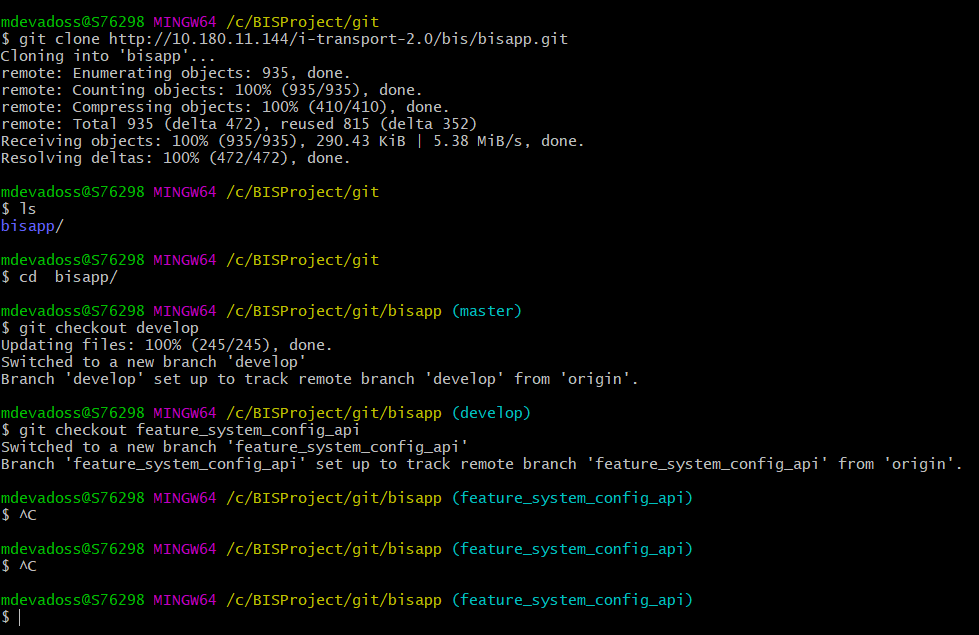
# BI Project

## Main Project Folder setup

1. Create Main Project Folder
   1. Create **Git** Folder
      1. Inside the folder Right click and open (Git Bash Here)
      2. 
      3. $ git clone <http://10.180.11.144/i-transport-2.0/bis/bisapp.git>
      4. $ ls
      5. $ cd bisapp/
      6. $ git checkout develop
      7. $ git checkout feature\_system\_config\_api

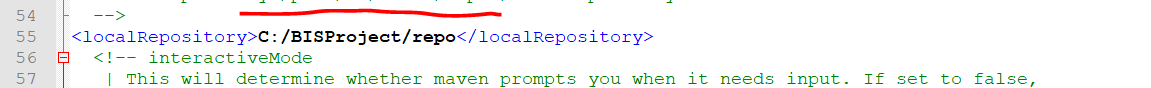


* 1. Create **repo** Folder
  2. Create **workspace** folder
     1. Point this workspace when your open a eclipse

## MAVEN File change

Change mavan/config/settings.xml

### Change settings.xml



Write project location and create repo folder

### Setup in Eclipse

1. Give Maven path

Eclipse - windows - preference - maven - Installation

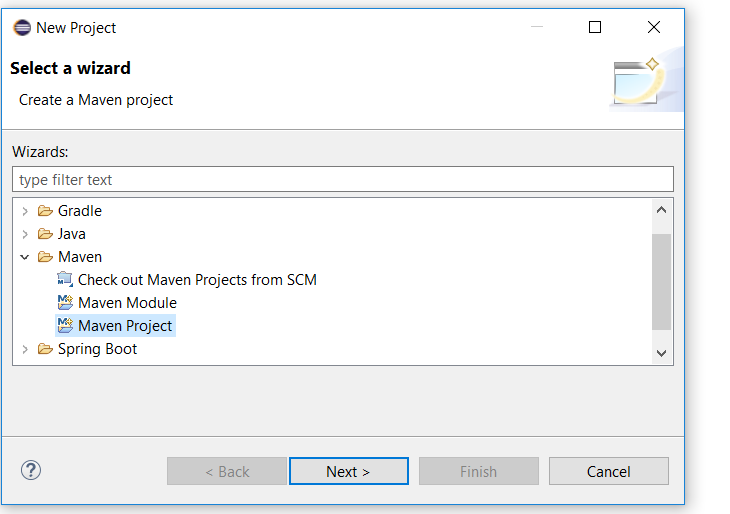
1. User setting location

Eclipse - windows - preference - maven - usersetting

Point the location “settings.xml”

# Eclipse Project folder setup

Eclipse - New project



Maven existing project - select root folder - (bis)

## Eclipse - Error during build

Maven compile a project in Eclipse IDE, but hits the following error messages :

$ mvn clean compile

[ERROR] COMPILATION ERROR :

[INFO] -------------------------------------------------------------

[ERROR] No compiler is provided in this environment. Perhaps you are running on a JRE rather than a JDK?

[INFO] 1 error

[INFO] -------------------------------------------------------------

[INFO] ------------------------------------------------------------------------

[INFO] BUILD FAILURE

[INFO] ------------------------------------------------------------------------

[INFO] Total time: 0.859 s

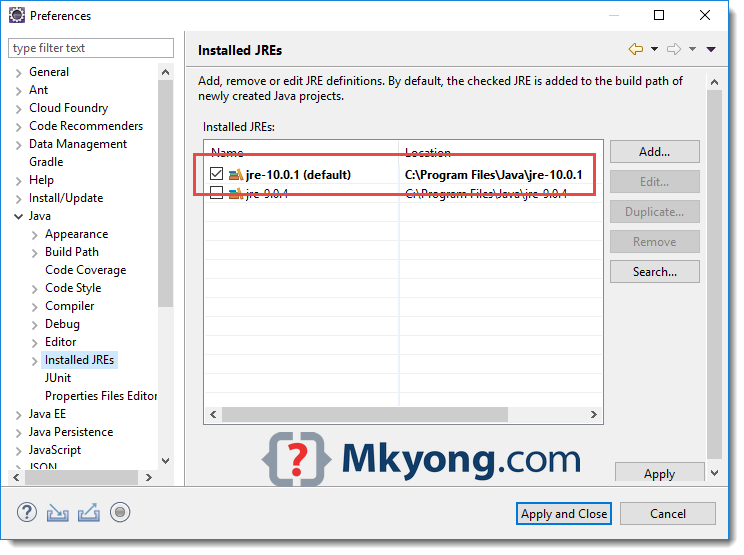
[INFO] Finished at: 2018-10-03T15:56:43+08:00

[INFO] ------------------------------------------------------------------------

Copy

P.S Tested with Maven 3.5.3 and Eclipse SimRel 2018-09

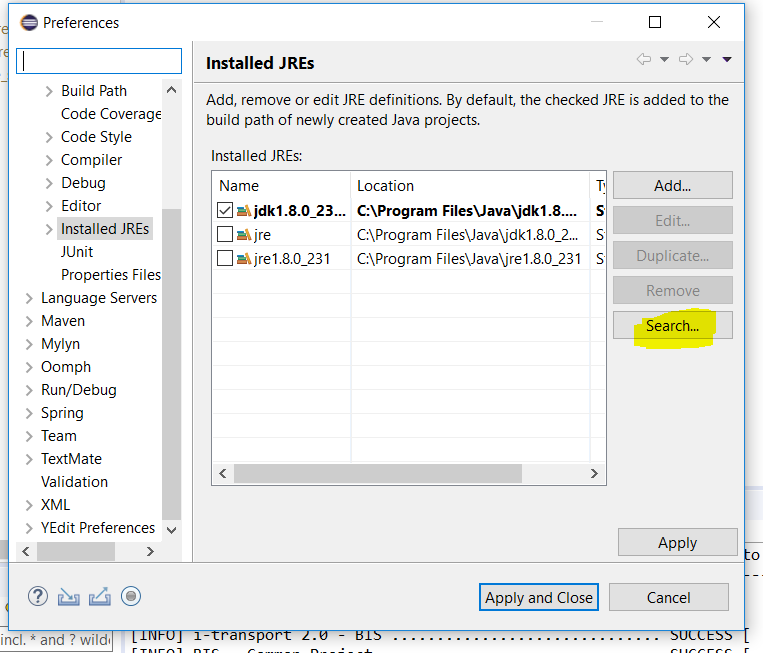
By default, Eclipse configured the JRE automatically.



