j-EDI Development Environment Setup for Mac & Linux

Follow these instructions to setup your development environment on Mac or Linux

All commands that are to be run in terminal can be seen by the **Bold Text**

1. Get access to the GIT source code repositories

https://github.com/Juniper/j-EDI

Contact one of the developers to obtain read access.

1. On a Mac or Linux, start the terminal application, as the shell will be used to install and configure the environment.
2. Setup SSH keys and add to Git repository for access to source code

Skip this section if your machine already has SSH keys for the Git repository.

If your Git SSH keys were on a NGSRX development VM and not on your Mac, then just copy the keys to your home directories .ssh directory and secure the files as per instructions below on creating new keys.

Generate new ssh-key to be used by git

***ssh-keygen -t rsa -C "<your\_email\_address>"***

***ssh-add ~/.ssh/id\_rsa 🡪 if you run into issues with this command, run ‘eval $(ssh-agent) first***

***cat* ~/.ssh/id\_rsa.pub** and copy the content to git Profile > RSA keys page:

Before attempting to check out the code from Git, make sure you have the correct permissions as follows, otherwise you will get permission errors and will not be able to check out the code.

-rw------- 1 jnguyen peribit 1743 Oct 9 10:07 id\_rsa

-rw-r--r-- 1 jnguyen peribit 401 Oct 9 10:07 id\_rsa.pub

chmod the permissions to match above if the permissions are different.

1. Checkout the framework and application repository from Git

**cd ~/pathtowhereyouwanttherepo**

**git clone https://github.com/Juniper/j-EDI**

1. Install Node and Node package manager (NPM)

**OPTION 1 (MAC):**

Download Mac OS X Installer from [www.nodejs.org/download](http://www.nodejs.org/download)

Execute the node installer and Node and Npm will automatically be installed.

**OPTION 1 (MAC):**

Install home brew package manager for Mac-- <http://brew.sh/> (check the homepage for more information on why homebrew is super useful)

**ruby -e "$(curl -fsSL** [**https://raw.githubusercontent.com/Homebrew/install/master/install**](https://raw.githubusercontent.com/Homebrew/install/master/install)**)"**

Install node using brew package manager

**brew install node**

To check if it worked:

**node –v** (should see something similar to v0.10.35)

**npm –v** (should see something similar to v2.1.17)

**OPTION 2 (LINUX):**

If installing in Ubuntu or similar linux distribution, consider installing linuxbrew and installing node through that (<http://www.blog.teamtreehouse.com/install-node-js-npm-linux)>

Linux Brew will be helpful later on when installing and configuring nginx.

1. Install Grunt globally

**sudo npm install –g grunt-cli**

1. Install sass version 3.2.13 (newer versions don’t generate correct css files)

On Mac, ruby is already installed so you just need to install sass

If you installed Linux Brew, ruby should be installed. If not, install ruby.

**sudo gem install sass –v 3.2.13**

**(**sudo gem install sass --version 3.2.13**)**

1. Generate CSS files

***cd ~/pathtorepo/j-EDI/slipstream***

***grunt sass:inplace***

1. Install Redis database and start redis

Download and compile redis

**cd ~**

**curl -O http://download.redis.io/redis-stable.tar.gz**

**tar -xvzf redis-stable.tar.gz**

**rm redis-stable.tar.gz**

**cd redis-stable**

**make -------------start from here --- will have to install xcode**

**sudo make install**

To tart redis server in backgrounds:

**redis-server&**

\*A redis-server process that listens on port 6379 will be running in the background\*

To kill the redis server:

**ps aux | grep redis-server**

(grab the redis-server process id)

**kill pid**

1. Start node server

**cd ~/pathtorepo/j-EDI/slipstream**

**node app.js**

1. Install nginx from a new terminal window

**OPTION 1 (MAC):**

Install home brew package manager for Mac-- <http://brew.sh/> (check the homepage for more information on why homebrew is super useful)

**ruby -e "$(curl -fsSL** [**https://raw.githubusercontent.com/Homebrew/install/master/install**](https://raw.githubusercontent.com/Homebrew/install/master/install)**)"**

Install nginx using brew package manager

**brew install nginx**

To check if it worked:

**nginx –v** (should see something similar to nginx/1.10.1)

**OPTION 2 (LINUX):**

Install linux brew for linux distributions -- Use this link or the link from step 5

<https://www.digitalocean.com/community/tutorials/how-to-install-and-use-linuxbrew-on-a-linux-vps>

