**NEXEN – Front-End Setup document**

Arunkumar Madeswaran

21-Jul-16

Contents

[1 Overview 3](#_Toc456857892)

[2 Download & Install 3](#_Toc456857893)

[3 Environment Variable Setup 3](#_Toc456857894)

[4 GIT Bash setup & Launch 5](#_Toc456857895)

[5 GIT Bash NPM modules installation 6](#_Toc456857896)

[5.1 Proxy configuration – do this for every session. 6](#_Toc456857897)

[5.2 npm Module Installation 6](#_Toc456857898)

[5.3 npm configuration 6](#_Toc456857899)

[6 Nexen Code setup 6](#_Toc456857900)

[6.1 Code download 6](#_Toc456857901)

[6.2 Install Dependencies 6](#_Toc456857902)

[6.3 Build & Test 7](#_Toc456857903)

[6.4 Run NEXEN 7](#_Toc456857904)

[7 Plugin Architecture Setup 9](#_Toc456857905)

[7.1 What is Plugin Arch 9](#_Toc456857906)

[7.2 Creating Plugin Arch for Project – one time setup – done 9](#_Toc456857907)

[7.3 Connecting to Plugin Arch 9](#_Toc456857908)

[7.4 Get a (feature-release)Branch Code 10](#_Toc456857909)

[7.5 NetBeans setup 11](#_Toc456857910)

[7.5.1 Plugins to install 11](#_Toc456857911)

[7.6 How to create Protractor & Karma 11](#_Toc456857912)

[7.7 Karma code coverage 11](#_Toc456857913)

[7.7.1 Enabling Code Coverage - Todo 11](#_Toc456857914)

[7.7.2 Running Karma with Code coverage 11](#_Toc456857915)

[7.8 Grunt commands 12](#_Toc456857916)

[7.8.1 Start Server 12](#_Toc456857917)

[7.8.2 How to run Karma 12](#_Toc456857918)

[7.8.3 How to run Protractor 12](#_Toc456857919)

[8 GIT commands 12](#_Toc456857920)

[8.1 Do not check-in Netbeans files 12](#_Toc456857921)

[9 Development Process 13](#_Toc456857922)

[10 How to create Merge request 13](#_Toc456857923)

[10.1 How to trigger manual build in Jenkins 16](#_Toc456857924)

[11 Deployment in Dev Node 16](#_Toc456857925)

[12 References 20](#_Toc456857926)

[12.1 MySource Stack 20](#_Toc456857927)

[12.2 What is SonarQube? 20](#_Toc456857928)

[13 Issues & Resolution 21](#_Toc456857929)

[13.1 grunt serve – unable to find local grunt 21](#_Toc456857930)

[13.2 Pre-receive hook declined 21](#_Toc456857931)

[14 Notes 25](#_Toc456857932)

[15 Adding OMX to Nexen. 27](#_Toc456857933)

[16 Issues related to API BXP2 server 28](#_Toc456857934)

# Overview

This document gives the detailed steps for Treasuery Trade associates to do nxn-frontend local setup.

MSS Link : <https://mysourcesocial.bnymellon.net/groups/nexen-ui>

# Download & Install

Please download & Install the following software packages. For all installation in this document, right click the executable and select “BNY Mellon –Run with Elevated Rights” option.

1. Package Managers - Npm & node.js

<https://nodejs.org/dist/v4.4.6/node-v4.4.6-x64.msi>

Note: if you have older version of Node please upgrade to latest version

1. Source Repository - Git

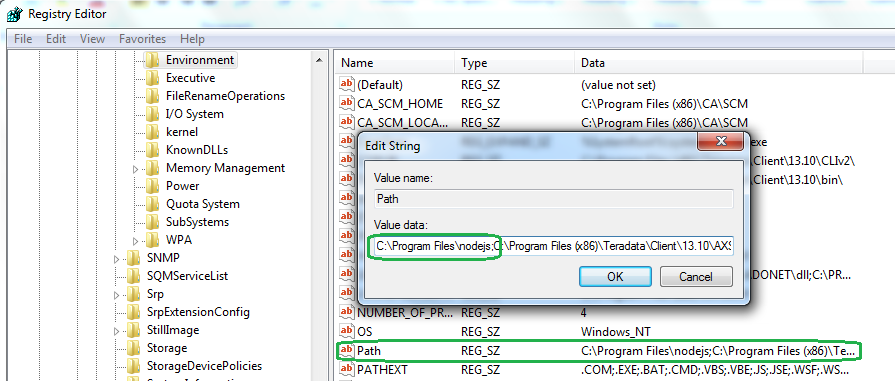
<https://github.com/msysgit/msysgit/releases/download/Git-1.9.5-preview20150319/Git-1.9.5-preview20150319.exe>

# Environment Variable Setup

1. Add node.js and Git location in environmental properties.

*Example: PATH = C:\Program Files\nodejs;C:\Program Files (x86)\Git\bin;%PATH%;*

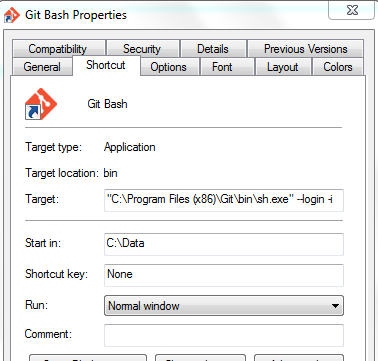
1. Note that nodejs location should be the first one in the PATH variable, other commands fail, if it is not so.
2. PATH variable under System Variables also should be appended with “*C:\Program Files\nodejs*”. Some of us might not have rights to update the System Variable.
3. Updating registry entry (Please be extra careful while doing this step.)
   1. Run 🡪 type regedit & launch in Elevated privilege mode
   2. Locate the key HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Control\Session Manager\Environment & double click the entry.
   3. Add “*C:\Program Files\nodejs;*” as the first entry in the value data text box and click ok.



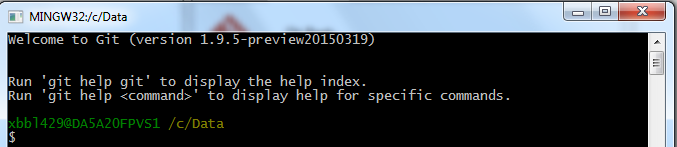
# GIT Bash setup & Launch

1. Change Git bash property : Start 🡪 Git Bash 🡪 properties (by right clicking)

Change the value of “Start in” to C:\Data



By doing the above, whenever you open your Git bash shell, the default home directly will be c:/Data



1. **Launch Git Bash**

Verify node and npm installed properly before proceeding next steps. To check that, run the below commands. (type node preceeding with two ‘-’  and version)

* 1. *node –version*  (should list the node version)
  2. *npm –version* (should list the npm version)

# GIT Bash NPM modules installation

## Proxy configuration – do this for every session.

Depending upon your region you need to add proxies to run the npm module installation command.

1. Via *npm config*

*$npm config set proxy*[*http://proxy.pershing.com:8080*](http://proxy.pershing.com:8080/)

*$npm config set https-proxy*[*http://proxy.pershing.com:8080*](http://proxy.pershing.com:8080/)

1. Via *export*

*export HTTP\_PROXY=*[*http://proxy.pershing.com:8080*](http://proxy.pershing.com:8080/)

*export HTTPS\_PROXY=*[*http://proxy.pershing.com:8080*](http://proxy.pershing.com:8080/)

## npm Module Installation

1. Install Grunt Command Line Interface Module

$ npm install --g grunt-cli

1. Install Grunt Module

$ npm install --g grunt

1. Install bower Module

$ npm install --g bower

## npm configuration

1. In DOS command prompt add nodejs, bower, git path to environmental properties.

Example: *path = c:\users\<<comit id>>\AppData\roaming\npm;c:\program files\nodejs;c:\program files\git; c:\users\<<comit id>>\AppData\roaming\npm \node\_modules\bower\bin;%path%*

1. Check grunt, grunt-cli and bower installed properly by running below version commands.

*bower --version* (should display the bower version)

*grunt --version* (should display the grunt and grunt-cli version)

# Nexen Code setup

## Code download

1. Run the below in your git bash terminal (console window) to download the nexen UI frontend source.

*$ git clone https://git.bnymellon.net/nexen/nxn-frontend.git*

               Local code location : *c:\Data\nexen\*

1. Run “cd nxn-frontend” and now your current directory will be c:/Data/nexen/nxn-frontend

## Install Dependencies

1. Run the following command in git bash

*$ npm install*

This command will install the application specific dependencies in nxn-frontend/node-modules directory.

1. Run the following command in git

*$ bower install*

This command will also install the packages mentioned in bower.json file.

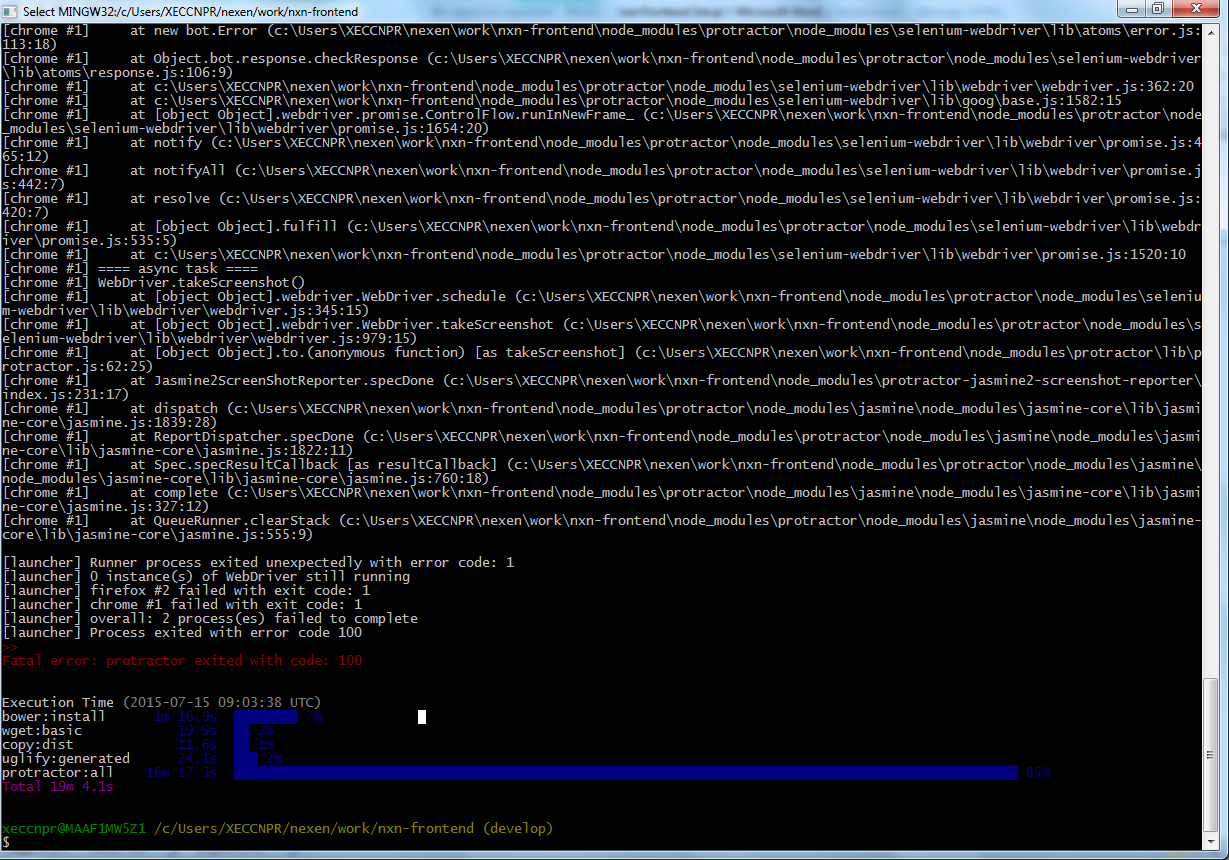
Note: If any issues (timeout issues) in above steps please setup proxy again as per Section 4.1.