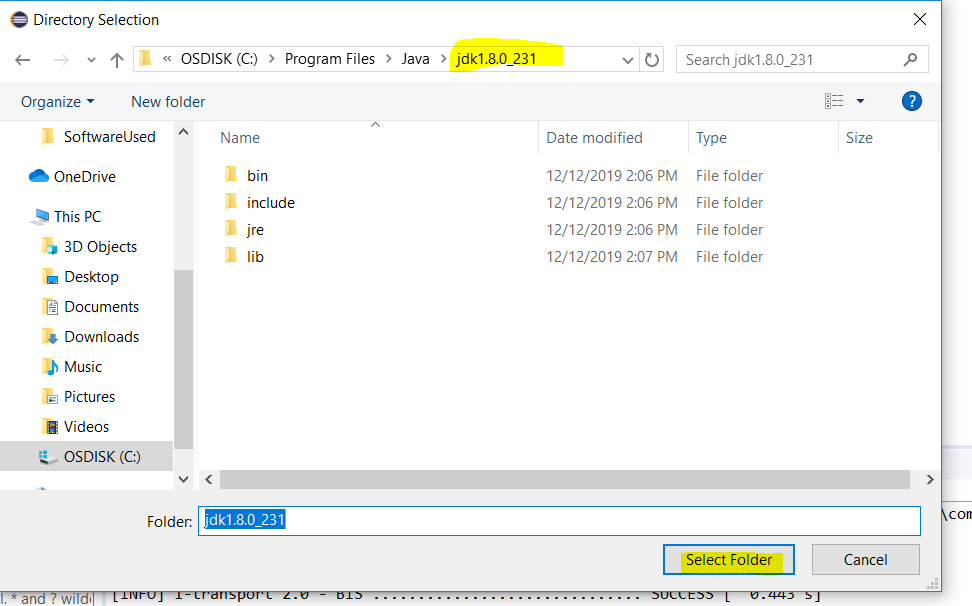
## Solution

Maven needs JDK to compile project. Make sure you have installed/downloaded JDK and add it into Eclipse IDE as installed JRE.

1. Windows -> Preferences -> Java -> Installed JREs -> Add… or Search , select a JDK folder.



**Use Search option to select a “JDK“ folder**



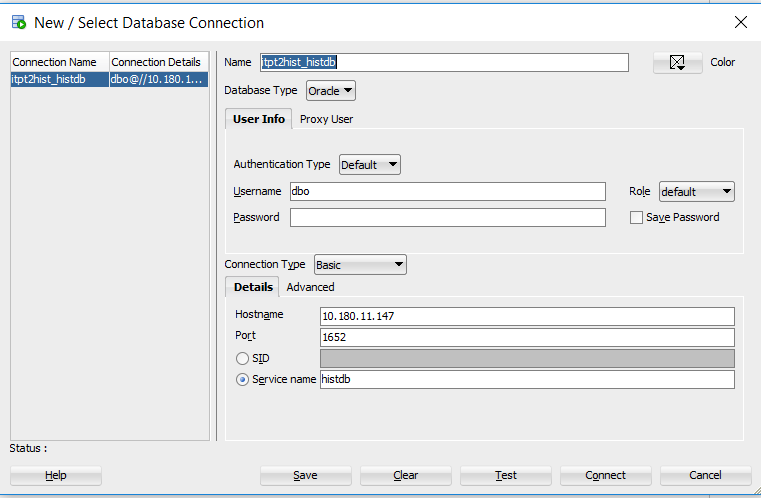
# Database setup

  url: jdbc:oracle:thin:@10.180.11.147:1652/histdb  
    driverClassName: oracle.jdbc.driver.OracleDriver  
    username: dbo  
    password: dbo\_12345

itpt2hist\_histdb

Oracle

[manjula.devadoss@ext.soprasteria.com](mailto:manjula.devadoss@ext.soprasteria.com) / red@Oracle19



Get and upload the ojdbc7.jar file

# BIS GIT Lab

## Folder URL

- Login - http://10.180.11.144/

- Go to -  [<http://10.180.11.144/i-transport-2.0/bis/>](http://10.180.11.144/i-tranport-2.0/BIS/bisapp)

