

Using Windows

How to set up a local development environment for Celo using Windows.

tip Many popular tools and resources for blockchain development are written for developers working on UNIX machines. It is common for developers working on Windows to encounter errors that are not covered in the documentation and have no luck with Google. Fortunately, Microsoft makes it easy to run a UNIX machine directly from a Windows desktop with the [Windows Subsystem for Linux](#).

Getting set up with Windows

Open PowerShell as an administrator and run

Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux Restart your computer when prompted.

Next, install a Linux distribution from the Microsoft Store. When developing this guide, we chose [Ubuntu](#).

Set up your Linux distro by setting a username and password then update and upgrade the packages by running the following command in the terminal:

```
sudo  
apt update &&  
sudo  
apt upgrade
```

You can view the source documentation for setting up the Linux distro [here](#) and the Microsoft documentation for setting up the Windows Subsystem for Linux [here](#).

Set up the Linux Environment

Now that you have Linux installed, let's install [nvm](#) and [yarn](#). Nvm (node version manager) makes it easy to install and manage different versions of Node.js. The following instructions are from the [celo-monorepo setup documentation for Linux](#).

Run the following commands in the Linux terminal.

Installing Nvm

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh |  
bash source ~/.bashrc
```

Setting up the desired Nvm version

nvm install x nvm alias default x Running node -v in the terminal should print a node version if it is installed correctly.

Yarn is a package manager similar to npm. The [celo-monorepo](#) uses yarn to build and manage packages. Install yarn with the following command.

Installing Yarn - <https://yarnpkg.com/en/docs/install#debian-stable>

```
curl  
-sS https://dl.yarnpkg.com/debian/pubkey.gpg |  
sudo apt-key add - echo  
"deb https://dl.yarnpkg.com/debian/ stable main"  
|
```

sudo

tee /etc/apt/sources.list.d/yarn.list sudo

apt-get update &&

sudo

apt-get

install

yarn Test that yarn is installed by running yarn --version .

Developing with WSL

You can now start working on your projects in your Linux environment. Install [the WSL VS Code extension](#) for a seamless integration between VS Code and WSL.

Be aware that networking will be different depending on which version of WSL you are using. The details of managing network interfaces goes beyond the scope of this guide, but you can [learn more here](#) . You are good to go! If you have any questions, [join our Discord server](#) and just ask.

Additional Resources

- [Windows Subsystem for Linux Installation Guide for Windows 10](#)
- [WSL: Ultimate Guide](#) [Edit this page](#) [Previous Celo Local Development Environment Using Mac](#) [Next Deploy with Replit](#)