

VSCode 远程开发：在 WSL2 下搭建 Node.js 开发环境

Author: 凌文龙, 彭玲 Time: 2021/5/24

VSCode 远程开发：在 WSL2 下搭建 Node.js 开发环境

[什么是 VSCode 远程开发环境?](#)

[为什么需要 VSCode 远程开发环境?](#)

[WSL 环境准备](#)

[Windows 商店安装 Ubuntu 20.04](#)

[在 VSCode 中安装 Remote - WSL](#)

[搭建 Node.js 开发环境](#)

[使用 WSL 环境](#)

[NVM 安装](#)

[Node.js 安装](#)

[Node.js 开发调试](#)

[Troubleshooting](#)

[1. Command 'nvm' not found](#)

[2. Can't find Node.js](#)

[VSCode 插件](#)

[结尾](#)

什么是 VSCode 远程开发环境?

基于 Windows 下的 Linux 子系统 WSL (Windows Subsystem for Linux), 在 VSCode 下搭建的一套 Linux 项目开发环境。

为什么需要 VSCode 远程开发环境?

前端开发同事曾有过 (又或 将来也会有) 这样的疑惑: “我本机开发环境跑得都是正常的, 但是测试环境跑不起来, 咋回事?”

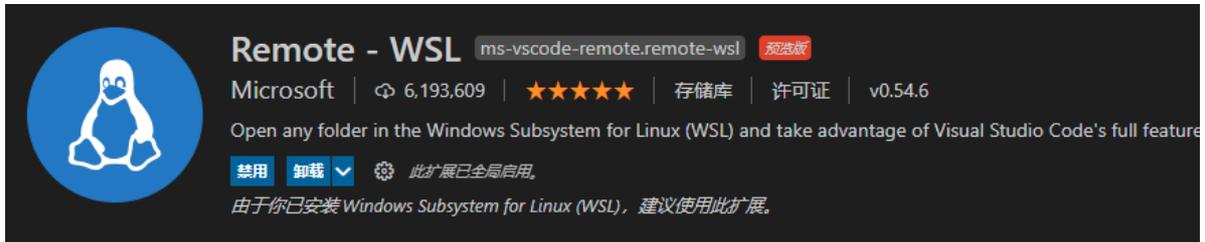
对此, 我们应该首先排除系统环境的问题, 怎么排除? 答案是“使开发环境与部署环境保持一致”; 如何保持一致, 开发环境向部署环境看齐, 部署一套 Linux 开发环境; 如何部署 Linux 开发环境, 正是本文的要旨, 接下来将一步一步解答...