## Build & Test

1. The following command will build your local nxn-frontend application including running the tests.

*$ grunt build*

1. This step might take quite a long and finally will launch browser. Automated test cases will run for quite a long time.
2. Upon completion the following screen will be displayed.

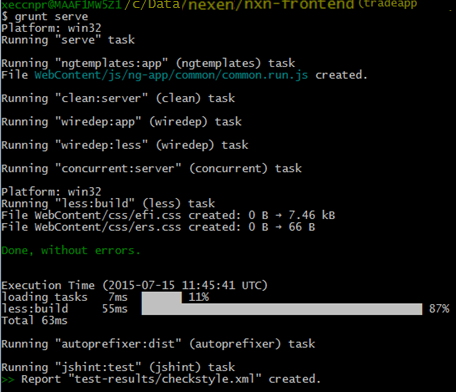


## Run NEXEN

The following command will deploy your local application in grunt server and launch the application with default port 9000.

*$ grunt serve*

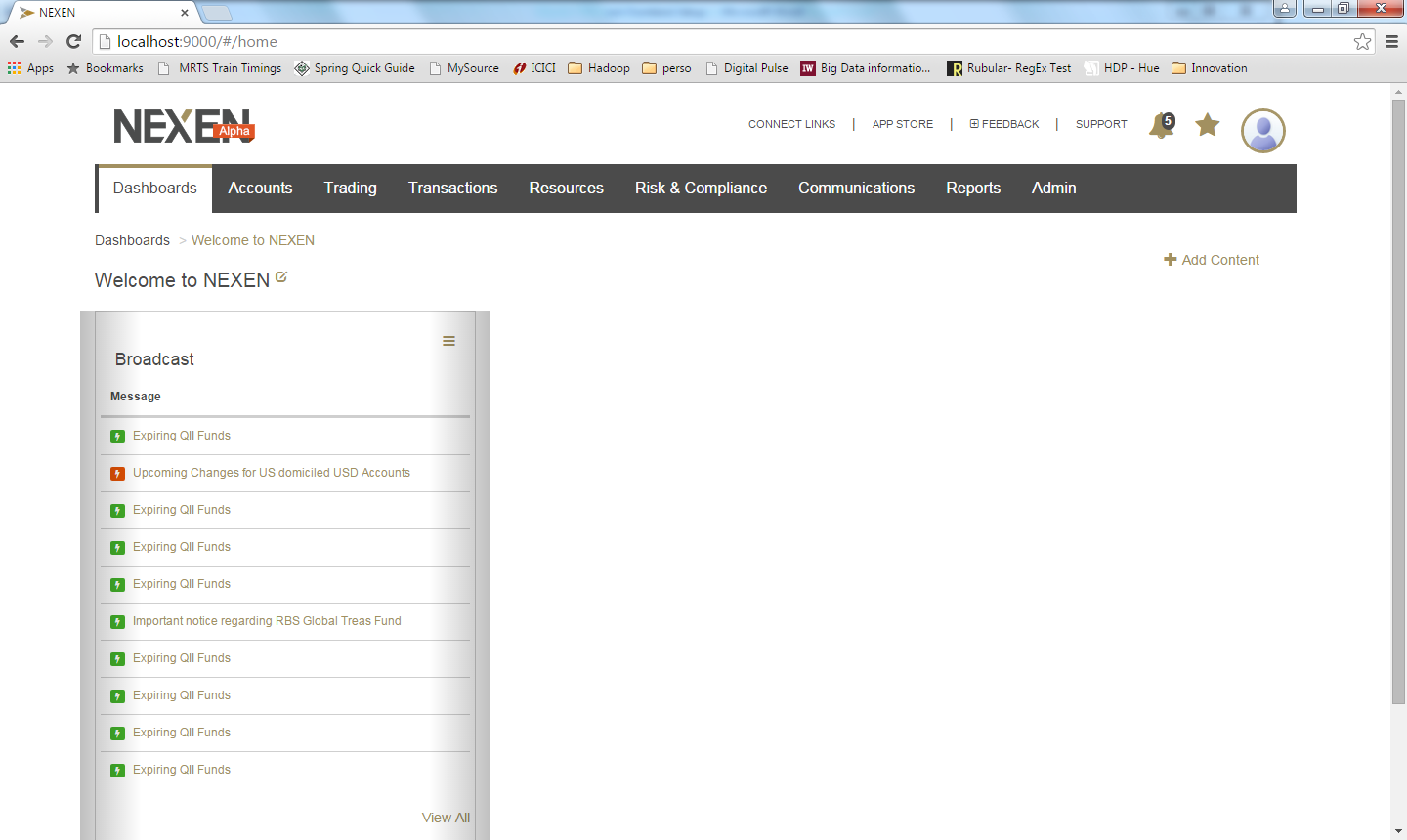
The following screen indicate a successful run of Nexen in local



A browser instance will be automatically launched with the following url

<http://localhost:9000/#/login>

Any user id /password (both 5chars length)



Anytime when Nexen prompts for PIN, enter 1234567890.

# Plugin Architecture Setup

## What is Plugin Arch

<https://mysourcesocial.bnymellon.net/groups/nexen-ui/projects/nexen-plugin-arch>

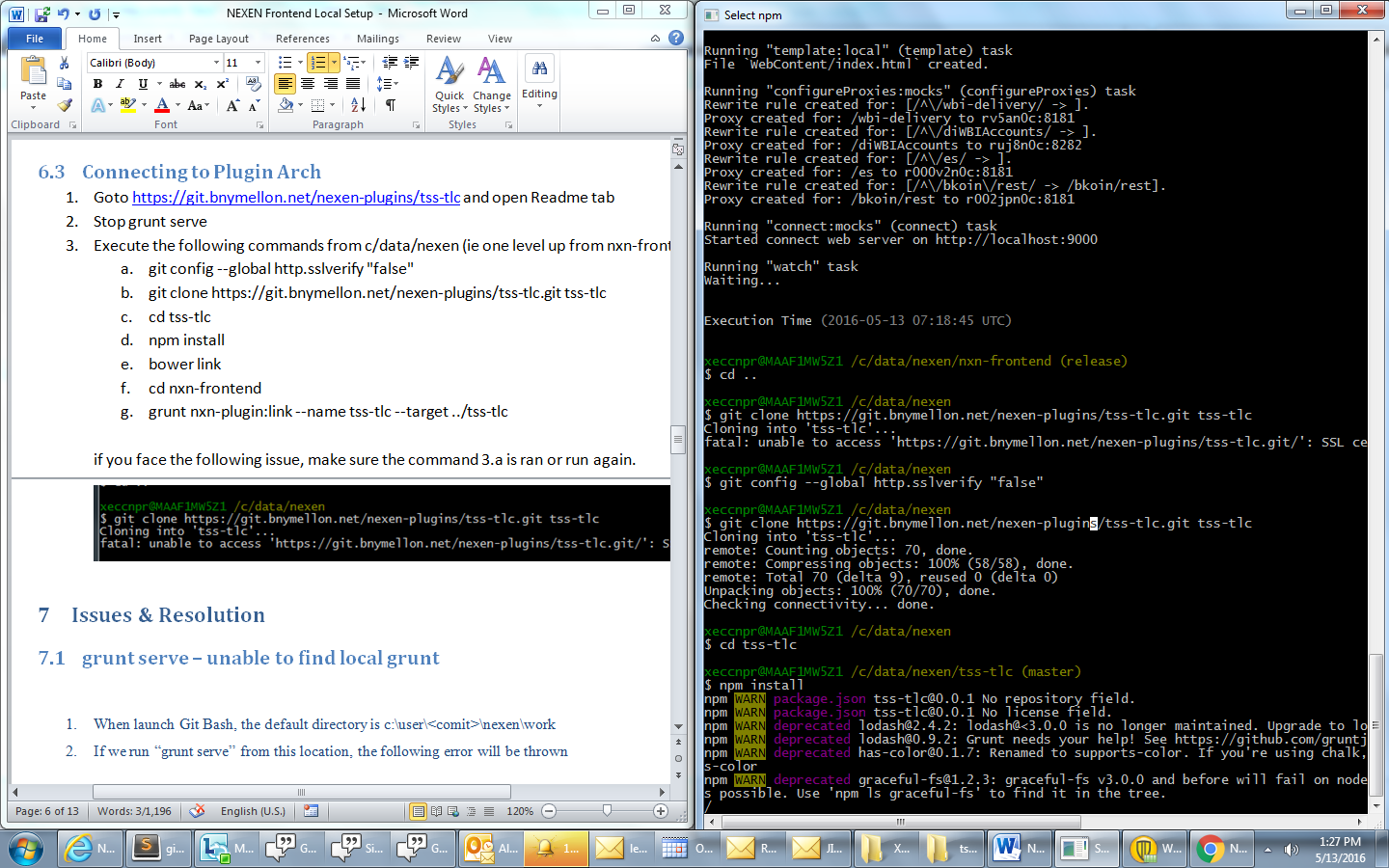
## Creating Plugin Arch for Project – one time setup – done

## Connecting to Plugin Arch

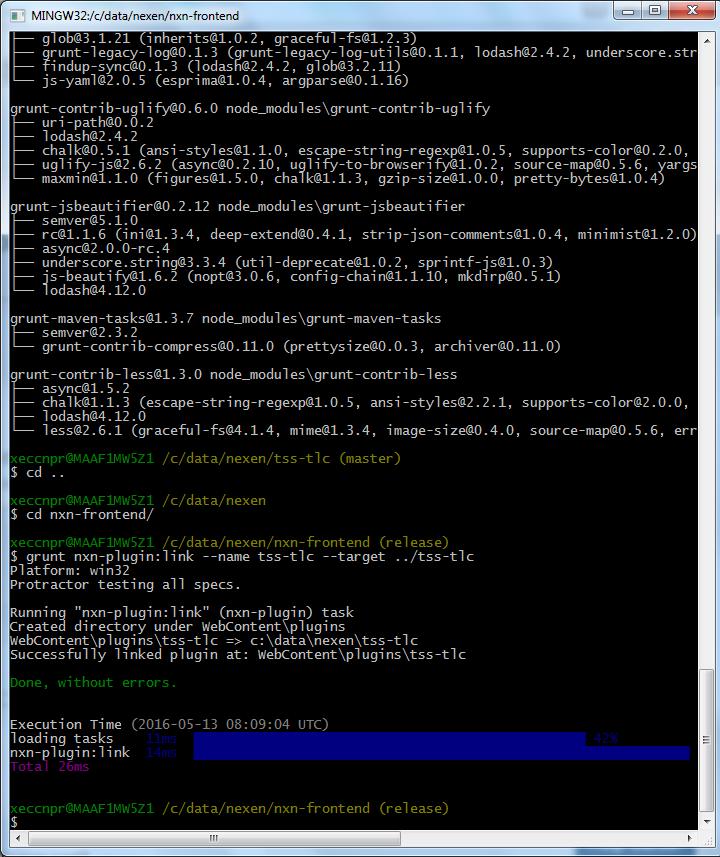
1. Goto <https://git.bnymellon.net/nexen-plugins/tss-tlc> and open Readme tab
2. Stop grunt serve (if your server is already running)
3. Execute the following commands from c/data/nexen (ie one level up from nxn-frontend)
   1. git config --global http.sslverify "false"