**sudo apt-get install build-essential curl git m4 ruby texinfo libbz2-dev libcurl4-openssl-dev libexpat-dev libncurses-dev zlib1g-dev**

**ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/linuxbrew/go/install)"**

add the following to your .bashrc file:

export PATH="$HOME/.linuxbrew/bin:$PATH"

export MANPATH="$HOME/.linuxbrew/share/man:$MANPATH"

export INFOPATH="$HOME/.linuxbrew/share/info:$INFOPATH"

**brew install nginx** (may need to restart terminal or run brew update or brew doctor beforehand)

To check if it worked:

**nginx –v** (should see something similar to nginx/1.10.1)

1. Configure nginx to use https and point to 10.159.27.37 Space Demo Server

MAC:

**rm /usr/local/etc/nginx/nginx.conf**

LINUX:

Location of the nginx files depends on where you installed it with linux brew. Check the installation details. Will likely be similar to /.linuxbrew/etc/nginx

**rm nginx.conf**

*Sharepoint Microsoft Word on-line in view only mode displays text in Appendix A through Appendix C with extra carriage return linefeeds. If you open document using your local Mac office Microsoft word then sharepoint downloads the file to the Mac and does not add the additional characters to the file. Use the downloaded file to cut and paste the appendices.*

**Create a new file /usr/local/etc/nginx/nginx.conf**

**Cut and paste this document appendix A contents to file /usr/local/etc/nginx/nginx.conf**

1. Create /usr/local/etc/nginx/server37.crt certificate file

**Create a new file /usr/local/etc/nginx/server37.crt**

**Cut and paste this document appendix B contents to file /usr/local/etc/nginx/server37.crt**

1. Create /usr/local/etc/nginx/server37.key key file

**Create a new file /usr/local/etc/nginx/server37.key**

**Cut and paste this document appendix C contents to file /usr/local/etc/nginx/server37.key**

1. Start up nginx

**sudo nginx**

You can stop nginx at any time by typing : sudo nginx –s stop

1. From a browser open a new window to test that POC is functioning

NOTE: USE HTTPS not http – for demo purposes use http

<http://localhost:3000/assets/js/widgets/tabContainer/tests/testTabContainer.html>

Appendix A: Contents of nginx configuration file: /usr/local/etc/nginx/nginx.conf