Front End URL

<http://10.180.11.144/i-transport-2.0/bis/bisfrontend>

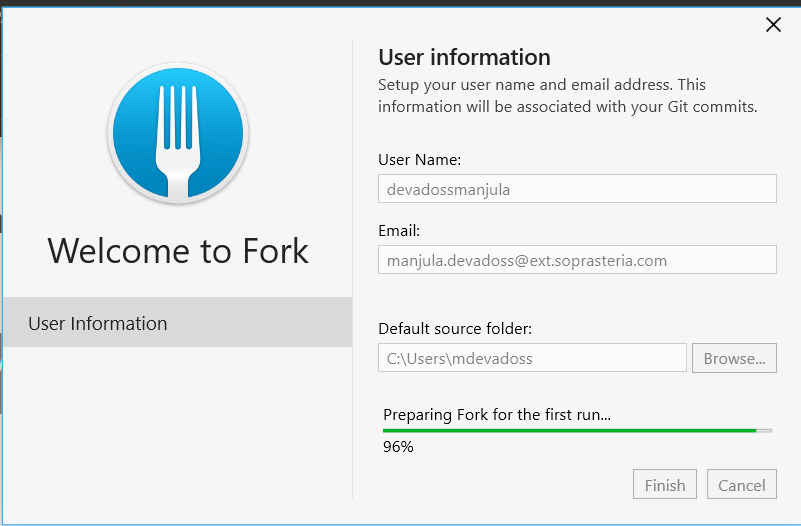
Back End URL

<http://10.180.11.144/i-transport-2.0/bis/bisapp>

## Install Fork and Git bash

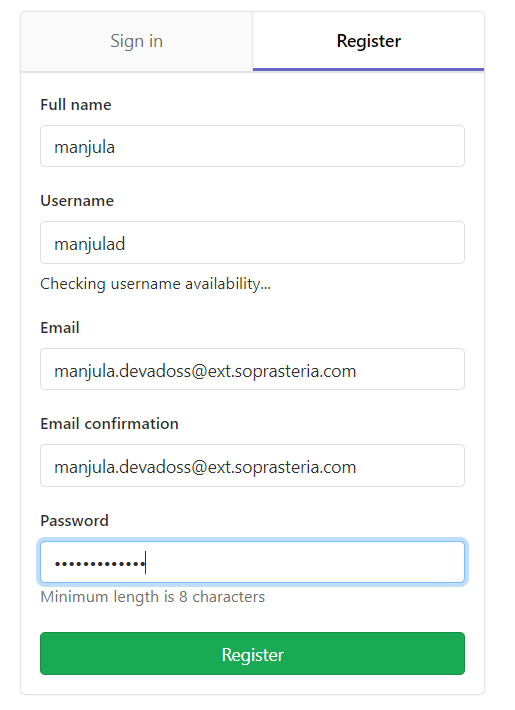
- https://git-scm.com/downloads

- <https://git-fork.com/>



## Gitlab credential

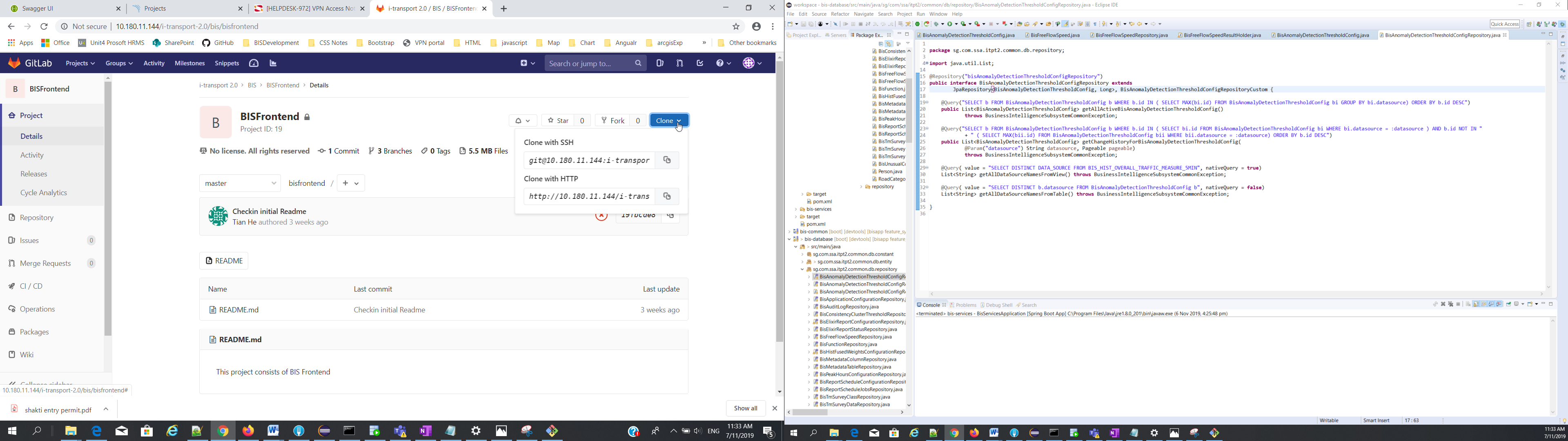
manjulad / sopra@hit1119



## Initial Setup

### Using Git Bash Command window

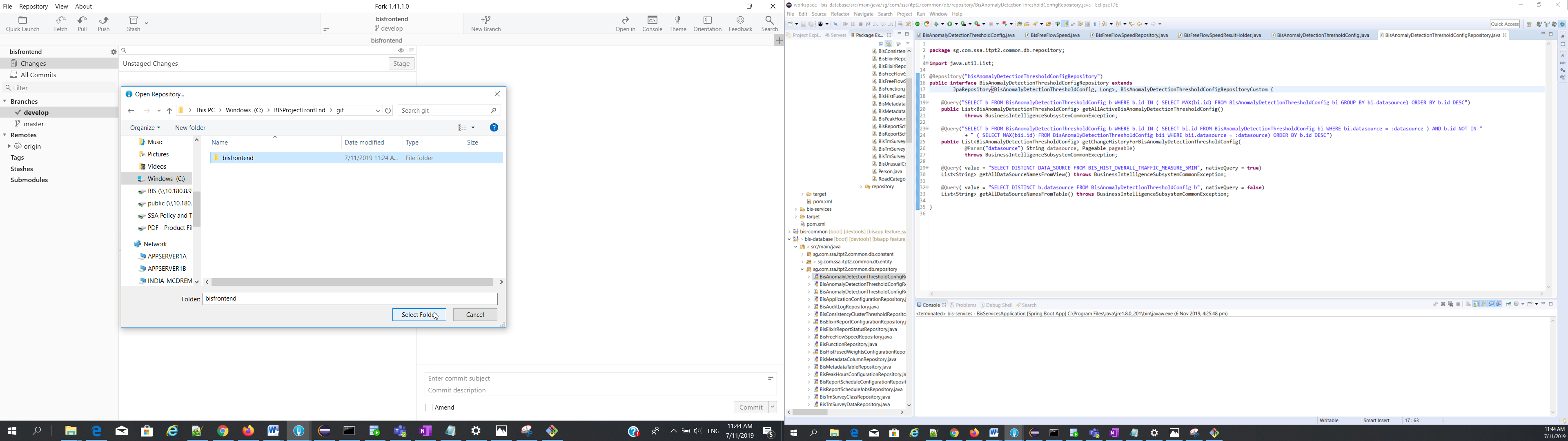
First time Clone : Look the blow picture



* Create “BISProjectFrontEnd” - create “git” – inside “git”
* right click “git bash here”
* Copy url from the gitlab main webpage
* Type command “git clone <http://10.180.11.144/i-transport-2.0/bis/bisfrontend.git>”
* Cd “BISProjectFrontEnd/git/bisfrontend” (local project folder name)
* Git checkout “develop” (remote folder)
* Git pull (Files copy remote to local)

### Using Fork window

* Open Fork App
* File Menu – OpenRepsitory
* Choose your project folder “BisProjectFrontEnd\git\bisfrontend” – select folder



# PostMan

|  |  |
| --- | --- |
| Postman | [manjula.devadoss@ext.soprasteria.com](mailto:manjula.devadoss@ext.soprasteria.com) / ssauser2019/ ac@Post19 |