(tmp work around to disable SSL verify)

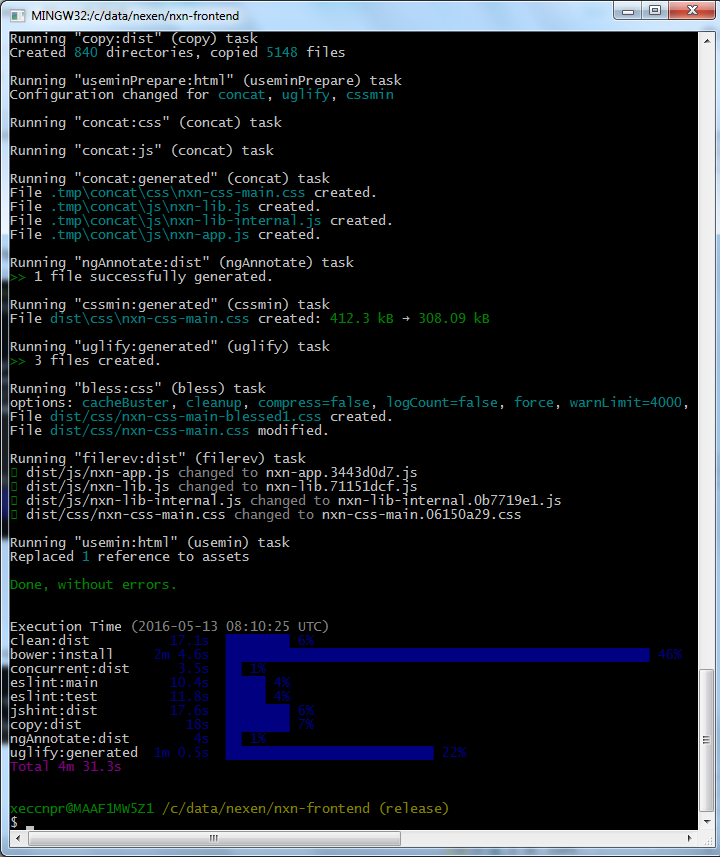
* 1. git clone https://git.bnymellon.net/nexen-plugins/tss-tlc.git tss-tlc



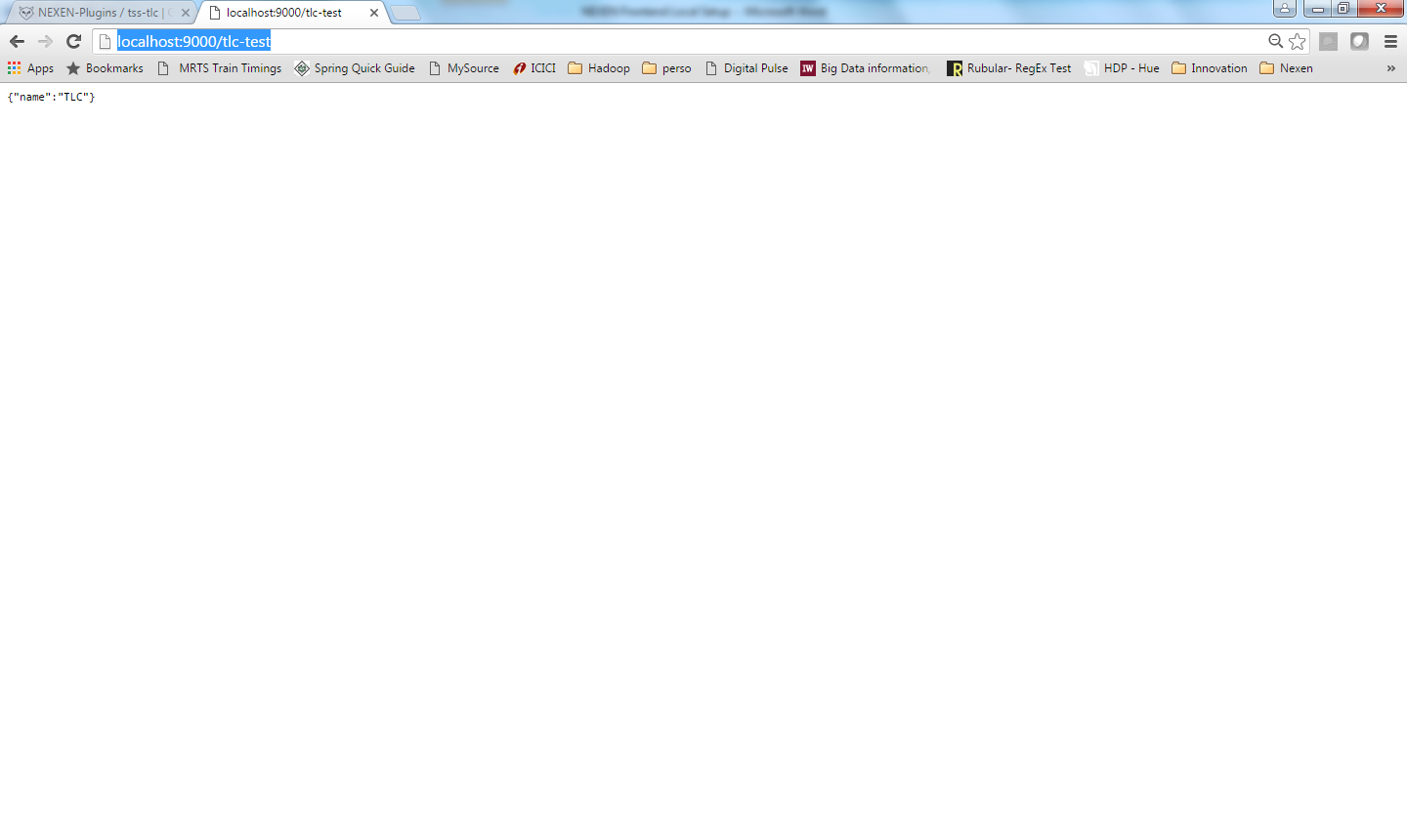
* 1. cd tss-tlc
  2. npm install
     1. bower link
  3. cd ..
  4. cd nxn-frontend
  5. grunt nxn-plugin:link --name tss-tlc --target ../tss-tlc



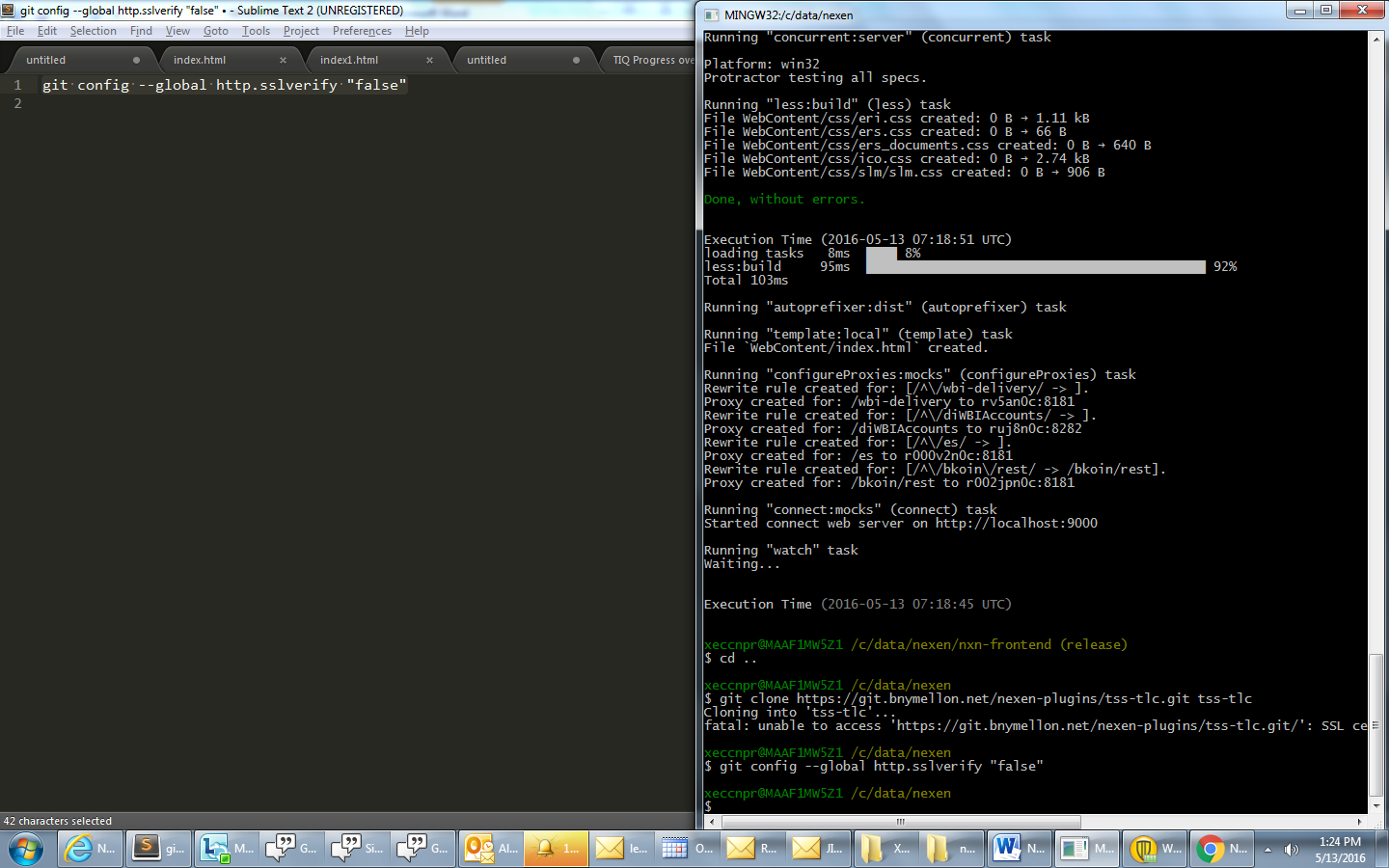
* 1. cd ..
  2. cd tss-tlc
  3. git checkout release
  4. cd ..
  5. cd nxn-frontend
  6. grunt build-notests



* 1. grunt serve
  2. To Verify the tss-tlc branch is working, Enter the following url in chrome http://localhost:9000/tlc-test

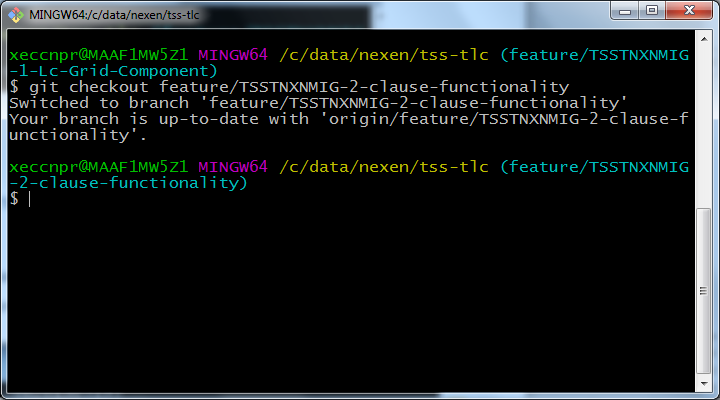


if you face the following issue, make sure the command 3.a is ran or run again.



