

# Front-end Engineer Environment Setup

Scott C. Krause | 2021

<b>References</b>	<b>2</b>
<b>zshell</b>	<b>2</b>
Default Mac Shell	2
<b>XCode</b>	<b>2</b>
Apple Command Line Dev Tools	2
<b>Finder</b>	<b>3</b>
Finder Config	3
<b>Browsers</b>	<b>3</b>
Chrome Extensions	3
<b>Brew</b>	<b>3</b>
Third party Mac Installer	3
<b>IDE</b>	<b>4</b>
VS Code to Path	4
Visual Studio Code Extensions	4
<b>Node.js</b>	<b>4</b>
Node Version Manager, Node.js, NPM, and global packages	4
Global Packages	5
<b>VPN Dev</b>	<b>5</b>
Tunnelblick (contact administrator first)	5
<b>Java</b>	<b>5</b>
Java 8 SDK	5
<b>Scala</b>	<b>6</b>
Scala Code Runner	6
Scala Turn Off Metrics (Big Sur)	6
<b>Scala Build Tool</b>	<b>7</b>
sbt troubleshooting	7
<b>git</b>	<b>7</b>
Git command line workflow	7
<b>jq</b>	<b>8</b>
Command Line JSON (i think)	8
<b>Productivity Tools</b>	<b>8</b>
Optional but recommended	8

## References

<a href="#">Boostgraph</a>	<a href="#">Cytoscape</a>
<a href="#">JanusGraph</a>	<a href="#">Legacy Doc</a>
<a href="#">Node Version Manager</a>	<a href="#">Vueitify</a>
<a href="#">Undo-Redo_Vuex</a>	<a href="#">Graph Databases</a>

## zshell

Default Mac Shell
<pre>nano ~/.zshrc</pre>
<pre>export NVM_DIR="\$HOME/.nvm" [ -s "\$NVM_DIR/nvm.sh" ] &amp;&amp; \. "\$NVM_DIR/nvm.sh" # This loads nvm</pre>
<pre>printenv</pre>

## XCode

Apple Command Line Dev Tools
<pre>xcode-select --install</pre>
<pre>xcode-select --version</pre>

# Front-end Engineer Environment Setup

Scott C. Krause | 2021

## Finder

Finder Config
<code>defaults write com.apple.Finder AppleShowAllFiles true</code>
<code>killall Finder</code>
Menu > Finder > Preferences [ Select Drive ]

## Browsers

<a href="#">Chrome</a> / <a href="#">Chrome Canary</a>
Chrome Extensions
<a href="#">Vue Extension</a>
<a href="#">Overrides</a>
<a href="#">AxE</a>
<a href="#">Vue Force Dev</a>

## Brew

Third party Mac Installer
<code>/bin/bash -c "\$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"</code>
<code>brew --version</code>

# Front-end Engineer Environment Setup

Scott C. Krause | 2021

## IDE

VS Code to Path
Command + Shift + P Shell command: Install 'code' command in path
Visual Studio Code Extensions
<a href="#">Ventur</a>
<a href="#">Prettier</a>
<a href="#">Postman Interceptor</a>

## Node.js

<a href="#">Node Version Manager</a> , Node.js, NPM, and global packages
<code>curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.37.2/install.sh   bash</code>
<code>nvm --version</code>
<code>nvm install --lts</code>
<code>node --version</code>
<code>npm install -g npm</code>
<code>npm --version</code>

# Front-end Engineer Environment Setup

Scott C. Krause | 2021

## Global Packages

```
npm install -g supervisor live-server svgo
```

```
npm install -g npm  
npm list -g --depth=0
```

## VPN Dev

Tunnelblick (contact administrator first)

Download Tunnelblick: <https://tunnelblick.net> and install it.

Please change your credentials by going to <https://vpn.dev.gomplxos.com>

<https://vpn.dev.gomplxos.com> click on the remote access tab and pick the 3rd download from the list to download the Ovpn file

Verify <http://universe.qweb.qos/>

## Java

### Java 8 SDK

```
brew update  
brew tap caskroom/versions  
brew cask install adoptopenjdk8  
brew install jenv  
echo 'export PATH="$HOME/.jenv/bin:$PATH"' >> ~/.zshrc  
echo 'eval "$(jenv init -)"' >> ~/.zshrc
```

See also:  
[SDKMan](#)  
[JENV](#)

```
java -version
```

```
ls /Library/Java/JavaVirtualMachines  
ls /Library/Java/JavaVirtualMachines/jdk1.8.0_251.jdk/Contents/Home
```

# Front-end Engineer Environment Setup

Scott C. Krause | 2021

```
Printenv JAVA_HOME
```

```
jenv doctor
```

```
cat ~/.zshrc
```

## Scala

### Scala Code Runner

```
brew update  
brew install scala  
brew install sbt  
brew install sbt@1
```

```
scala --version
```

### Scala Turn Off Metrics (Big Sur)

qweb-cyber/src/main/resources/application.conf

```
kamon.modules {  
  host-metrics {  
    enabled = no  
    enabled = ${?COLLECT_HOST_METRICS}  
  }  
  jvm-metrics {  
    enabled = yes  
    enabled = ${?COLLECT_JVM_METRICS}  
  }  
  process-metrics {  
    enabled = no  
    enabled = ${?COLLECT_PROCESS_METRICS}  
  }  
}
```

# Scala Build Tool

sbt troubleshooting
brew install sbt@1
scala --version
sbt --version
sbt clean (Optionally delete .ivy2 )
/usr/local/etc/sbtopts -mem 2048 (Optional Increase Mem)

## git

Git command line workflow
git clone -b CYB-NNNN-fe-desc --single-branch https://gitlab.com/qomplx/teams/cyber/qweb-cyber.git
git branch
<div>After a pull of master to get up to date, and the story branch checked out in your local, either a git merge master or git rebase master. Followed by a fresh docker image of that.</div> <pre> git status git branch git checkout master git pull origin master git branch git checkout -b temp git checkout CBY-NNNN-fe-desc git rebase master git push origin CBY-NNNN-fe-desc git status git push origin CBY-NNNN-fe-desc --force-with-lease </pre>

# Front-end Engineer Environment Setup

Scott C. Krause | 2021

*Grab a branch*

```
git clone --branch <branchname> <remote-repo-url>
```

## jq

### Command Line JSON (i think)

```
Brew install jq
```

```
Jq --version
```

## Productivity Tools

### Optional but recommended

<a href="#">Branchable</a>	You use special command lines in some repositories to implement specific workflows. Myrepos provides a <code>mr</code> command, which is a tool to manage all your version control repositories.
<a href="#">Robo3T</a>	Studio 3T - the easiest, fastest MongoDB client yet
<a href="#">PSequel</a>	PSequel provides a clean and simple interface for you to perform common PostgreSQL tasks quickly.
<a href="#">Postman</a>	Postman's features simplify each step of building an API and streamline collaboration so you can create better APIs—faster.