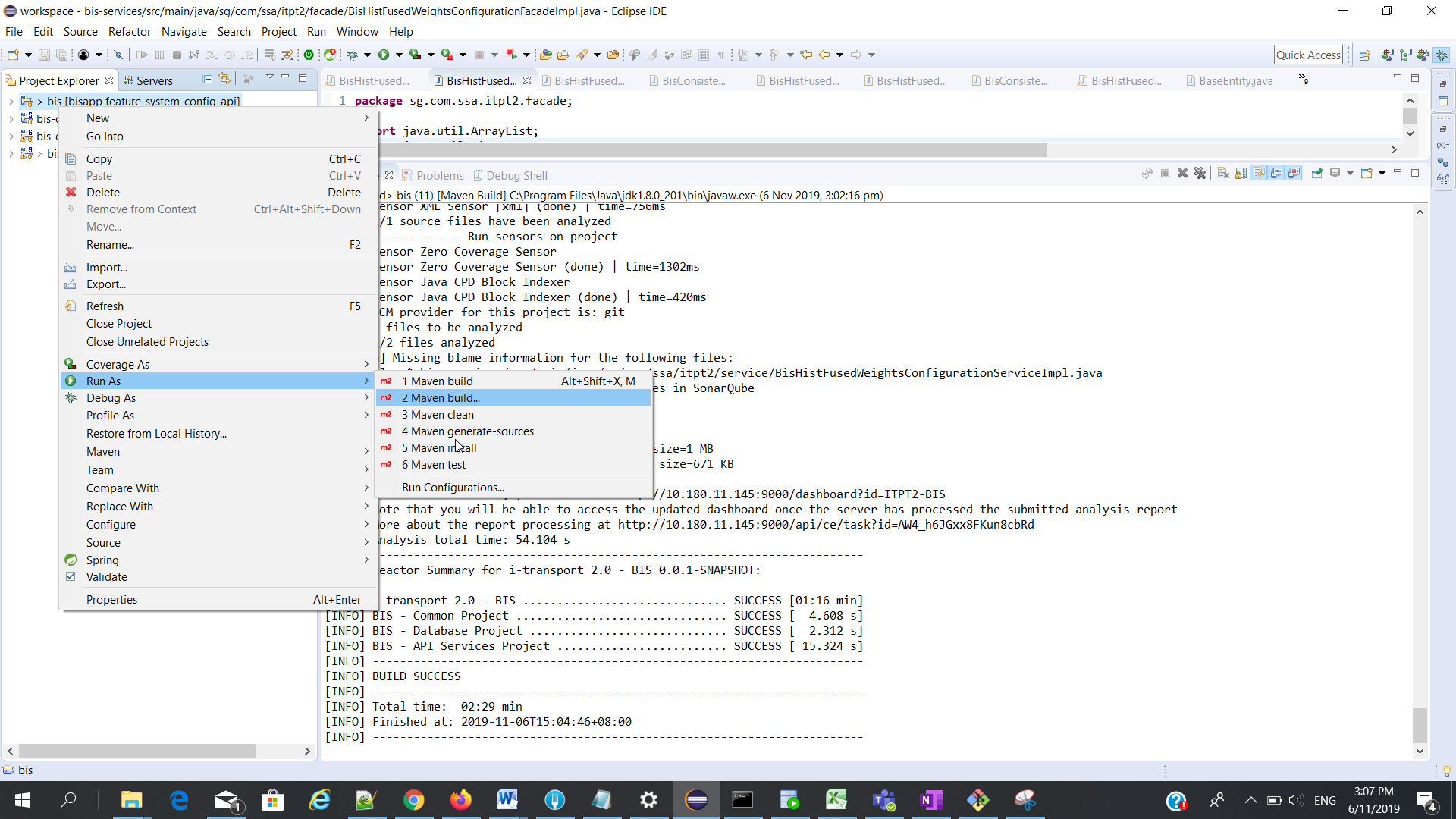
# Sonar

### Setup

From eclipse

Select main project folder – right click “Run As” choose “maven build “(2) option

Type “sonar : sonar”



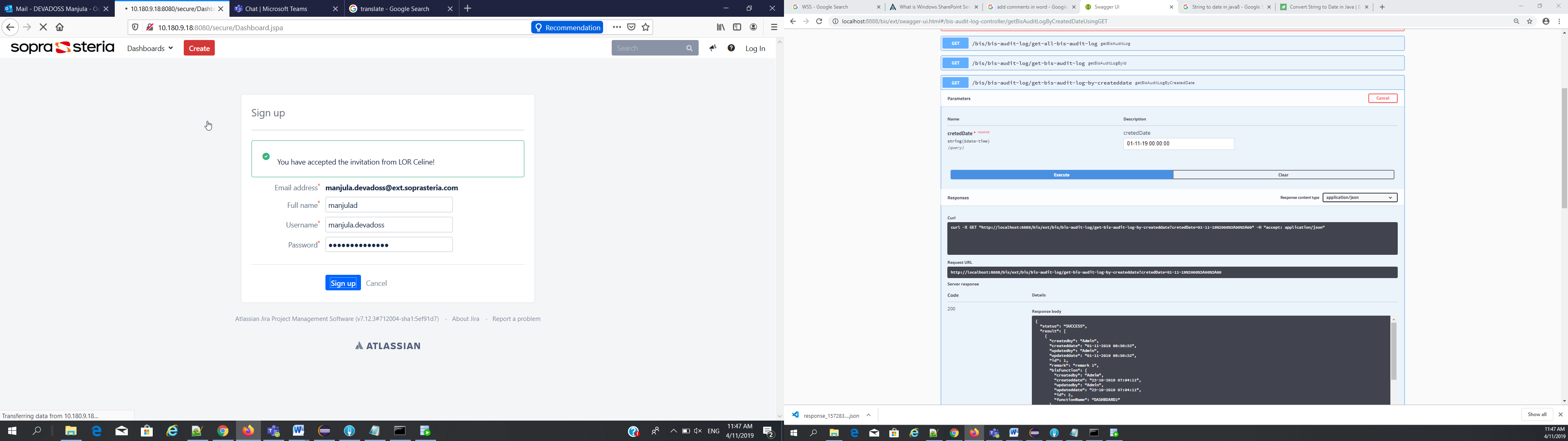
### URL

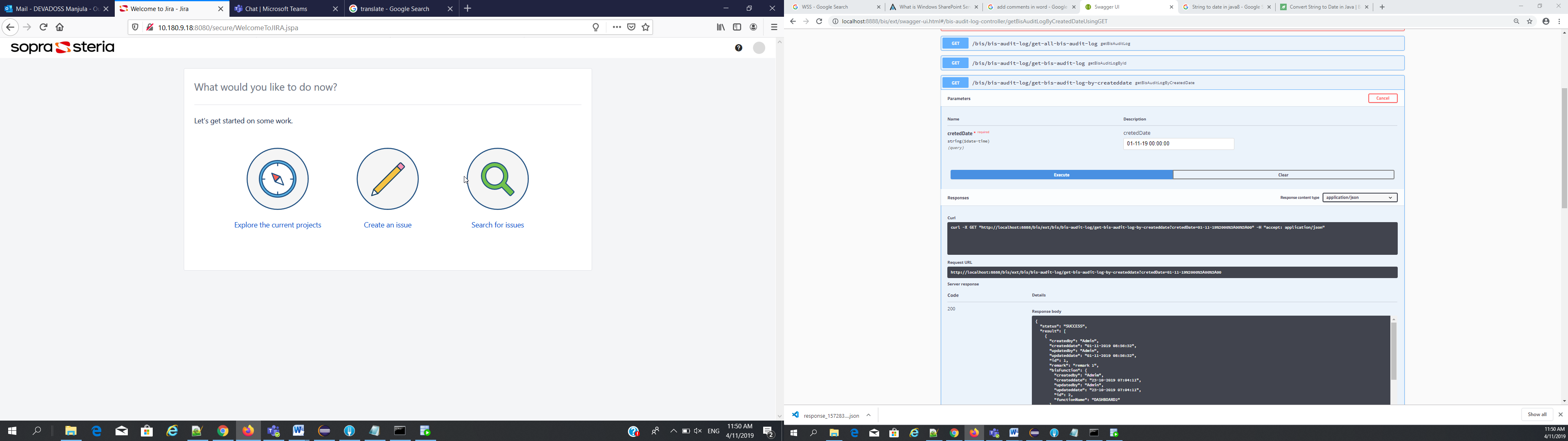
<http://10.180.11.145:9000/projects?sort=-analysis_date>