## Get a (feature-release)Branch Code

1. Have 2 GIt Bash windows open
   1. One for “grunt serve” to run the server
   2. One for working with branches
2. Have 2 GIt Bash windows open
3. Issue the following command to switch to a branch
   1. git checkout feature/TSSTNXNMIG-2-clause-functionality



## NetBeans setup

<https://myshare.bnymellon.net/sites/FMTSTech/Treasury/knowledge_base/TBS/Documents/Architecture/nexen/native%20integration/NEXEN%20Internship/Presentations/NetBeans%20IDE%208.1%20Setup%200.1.docx>

### Plugins to install

#### ESLint

## How to create Protractor & Karma

<https://mysourcesocial.bnymellon.net/docs/DOC-13670>

## Karma code coverage

### Enabling Code Coverage - Todo

### Running Karma with Code coverage



## Grunt commands

### Start Server

* grunt serve – to start server (by default mock mode)
* grunt serve:nomock – to start server in no mock mode (connecting to API)
* grunt build – to build our code, this runs with test cases
* grunt build-notests – to build our code, this runs without test cases
* grunt format – to format code

### How to run Karma



### How to run Protractor

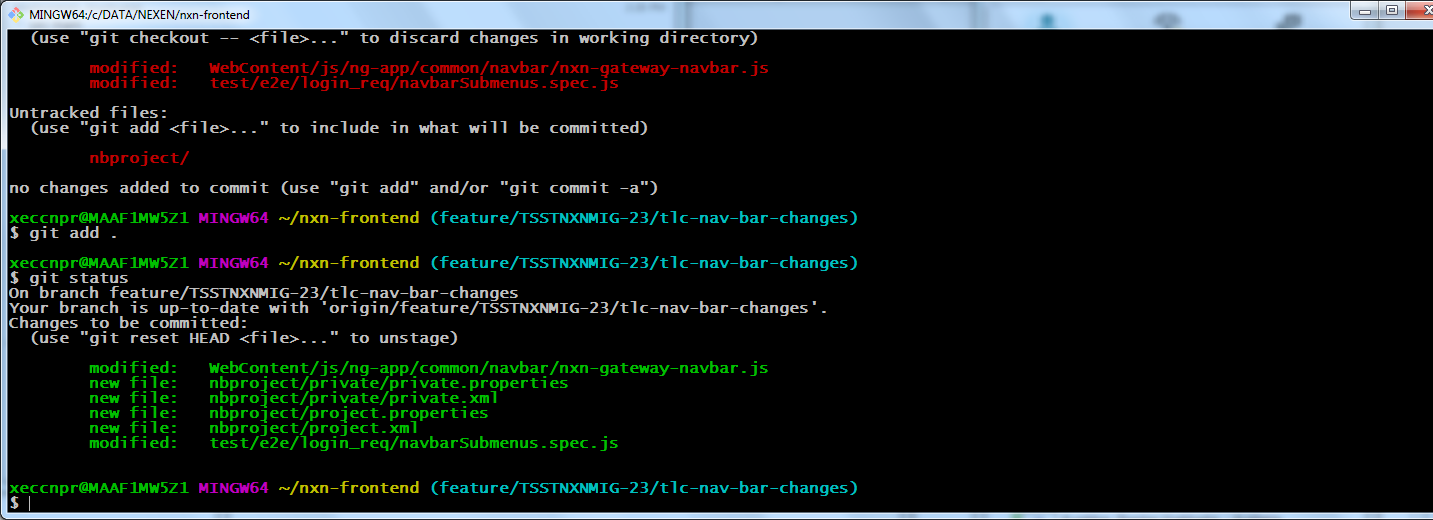


# GIT commands

<https://mysourcesocial.bnymellon.net/docs/DOC-20939>

## Do not check-in Netbeans files

Please delete the following files (nbproject\*) from your local git before check-in.

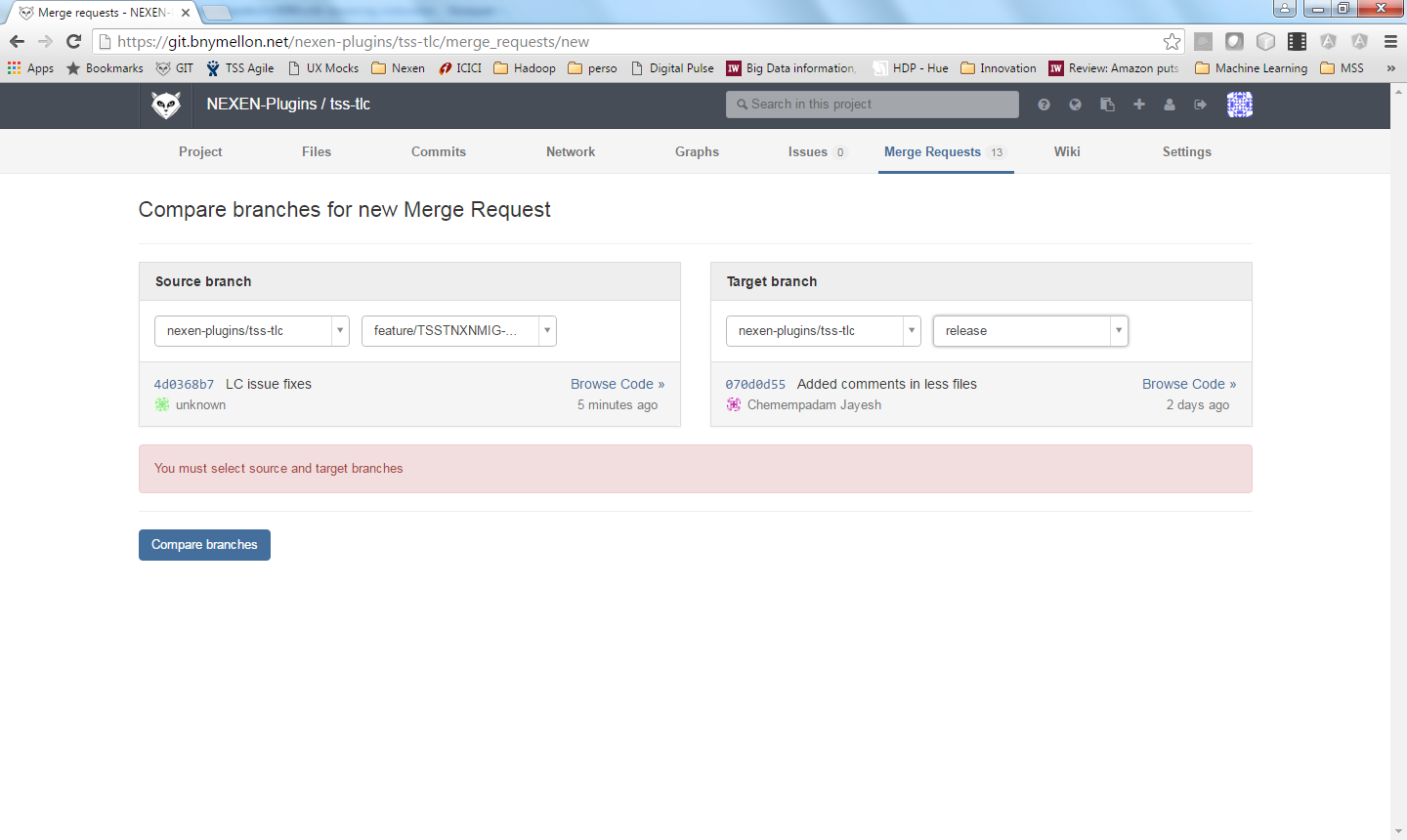


# Development Process

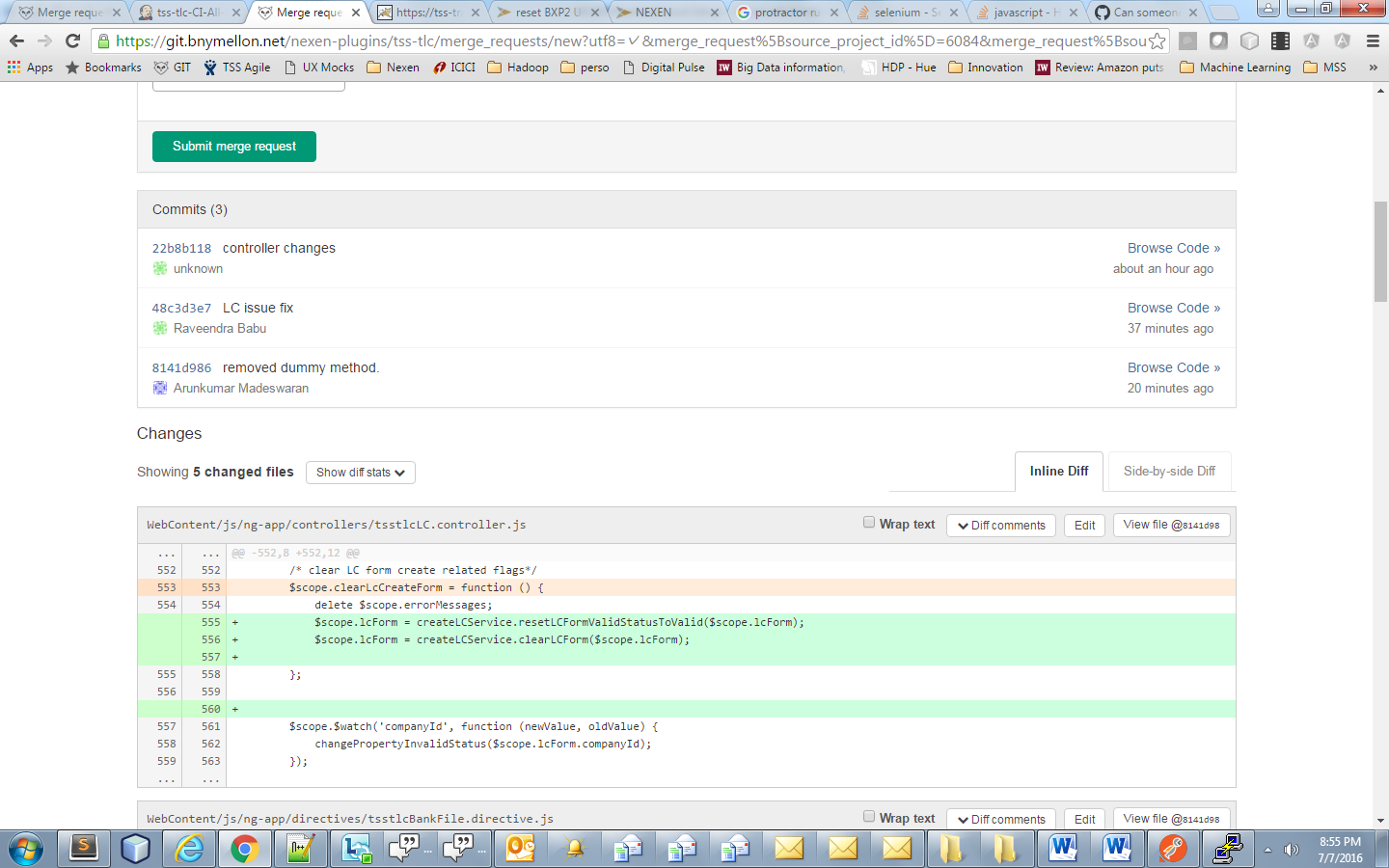
1. Create Feature Branch – dev
   1. Development
   2. E2E – Protractor
   3. Functional - Karma
2. Merge request to release - Dev & SM approves (snapshot of release + individual ticket)
   1. Merge Build Should succeed along with Test cases
3. Optional: Create Integration Branch (for all tickets in sprint) = snapshot of release + Branches
4. Peer Review
5. GateKeeper Review
6. Deploy to Dev node using Deployment Co-ordinator.
7. CT testing and approval
8. Merge with Release