WSL 环境准备

Windows 商店安装 Ubuntu 20.04

安装过程略。

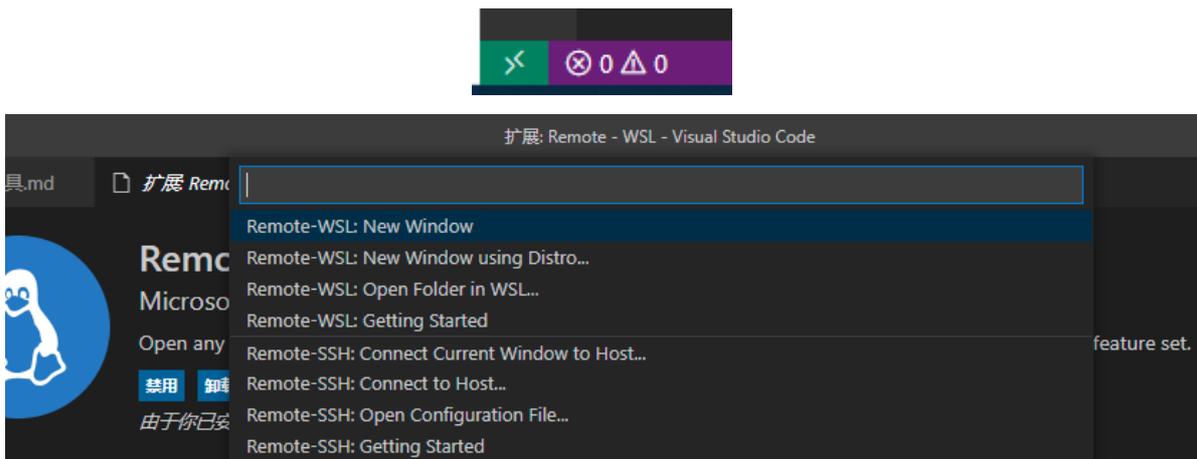
在 VSCode 中安装 Remote - WSL



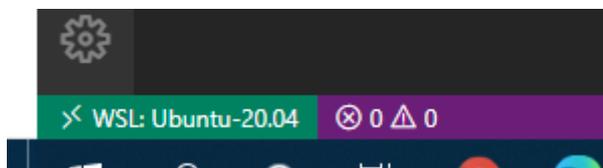
Tips: VSCode 安装 [Remote - WSL](#) 扩展后 需重启 VSCode, 否则, 在 VSCode 中启动项目调试会失败 (找不着 Node.js 环境)。

搭建 Node.js 开发环境

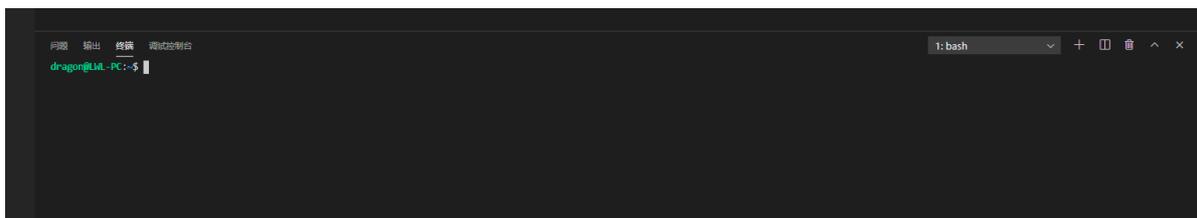
使用 WSL 环境



选择第一项, VSCode 会打开一个新的窗口, 在左下角提示当前连接的 WSL:



右侧终端就是 WSL 子系统的终端:



NVM 安装

NVM Linux 版本安装: <https://github.com/nvm-sh/nvm>

```
sudo apt-get update
```

```
wget -qO- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
```

```
=> Downloading nvm from git to '/home/dragon/.nvm'
=> Cloning into '/home/dragon/.nvm'...
remote: Enumerating objects: 343, done.
remote: Counting objects: 100% (343/343), done.
remote: Compressing objects: 100% (290/290), done.
remote: Total 343 (delta 38), reused 165 (delta 28), pack-reused 0
Receiving objects: 100% (343/343), 193.92 KiB | 2.98 MiB/s, done.
Resolving deltas: 100% (38/38), done.

remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 1 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), 1.84 KiB | 1.84 MiB/s, done.
From https://github.com/nvm-sh/nvm
* tag v0.38.0 -> FETCH_HEAD
* (HEAD detached at FETCH_HEAD)
  master
=> Compressing and cleaning up git repository

=> Appending nvm source string to /home/dragon/.bashrc
=> Appending bash_completion source string to /home/dragon/.bashrc
=> Close and reopen your terminal to start using nvm or run the following to use it now:

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 bash_completion
```

NVM 环境变量设置:

```
source .bashrc
```

NVM 安装验证:

```
nvm -v
```

Node.js 安装

接下来就可以像 Windows 中一样去使用 nvm 管理 Node.js 版本了:

```
nvm install 12.22.1
```

```
nvm use 12.22.1
```

```
dragon@LWL-PC:~$ nvm list
v10.24.1
-> v12.22.1
v14.16.1
default -> 12.22.1 (-> v12.22.1)
iojs -> N/A (default)
unstable -> N/A (default)
node -> stable (-> v14.16.1) (default)
stable -> 14.16 (-> v14.16.1) (default)
lts/* -> lts/fermium (-> v14.16.1)
lts/argon -> v4.9.1 (-> N/A)
lts/boron -> v6.17.1 (-> N/A)
lts/carbon -> v8.17.0 (-> N/A)
lts/dubnium -> v10.24.1
lts/erbium -> v12.22.1
lts/fermium -> v14.16.1
```