# JIRA Access

http://10.180.9.18:8080/secure/WelcomeToJIRA.jspa

manjula.devadoss / sopra@jira1119





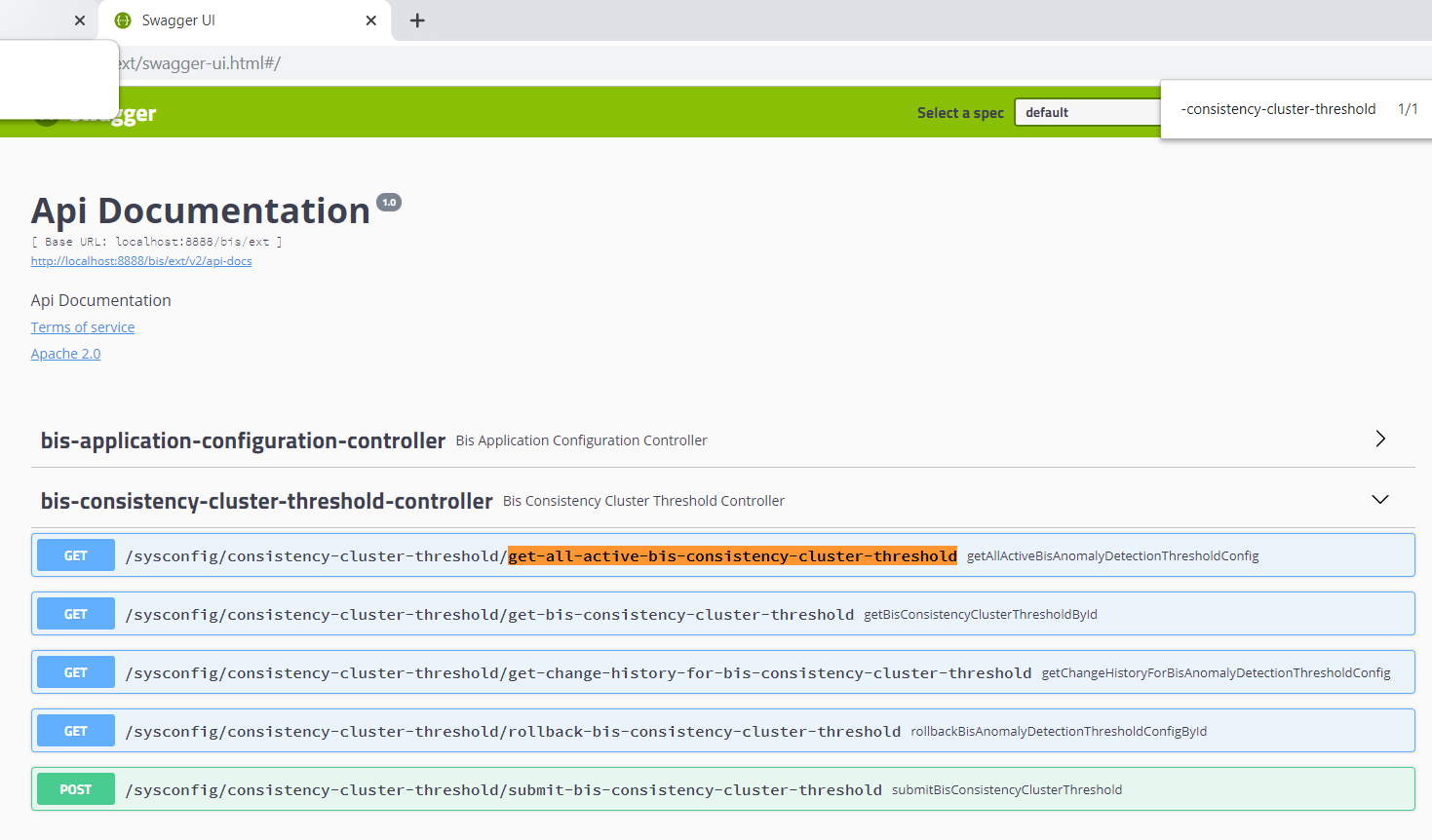
# Swagger URL

To Run RestAPI

<http://localhost:8888/bis/ext/swagger-ui.html>

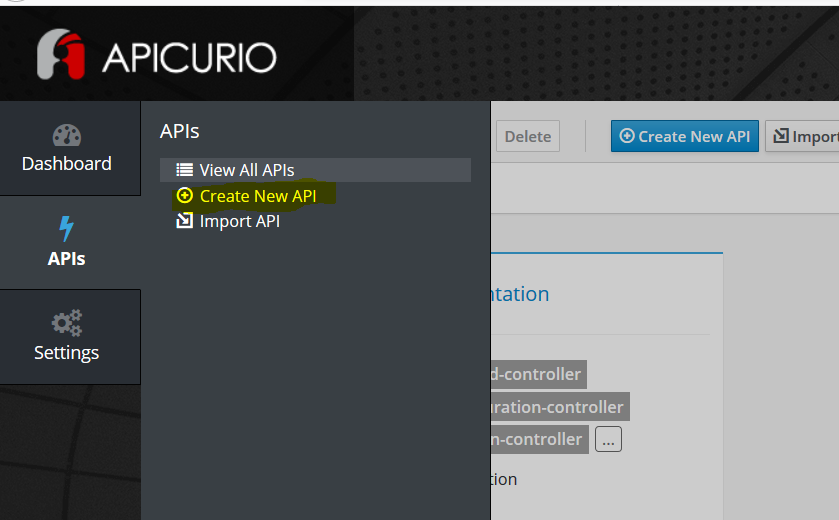
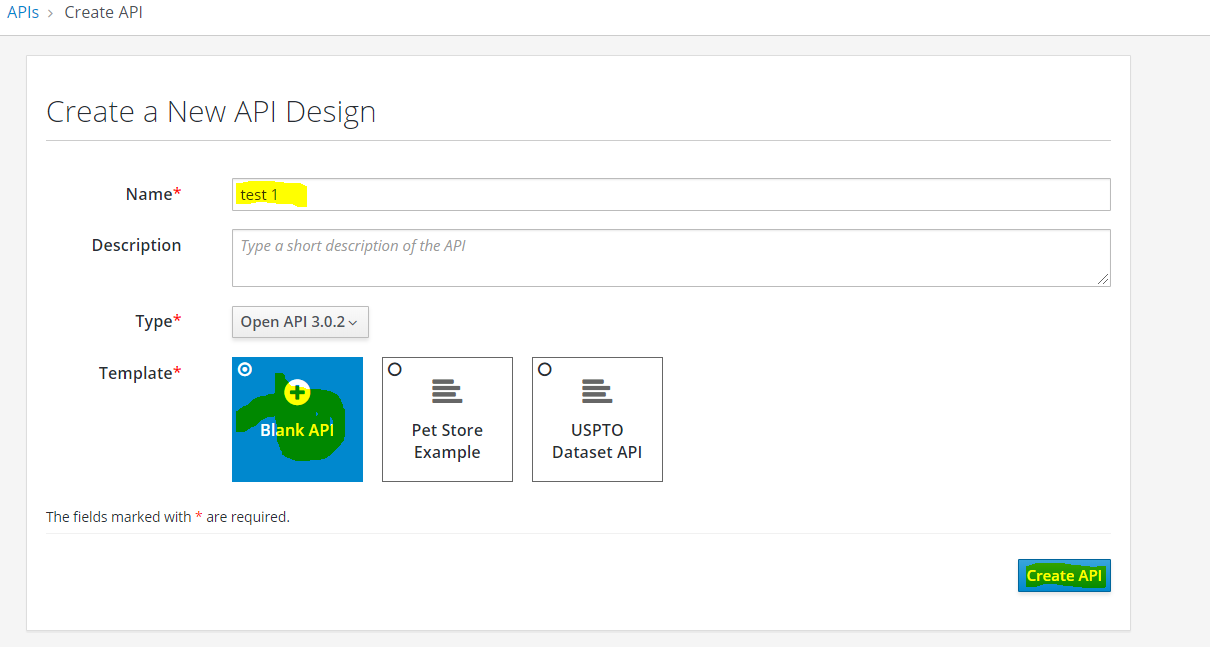
# Run the application - URL