# How to create Merge request

1. Click on Merge request 🡪 New Merge Request
2. Select Source Branch and Traget Branch as tss-tlc & release
3. Click Compare Branches



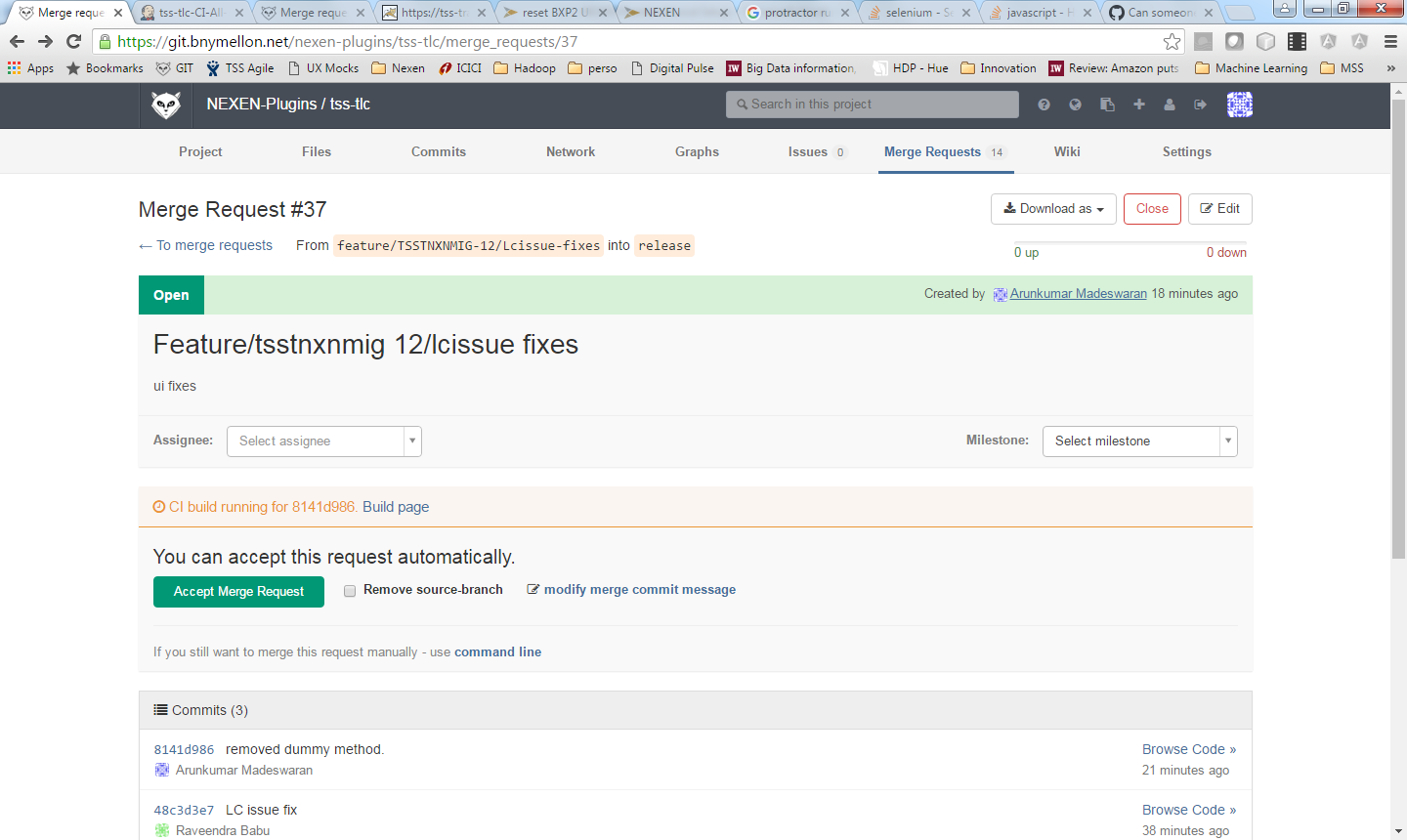
1. All the changed files will be listed with difference, please review and make sure you are adding only the required changes
2. Click Submit Merge Request.



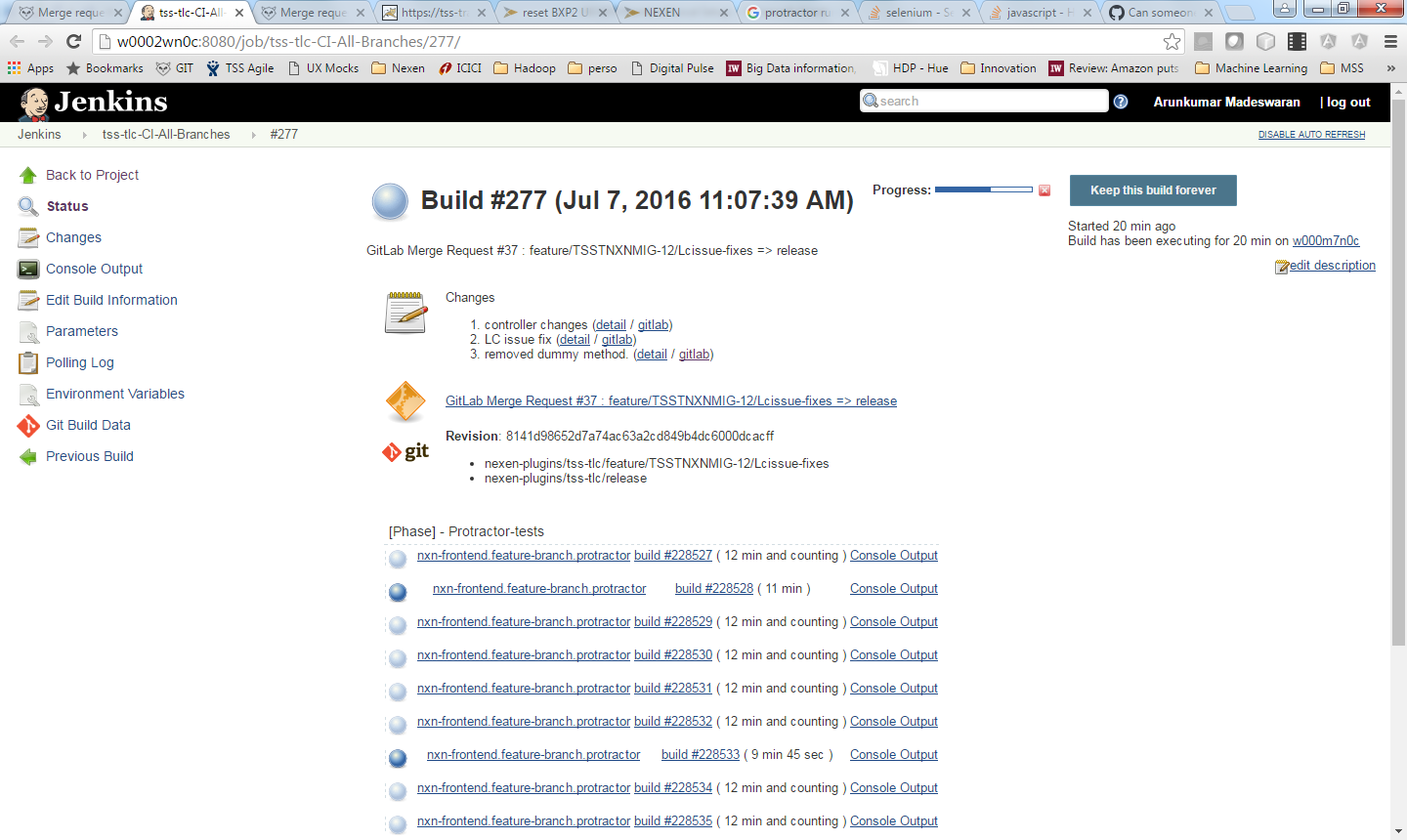
1. “CI Build Running for xxxxx” will be displayed.

Note: sometimes you might get “CI Build pending xxxxxx”, refresh this page after few min, the status should change to “Running”, else there might be a Jenkins issue.

1. Click on the “Build Page” hyperlink to check the status of the build.



1. Builds generally run for 30 min to 1.5 hours timespan. It basically runs all our test cases
2. Click on “Enable Auto Refresh”option (to right corner)



## How to trigger manual build in Jenkins

Automated build fails due to various reasons (environmental, dependencies, etc…), in these cases we can trigger the manual build.

Manual Build URL: <http://w0002wn0c:8080/job/tss-tlc-CI-All-Branches/build?delay=0sec>