**#user nobody;**

**worker\_processes 1;**

**#error\_log logs/error.log;**

**#error\_log logs/error.log notice;**

**#error\_log logs/error.log info;**

**#pid logs/nginx.pid;**

**events {**

**worker\_connections 1024;**

**}**

**http {**

**include mime.types;**

**default\_type application/octet-stream;**

**#log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '**

**# '$status $body\_bytes\_sent "$http\_referer" '**

**# '"$http\_user\_agent" "$http\_x\_forwarded\_for"';**

**#access\_log logs/access.log main;**

**sendfile on;**

**#tcp\_nopush on;**

**#keepalive\_timeout 0;**

**keepalive\_timeout 65;**

**#gzip on;**

**# HTTPS server**

**server {**

**listen 443 ssl;**

**server\_name localhost;**

**ssl\_certificate server37.crt;**

**ssl\_certificate\_key server37.key;**

**ssl\_session\_cache shared:SSL:1m;**

**ssl\_session\_timeout 5m;**

**ssl\_ciphers HIGH:!aNULL:!MD5;**

**ssl\_prefer\_server\_ciphers on;**

**location / {**

**proxy\_pass http://localhost:3000;**

**proxy\_http\_version 1.1;**

**proxy\_set\_header Upgrade $http\_upgrade;**

**proxy\_set\_header Connection 'upgrade';**

**proxy\_set\_header Host $host;**

**proxy\_cache\_bypass $http\_upgrade;**

**}**

**location /socket.io {**

**proxy\_pass http://localhost:3000;**

**proxy\_http\_version 1.1;**

**proxy\_set\_header Upgrade $http\_upgrade;**

**proxy\_set\_header Connection 'upgrade';**

**proxy\_set\_header Host $host;**

**proxy\_cache\_bypass $http\_upgrade;**

**}**

**location /mainui {**

**proxy\_pass https://10.159.27.37;**

**proxy\_http\_version 1.1;**

**proxy\_set\_header Upgrade $http\_upgrade;**

**proxy\_set\_header Connection 'upgrade';**

**proxy\_set\_header Host $host;**

**proxy\_cache\_bypass $http\_upgrade;**

**}**

**location /sm {**

**proxy\_pass https://10.159.27.37;**

**proxy\_http\_version 1.1;**

**proxy\_set\_header Upgrade $http\_upgrade;**

**proxy\_set\_header Connection 'upgrade';**

**proxy\_set\_header Host $host;**

**proxy\_cache\_bypass $http\_upgrade;**

**}**

**location /api {**

**proxy\_pass https://10.159.27.37;**

**proxy\_http\_version 1.1;**

**proxy\_set\_header Upgrade $http\_upgrade;**

**proxy\_set\_header Connection 'upgrade';**

**proxy\_set\_header Host $host;**

**proxy\_cache\_bypass $http\_upgrade;**

**}**

**}**

**}**

**Appendix B: server37.crt certificate /usr/local/etc/nginx/server37.crt**

-----BEGIN CERTIFICATE-----

MIIE5TCCA82gAwIBAgICRCswDQYJKoZIhvcNAQEFBQAwgaYxCzAJBgNVBAYTAi0t

MRMwEQYDVQQIEwpDYWxpZm9ybmlhMRIwEAYDVQQHEwlTdW5ueXZhbGUxHzAdBgNV

BAoTFkp1bmlwZXIgTmV0d29ya3MsIEluYy4xFDASBgNVBAsTC0p1bm9zIFNwYWNl

MRUwEwYDVQQDEwwxMC4xNTkuMjcuMzcxIDAeBgkqhkiG9w0BCQEWEXJvb3RAMTAu

MTU5LjI3LjM3MB4XDTE0MDkyOTE5MTE0MVoXDTE5MDkyOTE5MTE0MVowgaYxCzAJ

BgNVBAYTAi0tMRMwEQYDVQQIEwpDYWxpZm9ybmlhMRIwEAYDVQQHEwlTdW5ueXZh

bGUxHzAdBgNVBAoTFkp1bmlwZXIgTmV0d29ya3MsIEluYy4xFDASBgNVBAsTC0p1

bm9zIFNwYWNlMRUwEwYDVQQDEwwxMC4xNTkuMjcuMzcxIDAeBgkqhkiG9w0BCQEW

EXJvb3RAMTAuMTU5LjI3LjM3MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKC

AQEA0fOXSnnCVU5OeyBM98OPMBKcH8AuEmej67v/IMLO7D6gRp9Xf89PnEeoqqd6

n8aDo5CwsH8IoHrsLXK0xWugX8zNOglPLeS728LhjUbUyKwkHPDK/fYDLdurCoKa

5jLOgMPZrxopjoDBryfHEj2r8ludOxApLwXZ/2oKIo+rGeVxrDjlvn9nsjC9DUSc

e9/Z8eqJ/PnO1VSR15nEufysb/wnPGOoFM0JzasTheXo/niPUn8yfzfm9ilCFEEx

IDzXVtd6XuS94gaIjNOmUpr/JajKF7+Ge/ATRC/aWhL22Y87n289NiPfM87o8+pc

LdvAHNsc8rbIE1eq/G6Ur1P/+QIDAQABo4IBGTCCARUwDwYDVR0RBAgwBocECp8b

JTAdBgNVHQ4EFgQUikN0wK/kQ90fltCZ9Ou3Jxqkqb8wgdQGA1UdIwSBzDCByYAU

ikN0wK/kQ90fltCZ9Ou3Jxqkqb+hgaykgakwgaYxCzAJBgNVBAYTAi0tMRMwEQYD

VQQIEwpDYWxpZm9ybmlhMRIwEAYDVQQHEwlTdW5ueXZhbGUxHzAdBgNVBAoTFkp1

bmlwZXIgTmV0d29ya3MsIEluYy4xFDASBgNVBAsTC0p1bm9zIFNwYWNlMRUwEwYD

VQQDEwwxMC4xNTkuMjcuMzcxIDAeBgkqhkiG9w0BCQEWEXJvb3RAMTAuMTU5LjI3

LjM3ggJEKzAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBBQUAA4IBAQA0mY/TKaU1