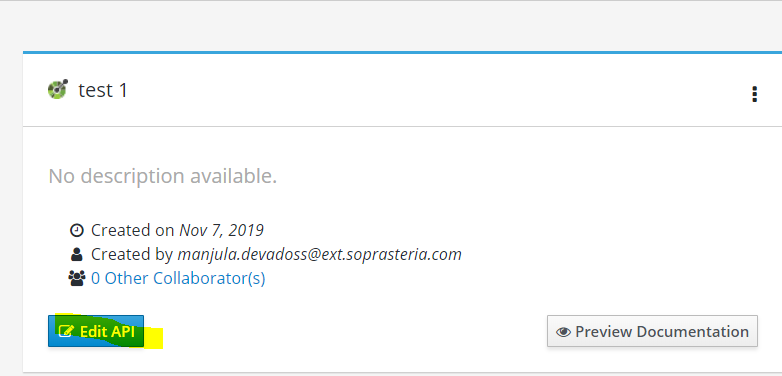
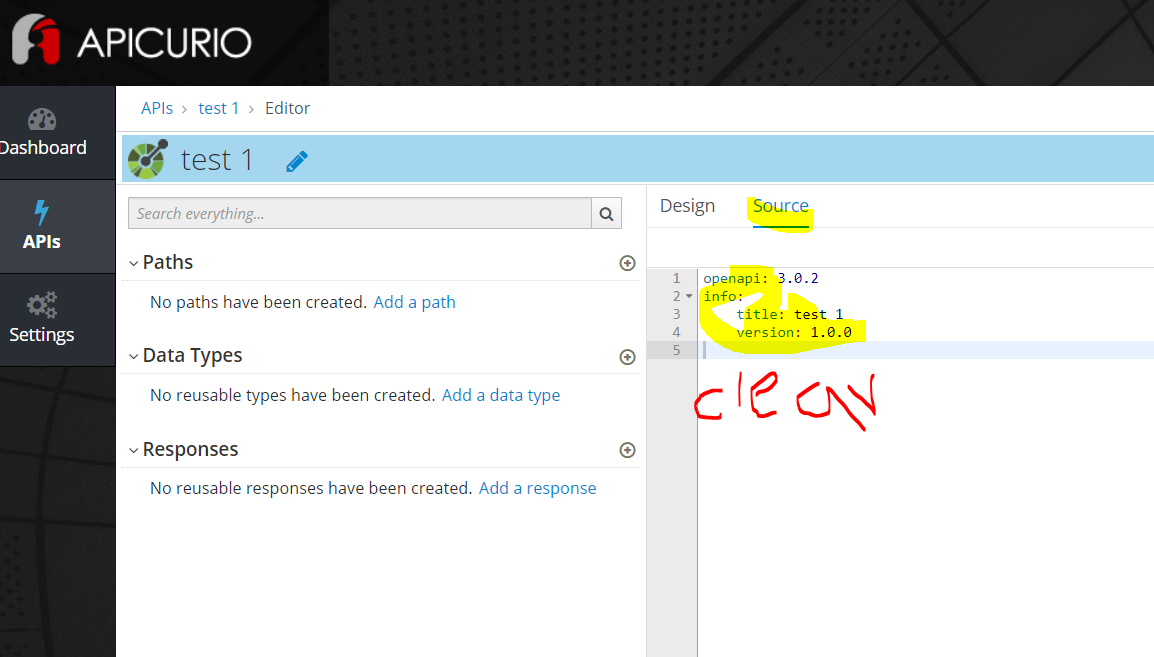
<http://localhost:8888/bis/ext/swagger-ui.html#/>



# Apicurio

* Goto : <https://www.apicur.io/>
* Choose “Try Live”
* ---- Choose “HitHub”
* Give github uname/password
* Sopraid/ac@Sopragit19

Clear the source and Copy from Swagger “<http://localhost:8888/bis/ext/v2/api-docs>”

Click the above link and copy the code



## WinSCP

Remote Folder Access

host name: 10.180.11.133

username : elixir

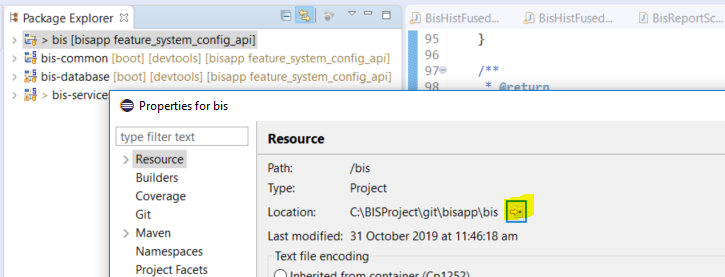
password: P@55w0rd12345

Remote Folder location /home/elixir/LargeReport

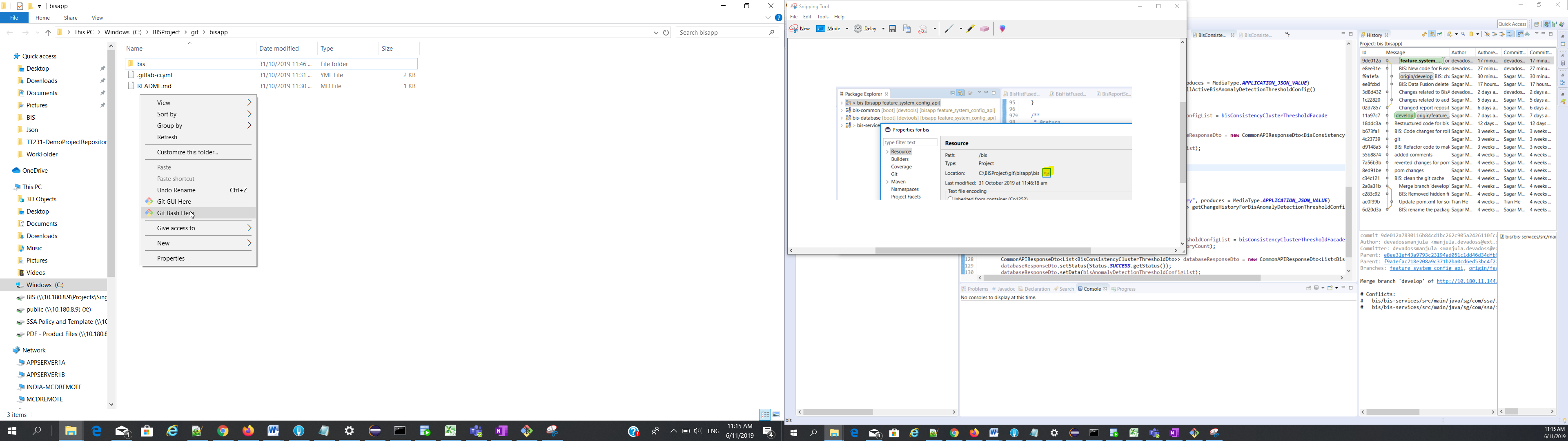
(Large report file stored here)

# Gitlab Pull and Push in After change

Using Eclipse goto the project work space



From the project – right click and choose “Git Bash here”



Command: git Pull



Command : git Pull origin develop



Merge to Original Development from Local Development.

goto develop folder

- git checkout develop



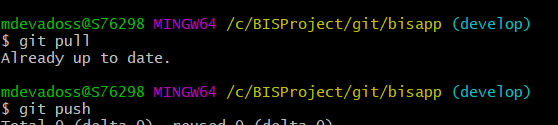
merge local(Feature) to develop folder

- git merge feature\_system\_config\_api

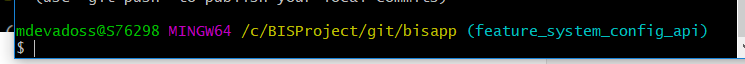


- git pull

- git push



goto local folder

* git checkout feature\_system\_config\_api
* 
* 

# GITLAB Comments Steps

## 1) Git Clone

- create folder "demo"

- open git bash inside domo folder

- git clone http://10.180.8.23/TT231\_ITPT2/BIS/bisapp.git

- cd bisapp/

Frontend clonning

<http://10.180.11.144/i-transport-2.0/bis/bisfrontend.git>

## 2) Git Create and Checkout Branch

- create branch ( feature\_test ) using gitlab

- git checkout feature\_test

## 3) Git Status

- git status

## 4) Git Add

- git add .

## 5) Git Commit

- git commit -m "Message Text"

## 6) Git Pull

- git pull

- git pull origin development

## 7) Git Push

- git push

* Create a .gitignore file with contents as \* or file name.

## 8) Git Merge - Branching (TO do with care)

- git checkout develop

- git merge "feature\_branch\_name"

- git pull

- git push

- git checkout "feature\_branch\_name"

- Tableaue and Elixir code will be override.

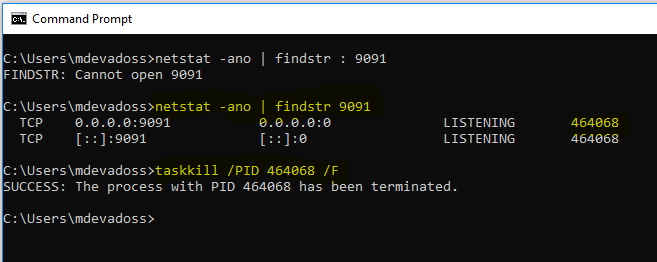
# Tomcat Error – 404

org.apache.catalina.LifecycleException: Protocol handler start failed

**Tomcat is not stopped properly, so kill the existing running port. 9091**

**Check the port : netstat –ano | findstr 9091**

**Kill Port : taskkill /PID 464068 /F**



# Augular 6 Setup

<https://www.djamware.com/post/5b87894280aca74669894414/angular-6-httpclient-consume-restful-api-example> -Need to see Modify `@NgModule` imports to be like this.