**gitlabSourceBranch:** feature/TSSTNXNMIG-387-DEMO-ISSUE-FIX-INTERNAL-SERVER (your branch that needs a manaul build)

**gitlabTargetBranch:** release (leave the default value)

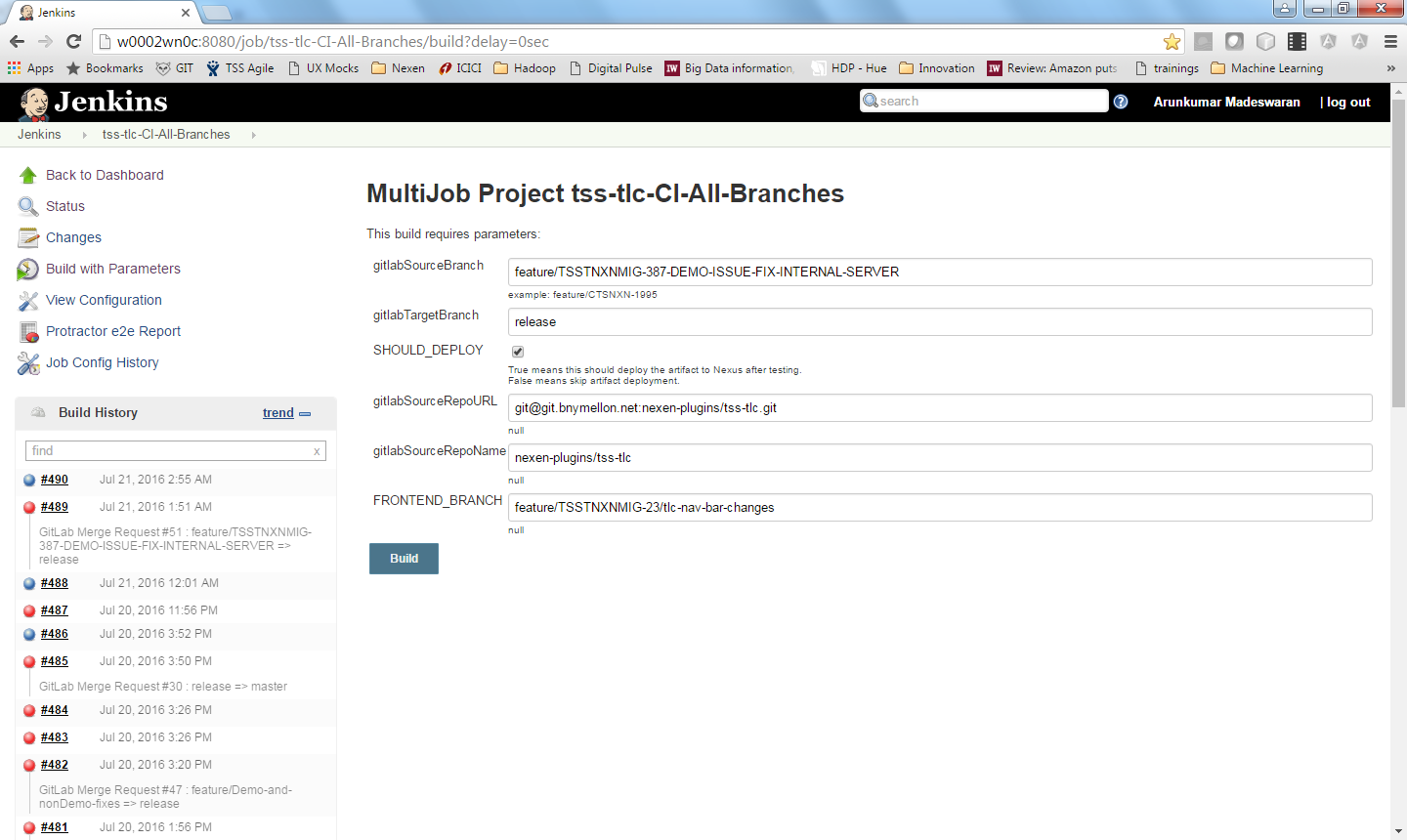
**SHOULD\_DEPLOY:** checked (leave the default value)

**gitlabSourceRepoURL:** (leave the default value)

**gitlabSourceRepoName:** nexen-plugins/tss-tlc

**FRONTEND\_BRANCH:**

this is usually "release", since our Navbar changes are not yet merged into nxn-frontend release, we are giving our navbar branch which is in nxn-frontend repo (http://w0002wn0c:8080/job/tss-tlc-CI-All-Branches/490/).



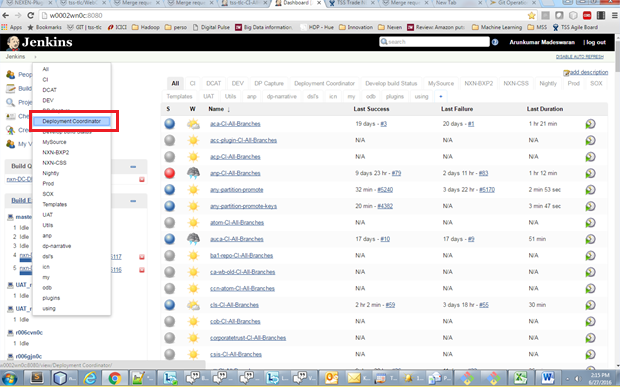
# Deployment in Dev Node

NEXEN Documentation: <https://mysourcesocial.bnymellon.net/docs/DOC-31952>

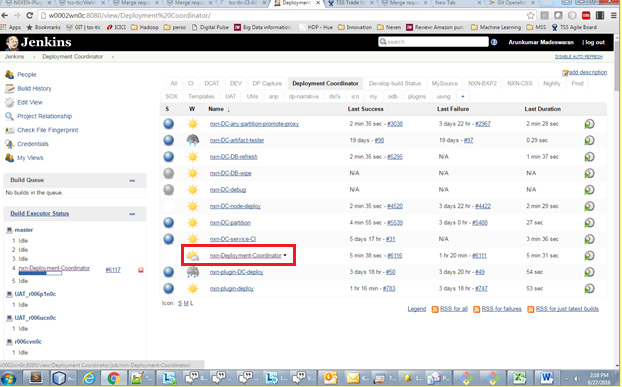
1. Select the Merge Request which is build completed successfully.



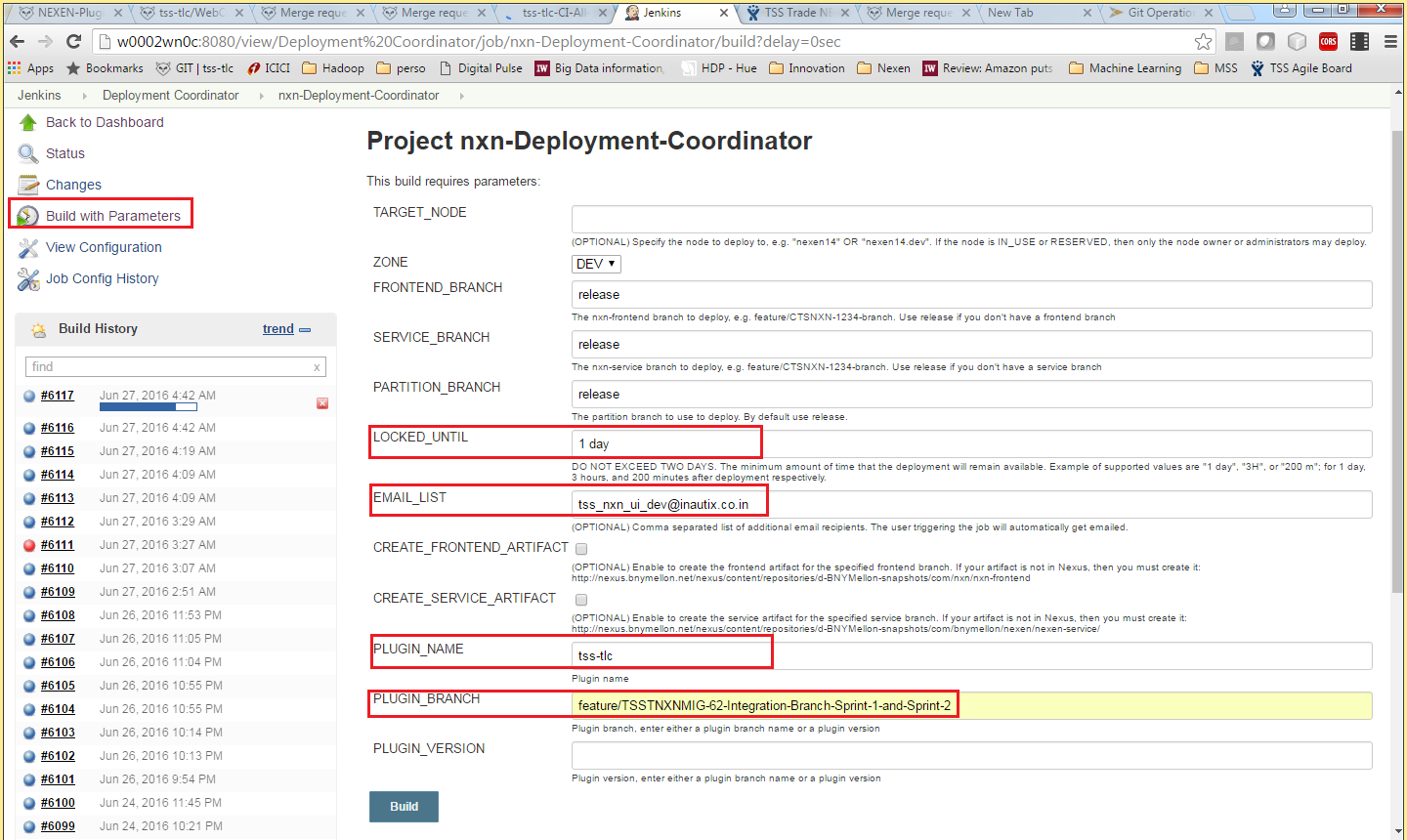
1. Open Jenkins <http://w0002wn0c:8080/> & select Deployment Co-ordinator



