3 npm config get registry

```

- 修改 npm 全局安装路径

注意：并将此目录添加到环境变量中

```

1 npm config set prefix "D:\node\global"
2 # 查看全局安装路径
3 npm prefix -g

```

- 修改 npm 全局缓存路径

```

1 npm config set cache "D:\node\cache"
2 # 查看全局缓存路径
3 npm config get cache

```

- 修改 .npmrc 加速 c++ 库的 npm 镜像

`vim ~/.npmrc` 增加如下配置

```

1 # init config
2 init-author-name = lyn
3 init-version = 0.1.0
4 init-license = MIT
5
6 # mirror config
7 sharp_dist_base_url = https://npm.taobao.org/mirrors/sharp-libvips/v8.9.1/
8 profiler_binary_host_mirror = https://npm.taobao.org/mirrors/node-
  inspector/
9 fse_binary_host_mirror = https://npm.taobao.org/mirrors/fsevents
10 node_sqlite3_binary_host_mirror = https://npm.taobao.org/mirrors
11 sqlite3_binary_host_mirror = https://npm.taobao.org/mirrors
12 sqlite3_binary_site = https://npm.taobao.org/mirrors/sqlite3
13 sass_binary_site = https://npm.taobao.org/mirrors/node-sass
14 electron_mirror = https://npm.taobao.org/mirrors/electron/
15 puppeteer_download_host = https://npm.taobao.org/mirrors
16 chromedriver_cdnurl = https://npm.taobao.org/mirrors/chromedriver

```

```
17 operadriver_cdnurl = https://npm.taobao.org/mirrors/operadriver
18 phantomjs_cdnurl = https://npm.taobao.org/mirrors/phantomjs
19 python_mirror = https://npm.taobao.org/mirrors/python
20 registry = https://registry.npm.taobao.org/
21 disturl = https://npm.taobao.org/dist
```

更改 yarn 配置

- 改变 yarn 全局 bin 位置(prefix)

注意：将此目录添加到环境变量中，这样使用 `yarn link` 设置软连接就可以直接使用了

```
1 yarn config set prefix "C:\Users\AppData\Local\Yarn"
2 # 查看 yarn 全局 bin 位置(prefix)
3 yarn global bin
```

- 改变 yarn 全局安装位置(folder)

```
1 yarn config set global-folder "C:\Users\AppData\Local\Yarn"
2 # 查看 yarn 全局安装位置(folder)
3 yarn global dir
```

- 改变 yarn 全局 cache 位置(cache)

这里我只把缓存路径改了，要不然缓存的包占 C 盘太大了

```
1 yarn config set cache-folder "D:\node\cache"
2 # 查看 yarn 全局 cache 位置(cache)
3 yarn cache dir
```

- 改变 yarn 全局 link 目录

```
1 yarn config set link-folder "C:\Users\AppData\Local\Yarn\Data\link"
2 # 查看 yarn 配置
3 yarn config list
```

Mac 环境变量

```
1 ## load nvm
2 export NVM_DIR=$HOME/.nvm
3 [ -s $(brew --prefix nvm)/nvm.sh ] && . $(brew --prefix nvm)/nvm.sh
4 [ -s $(brew --prefix nvm)/bash_completion ] && . $(brew --prefix
nvm)/bash_completion
5
6 ## yarn global path
7 export PATH=$(yarn global bin):$PATH
8
9 ## define alias
10 alias subl="'/Applications/Sublime
Text.app/Contents/SharedSupport/bin/subl'"
11 alias code="'/Applications/Visual Studio
Code.app/Contents/Resources/app/bin/code'"
12 alias weapp="'/Applications/wechatwebdevtools.app/Contents/MacOS/cli'"
13 alias typora="open -a typora"
```

```

14 alias proxy="all_proxy=socks5://127.0.0.1:1086"
15 # alias proxy="http_proxy=socks5://127.0.0.1:1086"
16 # alias proxy="https_proxy=socks5://127.0.0.1:1086"
17 alias reload=". ~/.bash_profile"
18 alias reset-dock="defaults delete com.apple.dock; killall Dock"
19 alias reset-launchpad="defaults write com.apple.dock ResetLaunchPad -bool
true; killall Dock"
20 alias show-hiddens="defaults write com.apple.finder AppleShowAllFiles -
bool true; killall Finder"
21 alias hide-hiddens="defaults write com.apple.finder AppleShowAllFiles -
bool false; killall Finder"
22 alias rm-ds="sudo rm -rf .DS_Store && rm -rf */.DS_Store"
23
24 ## export mirrors
25 export NODEJS_ORG_MIRROR=http://npm.taobao.org/mirrors/node
26 # export IOJS_ORG_MIRROR=http://npm.taobao.org/mirrors/iojs
27 export NVM_NODEJS_ORG_MIRROR=$NODEJS_ORG_MIRROR
28 # export NVM_IOJS_ORG_MIRROR=$IOJS_ORG_MIRROR

```

Mac 常用软件及源切换

安装 typora 免费版本

- [typora](#)
- [typora windows](#)

使用 iterm2 命令行

- [iterm2 + oh my zsh](#)

下载 homebrew

- 推荐使用 [homebrew 清华源](#) 下载
- 报错1: curl: (7) Failed to connect to raw.githubusercontent.com port 443: Connection refused
- 报错2:

```

1 Warning: Suspicious https://github.com/Homebrew/brew git origin remote
found.
2 The current git origin is:
3 https://github.com/Homebrew/homebrew-core
4
5 cd /usr/local/Homebrew/Library/Taps/homebrew/
6 rm -rf homebrew-core
7 git clone https://github.com/Homebrew/homebrew-core.git

```

修改 `/etc/hosts`

1. 不输入密码方式 [vim 强制修改root权限文件](#)

```

1 vim /etc/hosts
2 :w !sudo tee %
3 L
4 :q

```

2. 输入密码方式

```
1 sudo vim /etc/hosts
2 ## 输入管理员密码
3 :wq!
```