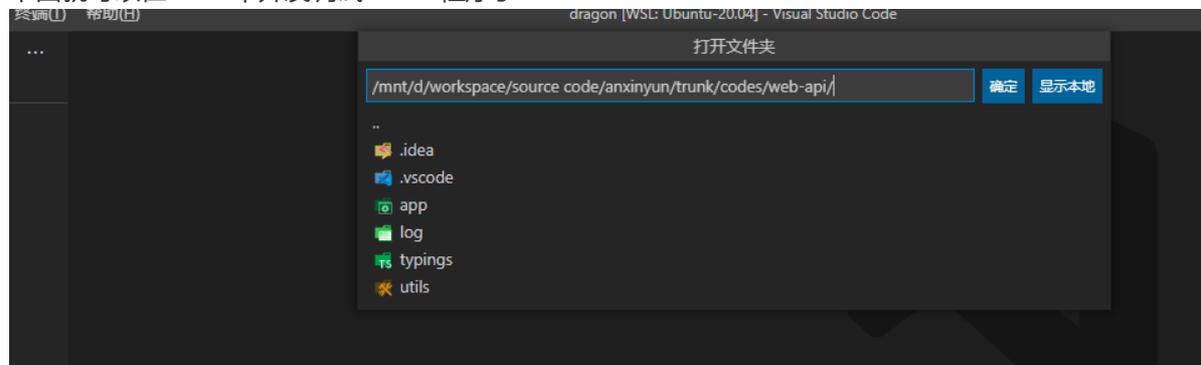
设置 npm 源

```
npm config set registry=http://10.8.30.22:7000
```

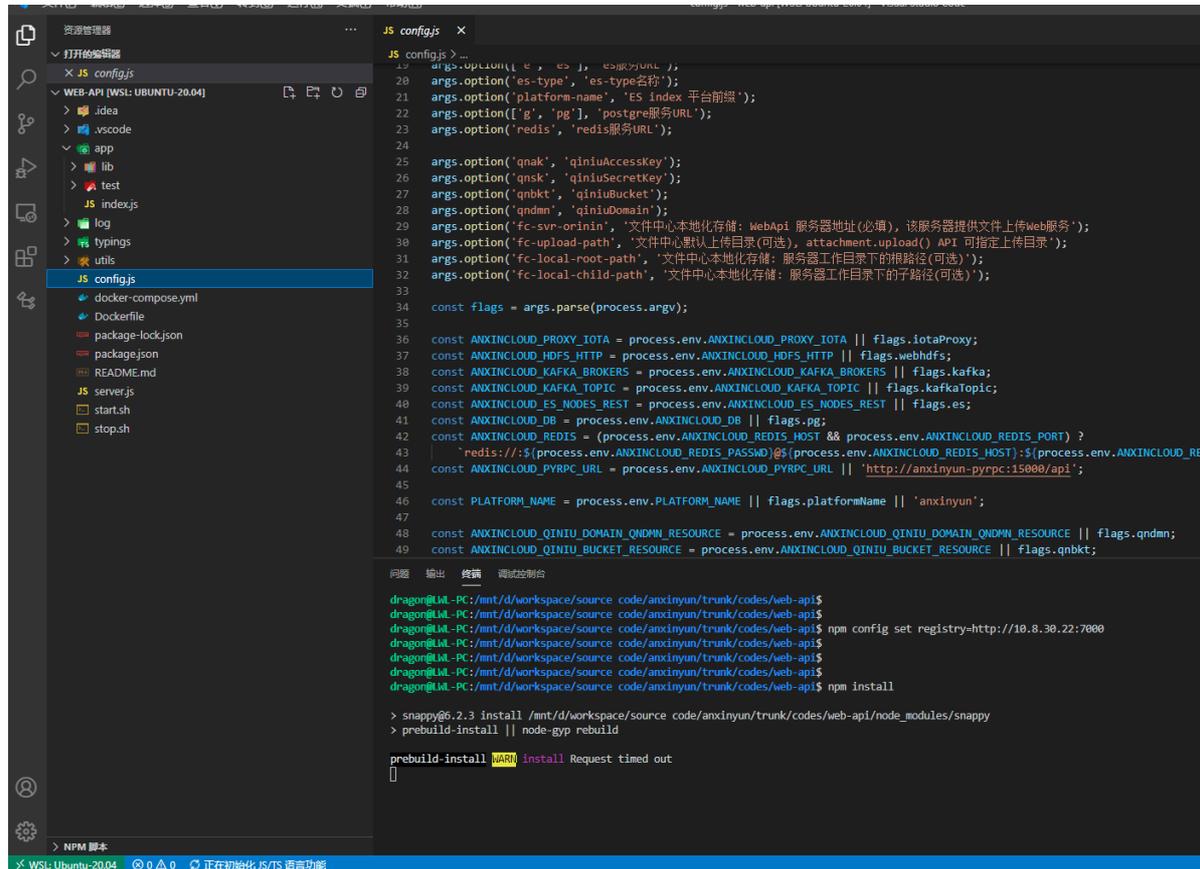
```
dragon@LWL-PC:~/mnt/d/workspace/source code/anxinyun/trunk/codes/web-api$
dragon@LWL-PC:~/mnt/d/workspace/source code/anxinyun/trunk/codes/web-api$
dragon@LWL-PC:~/mnt/d/workspace/source code/anxinyun/trunk/codes/web-api$ npm config set registry=http://10.8.30.22:7000
dragon@LWL-PC:~/mnt/d/workspace/source code/anxinyun/trunk/codes/web-api$
```