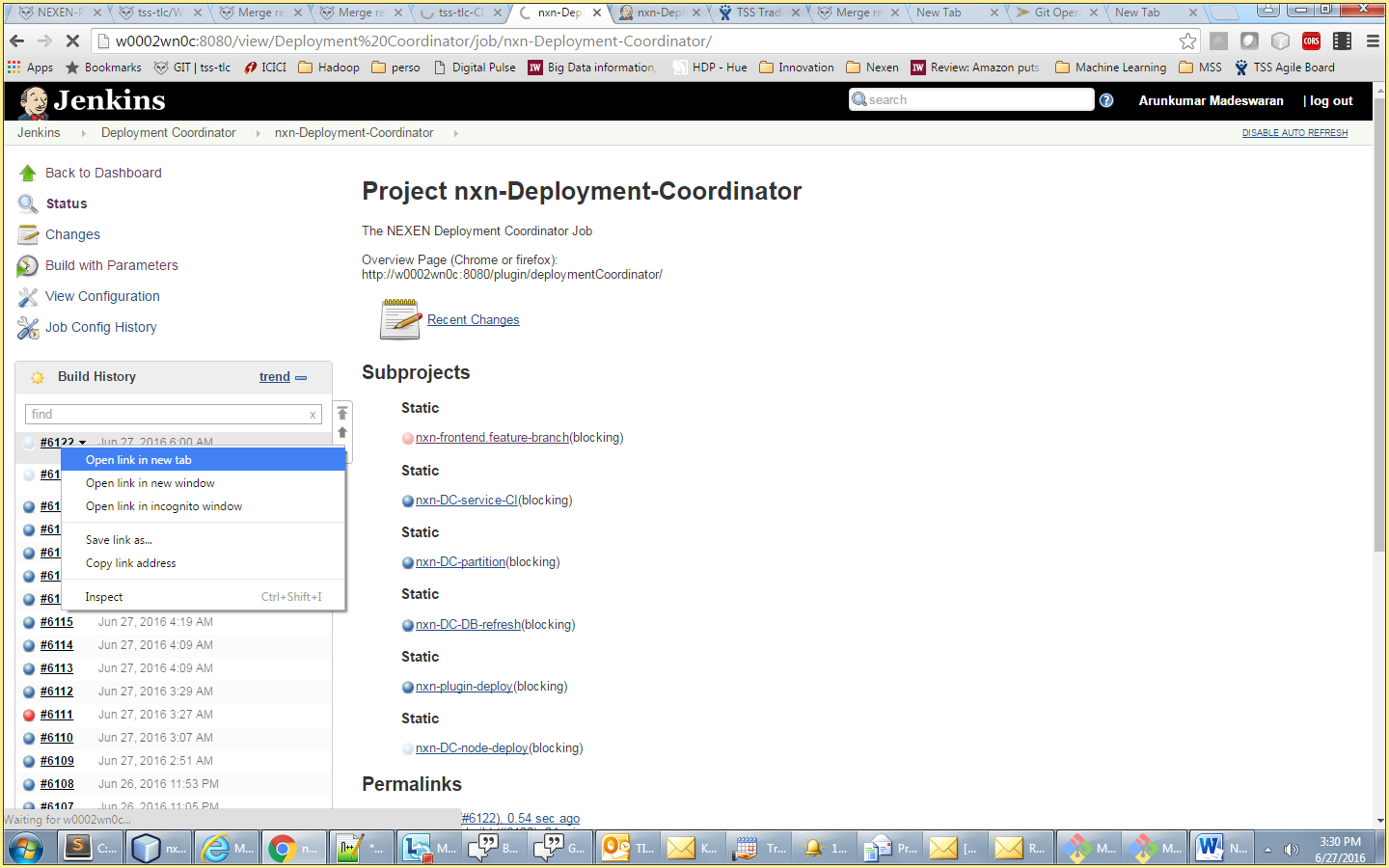
1. Select nxn-Deployment Coordinator



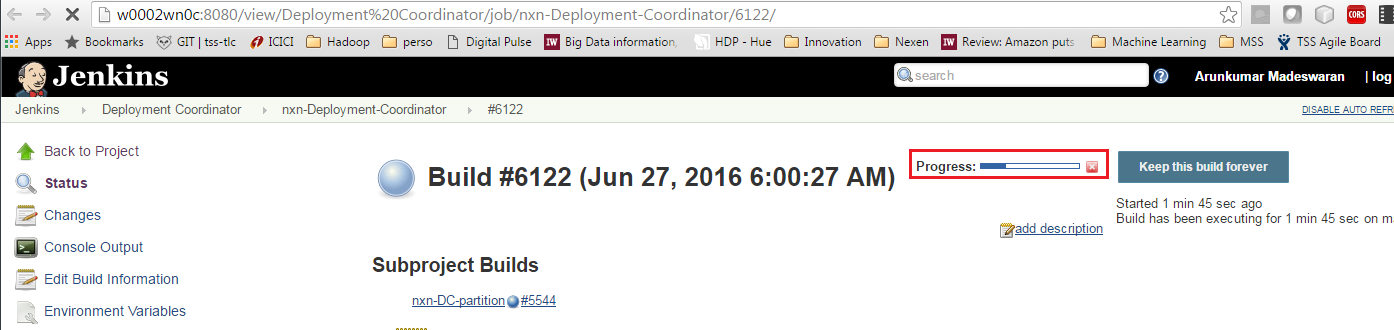
1. Select “Build with Parameters” and fill in the build parameters



1. LOCKED\_UNTIL: It defines the time until when your code will be availabale on the node. By default, it expires in 1 day and you need to extend, if required.
2. EMAIL\_LIST: It defines the email ids, which get the notification, once the deployment is completed. By default, the one who deploys gets the email notification.
3. PLUGIN\_NAME: Here, you need to give your plugin name.
4. PLUGIN\_BRANCH: Here, you give the plugin branch, which needs to be deployed
5. Click Build and open the build request ID (Build History panel in left) in a new tab.

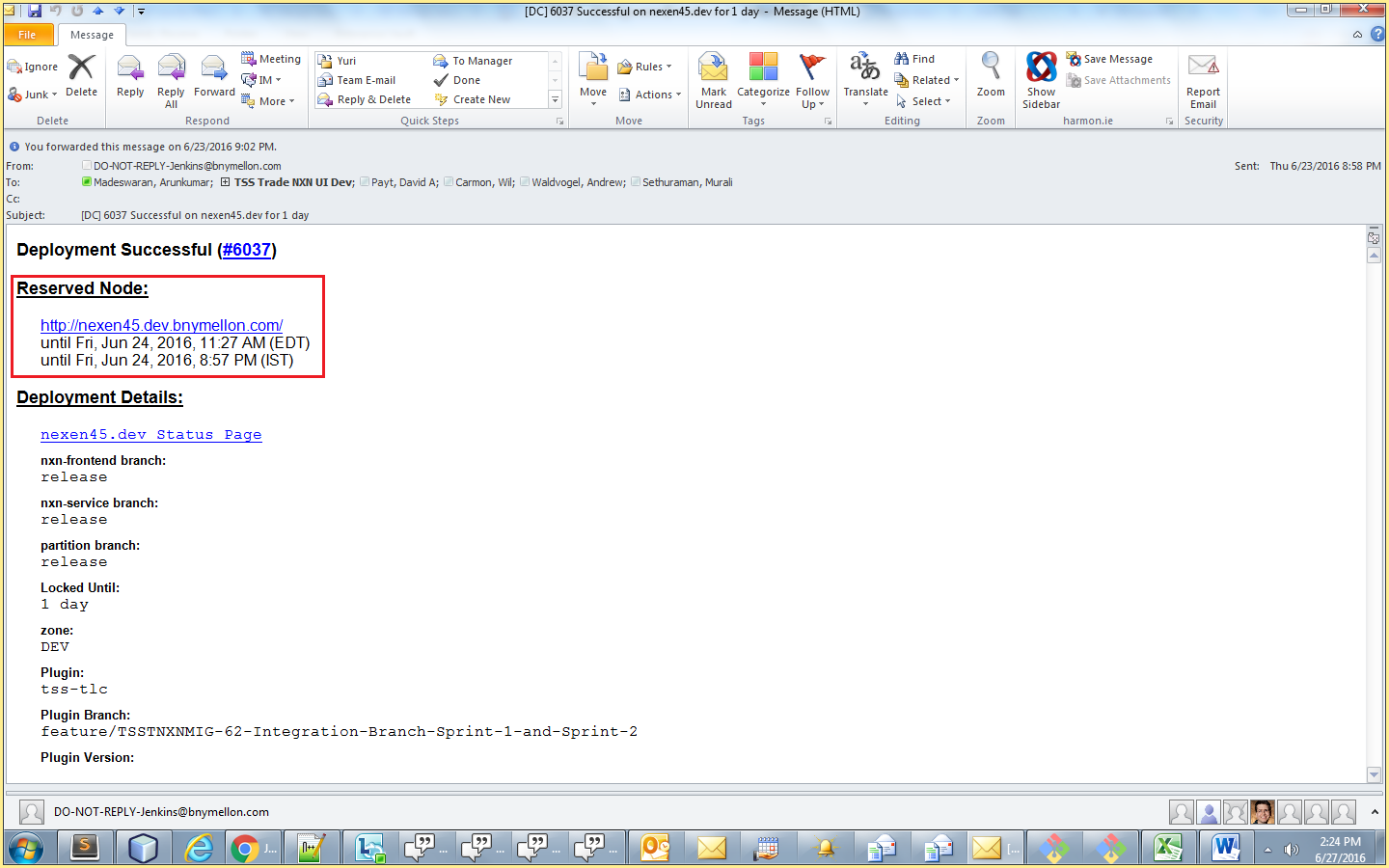


1. Progress of the deployment can be tracked in this page. Upon successful deployment, details will be emailed.



1. Sample Deployment email

The notification email gives you the details such as the reserved node & the timeline until which the node is reserved for you.



# References

## MySource Stack

MySource Stack is an internal Stack Overflow collaboration tool that can be quite helpful in socializing FAQs.

<https://mysourcestack.bnymellon.net/>

## What is SonarQube?

TSS TLC can also be configured in Sonar Qube to automate code review.

NEXEN Frontend already plugged in Sonar Qube (<http://r005een0c.bnymellon.net:9000/overview?id=nexen-nxn-frontend>).

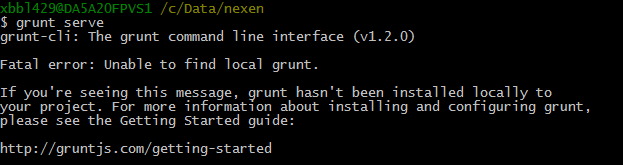
SonarQube does static analysis on code to find errors and measure quality. See the SonarQube home page at <http://www.sonarqube.org/>.

1. [How to Use SonarQube](https://mysourcesocial.bnymellon.net/docs/DOC-24654)
2. [SonarQube in Intellij IDEA](https://mysourcesocial.bnymellon.net/docs/DOC-52194)

# Issues & Resolution

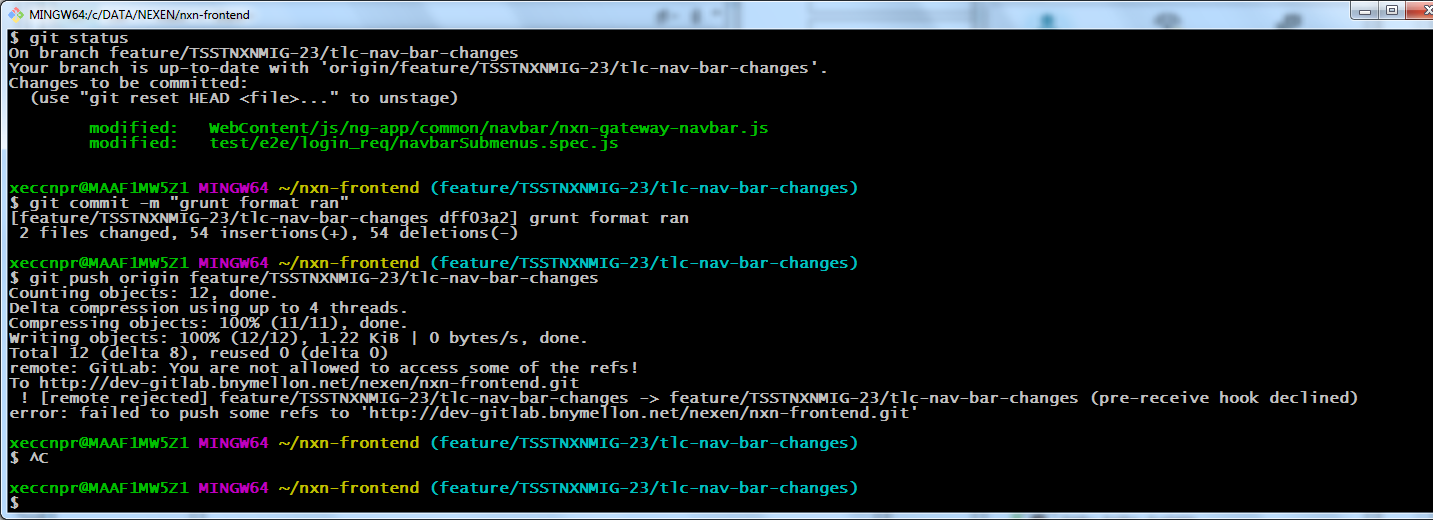
## grunt serve – unable to find local grunt

1. When launch Git Bash, the default directory is c:\user\<comit>\nexen\work
2. If we run “grunt serve” from this location, the following error will be thrown