Agz3ygsu9Sho24sfs3ycSrkreFBWDV00x4QxEhEhY+HqQq0sIiOl6++EVBgk77CC

w+D02YQrXXR5/clWfjCdXcBJMIVE4jn6NpafqvQ4tnvyXnQ67oGSNckIJr93U1v/

nxHdj9D55Sj4WMyx6aHl01wELUghLxIDSJJHhfp7NTW81VM0C4DyW4x66c4b40uv

fQLeoJRPy05TKlVn/Aok1+ChYo2w4Hm4TCk+RpLIaFczMcW+QwEgMqL72fWY2n4U

TTAuZQihuOZlan60pSsNr5sPH36TgOawG9KMgCzrO1IOWgpF4vrAtN72C+3J8v3p

wE8ioGpXtJZT

-----END CERTIFICATE-----

**Appendix C: server37.key certificate key /usr/local/etc/nginx/server37.key**

-----BEGIN RSA PRIVATE KEY-----

MIIEpAIBAAKCAQEA0fOXSnnCVU5OeyBM98OPMBKcH8AuEmej67v/IMLO7D6gRp9X

f89PnEeoqqd6n8aDo5CwsH8IoHrsLXK0xWugX8zNOglPLeS728LhjUbUyKwkHPDK

/fYDLdurCoKa5jLOgMPZrxopjoDBryfHEj2r8ludOxApLwXZ/2oKIo+rGeVxrDjl

vn9nsjC9DUSce9/Z8eqJ/PnO1VSR15nEufysb/wnPGOoFM0JzasTheXo/niPUn8y

fzfm9ilCFEExIDzXVtd6XuS94gaIjNOmUpr/JajKF7+Ge/ATRC/aWhL22Y87n289

NiPfM87o8+pcLdvAHNsc8rbIE1eq/G6Ur1P/+QIDAQABAoIBAGq8cj/Uou2Hf/Ks

uzSC4TrJW0HxC/sjYXyo+M2I/HWyaP0+aNd7Z7fF3oPT3U15ONdg6dLw4kvj8kow

tKJhUwkhL3y/2tc0ICVbcBvI1MioazujVy/BcGBR+yS+vU0mq0FfUD3rCRF/d/g6

OsSVQeNWnBq43pcfXHKzUC+95nik9n6f+fhvFwDcV6BeJdIZawzw+LJwARpJXNH8

olcmZWmVeh3+J9pw/5Rp/78kNNjS4rVNn1JVg/4NBUuPcLmzRrr7SjvUxLKzi5lq

auh1FLqAUTJRODjHdtvLgv9LShKr1YgUeY5+4p4nVxDtnJpVBRlh5o4wtvnmAgjm

ZG4mVgECgYEA/TS0/Ha0RwrbwfQdyaexAUMqFiMzxnk+h6DQB3CRmdPL8y2fljST

Z8hs4l8ShVJDw0uZpOmTp2/dAl+MH+3RtxQ2g4BT2gEBoH+GnQ0DpDbhRNXICt7G

d34OrNaha4uyg49Et04wzLVA8hytSszRpLJoAYw/xbNRCHAnU5IoOJkCgYEA1ESx

WNuKKO/WvJRd1hHDwoVCsPRmOPao7NpAco7y1iAfhFrM0Jt7T8Pb+tXvEMZZAWFE

lAYPheyqXJ9Lmc0qMmQ0pdxoKdGO2rS5miFm4NwlfhOOE3S4BJVCbRDBw97hxfC1

rLc/dBE32mjd9van+jRZkUPVRvv+fazmO1jdvmECgYAIEyg6bOau5U/AzbItfiWM

7sFm11SNXcx6sswvkI10JGb6qFfQiHiy9JdUWMfUbJ0wmzaZJcHJsy2PJ9DmK8Vm

W0SpyOvjpeiRX7/BPmsBT6a7dT3d6uDVarW5GVcogNn+jBEa72lq/dTGY9e67PE6

mV3rvrdB7MgGxtzaM20I8QKBgQCJdzsK+pJM1JXSZpsmaTbHoJmCE6Snu0XGrckS

53RoK0F963e8mRSQoZyyox1BVdKKTD2nRw2Cg7QIa7kAD5gGLlLSHxzuCCJljKiu

IneHo8o8Ddnyy6HohekQzMtPV+YK1EozYjEbr5LmRQeKSk6agJCIvM9AOy2iofgv

8usXwQKBgQDwO1lJZAsubdH3Hkt5jmKz8o8Dx1UkFUZMBzdGkby2jHtJc+zID0uu

RCf0j7twUddUd+qYtUoDogcEDca6IuyP7xLLefBlqAJf+cya3cJQOglMiIsTbn+/

mnWrNhC34+KGkRW6J2SSpOxkyO08RDdDWqwr0Yv6vU3p7V4tXXje9w==

-----END RSA PRIVATE KEY-----