Node.js 开发调试

下面就可以在 WSL 中开发调试 node 程序了：



这样就跟在本地开发一样了：



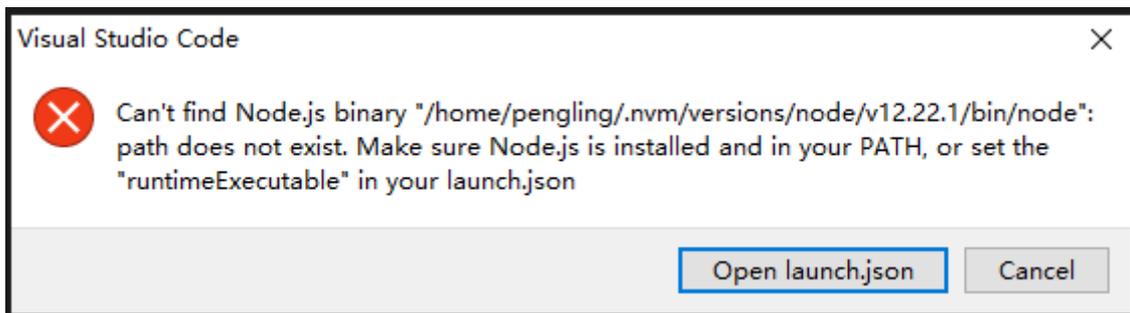
Troubleshooting

1. Command 'nvm' not found

NVM 环境变量问题，设置下 `source ~/.bashrc` 即可。

2. Can't find Node.js

如果遇到下面的问题 (找不到 Node.js)，重启 VSCode 即可。



VSCode 插件

之前在 Windows 系统下安装的插件跟 WSL 环境是独立的，所以，如果 WSL 环境下使用相关插件，需要重新安装——WSL环境下点击 `Install in WSL: Ubuntu-20.04` 标签即可。

结尾

再回到文章开头的问题：

前端开发同事曾有过（又或`将来也会有`）这样的疑惑：“我本机开发环境跑得都是正常的，但是测试环境跑不起来，咋回事？”

有了上面 VSCode 下 WSL 开发环境的搭建，就可以尝试复现测试环境，排除是否由于环境不一致导致的问题了，比如，代码对环境的强依赖问题。

排除代码问题后，重点排查配置及其他吧...