1. Create new folder “ng new angular6-restapi --style=scss --routing --skip-install”



1. Goto the directory “cd angular6-restapi ”



1. Install npm “npm install ”



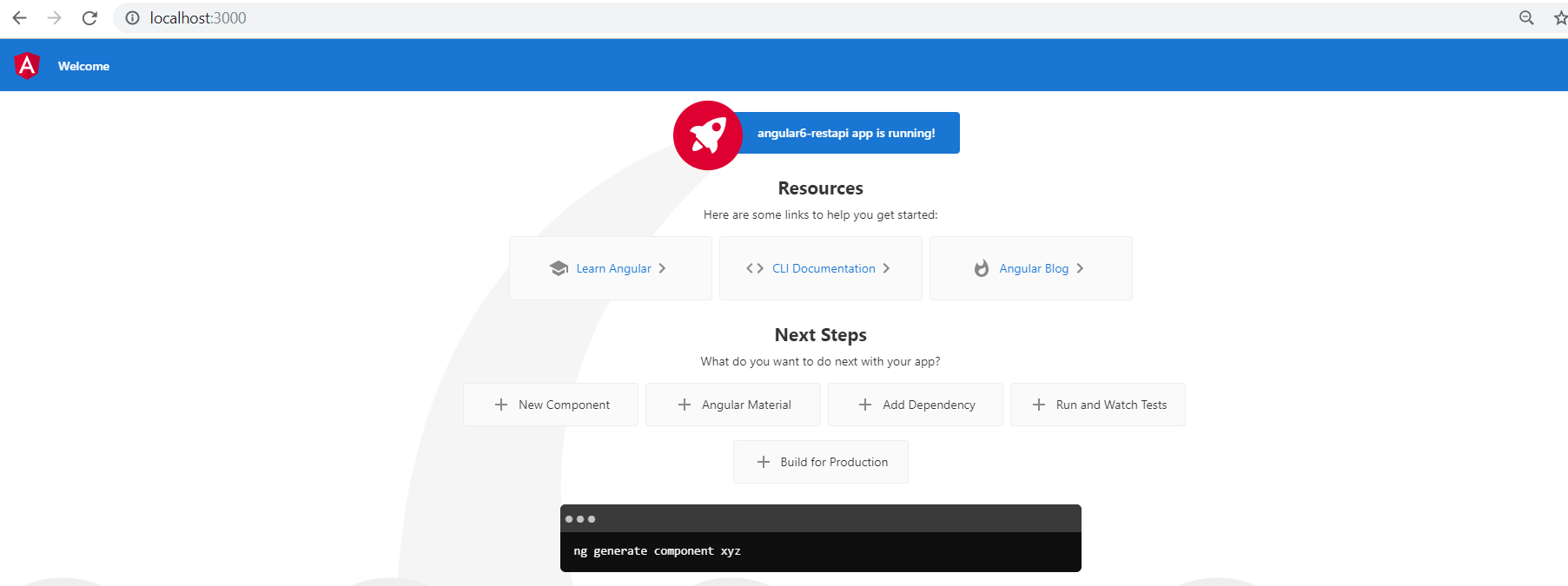
1. Run the new Angular 6 web application using your own host and port.

“ng serve --host 0.0.0.0 --port 3000”



1. Open browser

<http://localhost:3000/>



1. Open the File app.module.ts

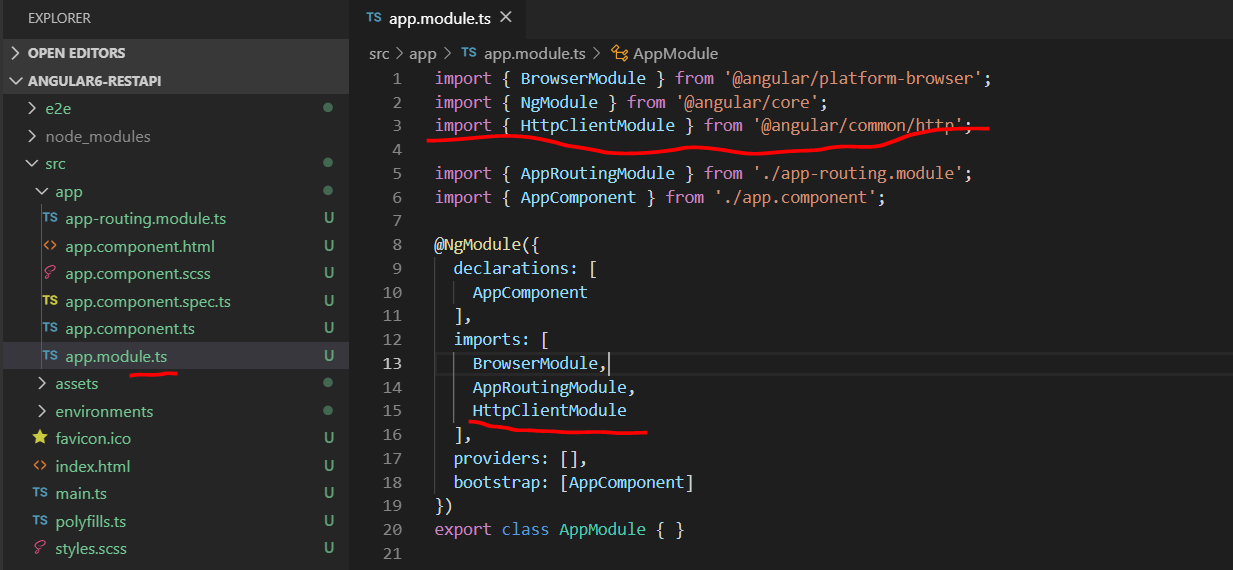
**import { HttpClientModule } from '@angular/common/http';**

**imports: [**

**BrowserModule,**

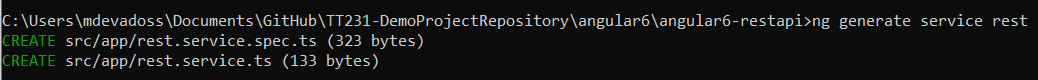
**HttpClientModule**

**],**



1. Create Angular 6 service for consuming REST Service

“ng generate service rest”



1. Open rest.service.ts add following lines

import { Injectable } from '@angular/core';

import { HttpClient, HttpHeaders, HttpErrorResponse } from '@angular/common/http';

import { Observable, of } from 'rxjs';

import { map, catchError, tap } from 'rxjs/operators';

const endpoint = '<http://localhost:3000/api/v1/>';

const httpOptions = {

  headers: new HttpHeaders({

    'Content-Type':  'application/json'

  })

};



1. asdfadsfadsfasf

# Class Diagram

<Class>

BisConsistencyClusterThresholdController

getBisConsistencyClusterThresholdById()

<Façade Interface>

BisConsistencyClusterThresholdFacade

getBisConsistencyClusterThresholdById(Long id)

< Façade Implementation Class>

BisConsistencyClusterThresholdFacadeImpl

getBisConsistencyClusterThresholdById(Long id)

<Service Interface>

BisConsistencyClusterThresholdService getBisConsistencyClusterThresholdById(Long id)

< Service Implementation Class>

BisConsistencyClusterThresholdServiceImpl

getBisConsistencyClusterThresholdById(Long id)

< Repository Interface>

BisConsistencyClusterThresholdRepository.java

DAO to DTO

# Logs Location

C:\SoprSteriaWork\mavenMultiModule\logs

# Coding procedure

1. create entity related to database table [refer bis free flow speed entity]

2. create repository

3. create service  [audit log service]

               interface

               impl

4.            create facade layer [dto]

               interface

               impl

5. create controller

               get all records

               get single record by id

               insert [dto should not contain id]

               update for weights [dto should have id]

               delete [by id]

               update only status by id

6. review your code

7 will check for sonar

# Scheduled Task in SpringBoot

<https://dzone.com/articles/running-on-time-with-springs-scheduled-tasks>

/\* @Scheduled(cron = "[Seconds] [Minutes] [Hours] [Day of month] [Month] [Day of week] [Year]") \*/

Fires at 12 PM every day:

@Scheduled(cron = "0 0 12 \* \* ?")