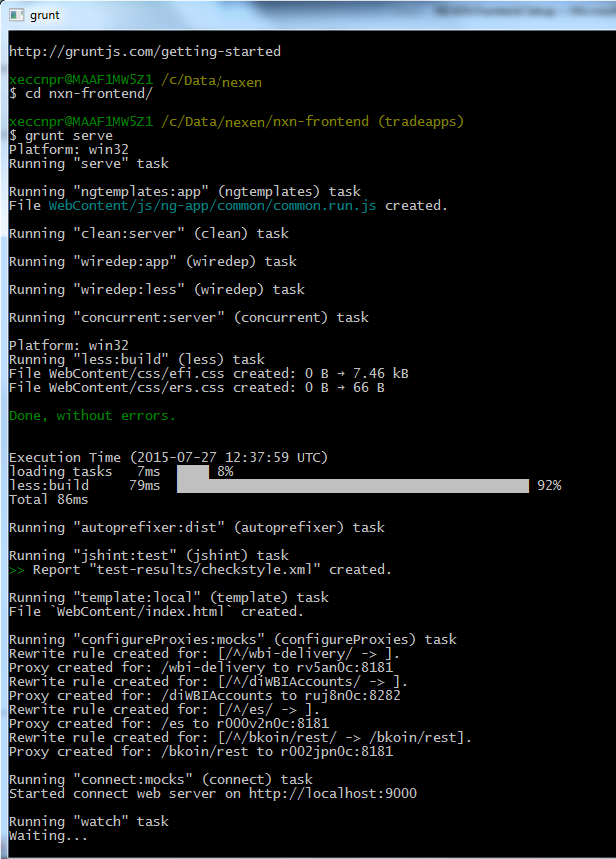


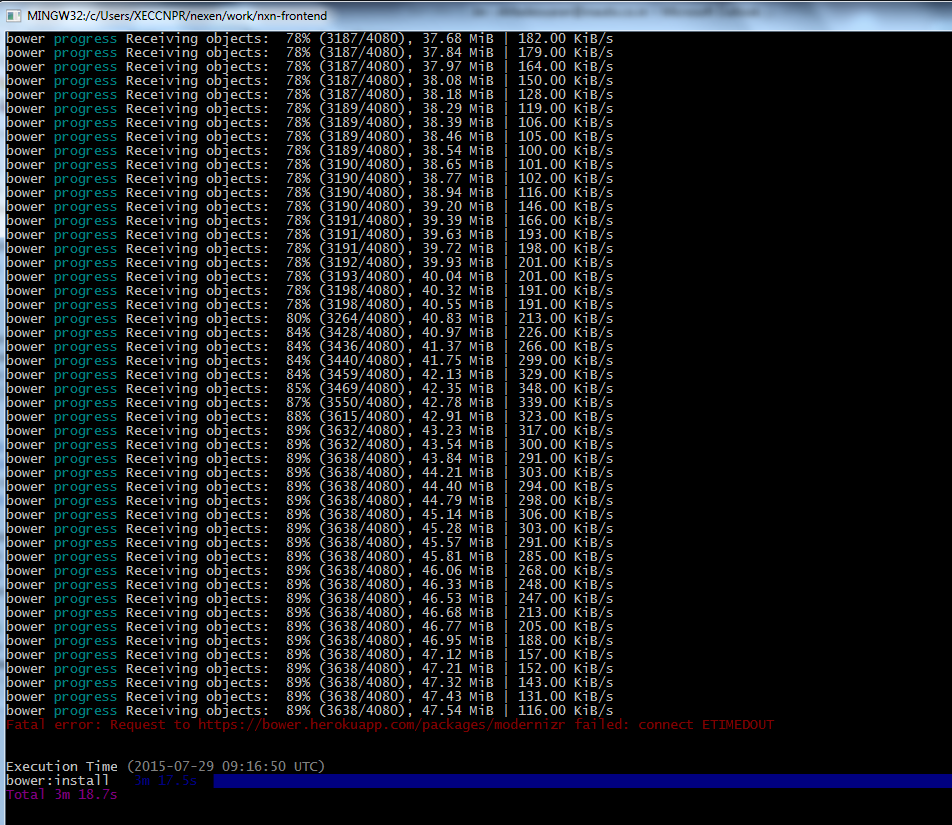
## Pre-receive hook declined

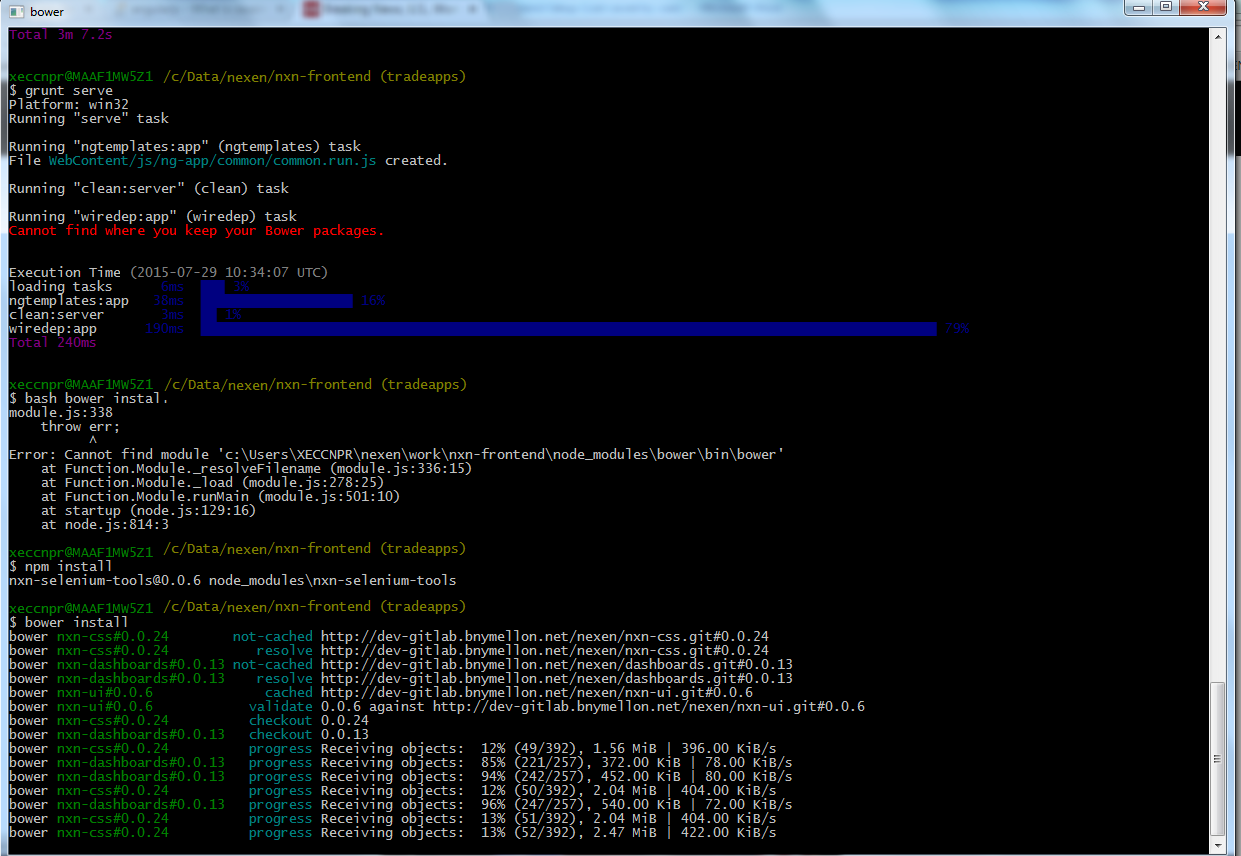
While doing “git push….”, No access to nxn-frontend will cause this issue. Please email nxn-core for git access to their project



1. Change directory to “nxn-frontend” and then “grunt serve” will work







# Notes

1. If you copy the command from the word document, you make copy the special char also.
2. https and git: you need to run the following command because bank only allow https

git config --global url."https://".insteadOf git://

1. add proxy for omx:

Gruntfile.js

var test\_proxies = [

{

context: '/omxws',

host: 'localhost',

port: 8181,

https: false,

xforward: false,

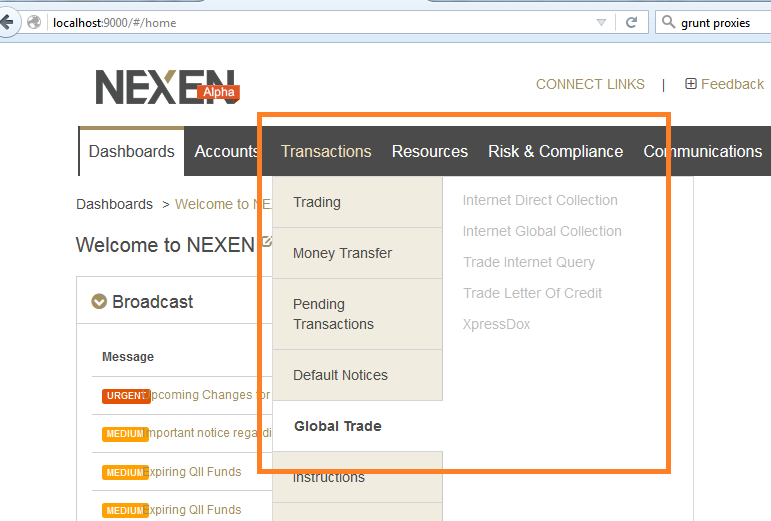
headers: {

'SMUSER':'tbstest'

},

},

1. Change menu:



nxn-gateway-navbar.js

{

"id": 15,

"name": "Global Trade",

"second\_level": [

{"id": 54, "name": "Internet Direct Collection"},

{"id": 55, "name": "Internet Global Collection"},

{"id": 56, "name": "Trade Internet Query"},

{"id": 57, "name": "Trade Letter Of Credit"},

{"id": 58, "name": "XpressDox"},

]

},

# Adding OMX to Nexen.

1. Dowload the code from GIT repository - [**file:////wsomdavigc01/git.data/omx**](file://wsomdavigc01/git.data/omx)

1. Create a new directory under WebContent\js\jg-app called omx.

2. Add the omx.module.js file inside it.

3. Create a new directory under WebContent\js\ng-app called omx.

4. Add the TlcOmx.Html file inside it.

5. Add following code : 'nxn.omx’ as an element of the dependency array into the WebContentent\js\ng-app\app.js file as a dependency for the module. This tells your app that you would like that nxn.omx module to be used.

6. open up the following file: mocks\pagecontext\pageContextMock.js. this is the mock page context object for our local development environment. There is a pageContext variable. place the following as the last item at the end of the navbar array:

**"434": {**

**"id": 434,**

**"description": "omx",**

**"enabled": true,**

**"state": [**

**"omx"**

**]**

**},**

1. Open the /WebContent/js/ng-app/common/navbar/nxn-gateway-navbar.js file. Locate in the nvbar variable the object with the name: "MANAGE" as a top level item (id: 15). Add the following line as an item in the "second\_level" array:

**{**

**"id": 54,**

**"title": "Letters of Credit",**

**"items": [**

**{"id": 434, "name": "omx","state": "omx"},**

**{"id": 56, "name": "Trade Workstation"},**

**{"id": 57, "name": "Xpress Dox"}**

**]**

**},**

1. Refresh your browser and look under Transactions>Manage > Omx. If you click on the link it will show you the contents of TLCOmx.html partial.

<http://da5acdvyvq1:8181/omxwsapi/v1/clauses/>

# Issues related to API BXP2 server

The issues and resolution for API dev server (r009hsn0c) are available in [BXP server r009hsn0c Issues](https://share.ams.bnymellon.net/sites/FMTSTech/Treasury/knowledge_base/TBS/Documents/Architecture/nexen/native%20integration/NEXEN%20Internship/Presentations/BXP%20server%20r009hsn0c%20Issues.docx)