Fires at 10:15 AM every day in the year 2005:

@Scheduled(cron = "0 15 10 \* \* ? 2005")

Fires every 20 seconds:

@Scheduled(cron = "0/20 \* \* \* \* ?")

Below is a breakdown of the components that build a cron expression.

* Seconds can have values 0-59 or the special characters , - \* / .
* Minutes can have values 0-59 or the special characters , - \* / .
* Hours can have values 0-59 or the special characters , - \* / .
* Day of month can have values 1-31 or the special characters , - \* ? / L W C .
* Month can have values 1-12, JAN-DEC or the special characters , - \* / .
* Day of week can have values 1-7, SUN-SAT or the special characters , - \* ? / L C # .
* Year can be empty, have values 1970-2099 or the special characters , - \* / .

# Use Case diagram

## Anomaly detection and removal

Table Name:

BIS\_ANOMALY\_DETECTION\_THRESHOLD\_CONFIGURATION

Query :

Edit Single

Rollback

Save Single Record

Save

Save Many Record

Edit Many

## Edit Fusion Weightage

Requester View

**Tables :**

**BIS\_HIST\_FUSED\_WEIGHTS\_CONFIGURATION;**

**BIS\_FUSION\_MODEL**

**Query 1: Select all values list by model\_number and show the highest id.**

**Query 2: Inset record in** HIST\_FUSED\_Weightage\_configuration **Table with Drafted Status.**

**Query 3: Inset record in** HIST\_FUSED\_Weightage\_configuration **Table with Pending Status.**

**Query 3: History - Search Last 3 highest id** HIST\_FUSED\_Weightage\_configuration **Table with Completed Status.**

**Notes : in table Model\_Description - is lable name of Datasource**

Requester

Proposed Weight

End WorkFlow

Edit

Apply

Proposed Weight

OK

Proposed Weight

Edit Fusion Weightage Approver View

Approver

Edit

Apply

OK

End WorkFlow

Implemented Weight

Implemented Weight

Implemented Weight

Approver is also able to implement weights by passing the approval workflow in OWM as shown in Figure 5‑70 Control panel for approver and Figure 5‑71 Propose Weights for Live Fusion (approver). If the status of fusion approval workflow is shown as “pending”, approver will not be able to edit or implement weights until the workflow is completed (see Figure 5‑72).

Edit Fusion Weightage Approver View HISTORY

History

Apply

OK

Implemented Weight

Implemented Weight

End WorkFlow

Edit Fusion Weightage Email Recipients GUI / Workflow

View

Save

Approver Details

Approved Recipients Detail

Search

Add

## Pending task in data Fusion Weightage

Fusion weights implementation Web services (TIQ)

Ask Sofiya

The Traffic Information Quality (TIQ) submodule in Cohesive Operations Subsystem (COS) will consume the following Web services for the live implementation of the new fusion weights after the implement action is executed in 5.8.3.1.2

**Anomaly Detection and Removal web services**

BIS will consume the following webservice from Traffic Information Quality (TIQ) sub module in Cohesive Operations Subsystem (COS) to receive the threshold.

Below is the WEB SERVICE that BIS will use to transfer the threshold to TIQ.

speed factor - not needed

For the Anomaly Detection and Removal web services

I have following data in Json format

[

{

"createdby": "Admin",

"createddate": "14-11-2019 11:22:05",

"updatedby": "Admin",

"updateddate": "19-11-2019 12:58:00",

"id": 103,

"datasource": "EMAS",

"minSpeed": 0,

"maxSpeed": 0,

"minOcc": 0,

"maxOcc": 0,

"minVolume": 0,

"maxVolume": 22,

"remark": "ghg"

}

]

Referring Table : BIS\_ANOMALY\_DETECTION\_THRESHOLD\_CONFIGURATION

What Data source do you need and the group of?? What additional tag ??

Where do I see following info?

"maxSpeed",

"roadCapacity",

“createDate",

"roadType",

"speedFactor",

"volumeFactor",

"userId"

## Report Repository

Table Name : BIS\_REPORT\_SCHEDULE\_CONFIGURATION" –

* Param\_Start time, Param\_end time? Where is the values

Table Name: BIS\_REPORT\_SCHEDULE\_JOB

### Table Name : BIS\_REPORT\_STATUS - When will update the status.?

Elixier Report Tables

BIS\_REPORT\_STATUS

BIS\_REPORT\_CONFIGURATION

Input

Save

Report

Schedule

Search

Add

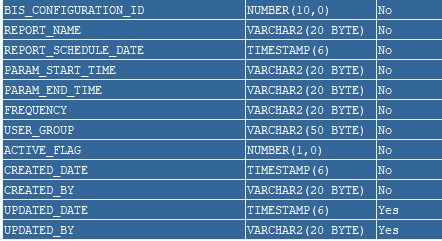
Output

**When you schedule the report – Record added in the following two tables.**

BIS\_REPORT\_SCHEDULE\_CONFIGURATION

BIS\_REPORT\_SCHEDULE\_JOB\_LOG





1. when this table get records

2. is there any link between -

bis\_report\_schedule\_job\_log table &

bis\_report\_status

3.what are the status data - in two tables

4.bis\_report\_schedule\_configuration - table

frequency?

user\_group?

active flag?

where do i get reportname?

Records inserted into two tables

BIS\_REPORT\_SCHEDULE\_CONFIGURATION

BIS\_REPORT\_SCHEDULE\_JOB\_LOG

downloadfiles

BIS\_REPORT\_STATUS - .xl file name but no location

BIS\_REPORT\_CONFIGURATION - .rml file server location

BIS\_APPLICATION\_CONFIGURATION - path,ul,folder location

**One to many and Many to one**

Table : BIS\_REPORT\_SCHEDULE\_CONFIGURATION

id : BIS\_CONFIGURATION\_ID (present one time)

public class BisReportScheduleConfiguration {

@OneToMany(mappedBy = "bisReportScheduleConfiguration")

private Set<**BisReportScheduleJob**> bisReportScheduleJobSet;

}

Table : BIS\_REPORT\_SCHEDULE\_JOB

id : BIS\_CONFIGURATION\_ID (present many times)

@ManyToOne

@JoinColumn(name = "BIS\_CONFIGURATION\_ID", nullable = false)

private BisReportScheduleConfiguration bisReportScheduleConfiguration;

Check

ExecutorService

Future

# Scheduler in Spring boot

@Scheduled(cron = "0 1 1 \* \* ?")

Below you can find the example patterns from the spring forum:

\* "0 0 \* \* \* \*" = the top of every hour of every day.

\* "\*/10 \* \* \* \* \*" = every ten seconds.

\* "0 0 8-10 \* \* \*" = 8, 9 and 10 o'clock of every day.

\* "0 0 8,10 \* \* \*" = 8 and 10 o'clock of every day.

\* "0 0/30 8-10 \* \* \*" = 8:00, 8:30, 9:00, 9:30 and 10 o'clock every day.

\* "0 0 9-17 \* \* MON-FRI" = on the hour nine-to-five weekdays

\* "0 0 0 25 12 ?" = every Christmas Day at midnight

Cron expression is represented by six fields:

second, minute, hour, day of month, month, day(s) of week