Installation and Configuration for a new Tripwire Virtual Machine

[Access the VDI](#_e18x4ge0ywm3)

[Connect using Citrix](#_1fob9te)

[VDI’s Information](#_4ty7qai1apgz)

[Citrix VM](#_3dy6vkm)

[Map Fileshare](#_1g5jim5xsn15)

[cx\_Oracle](#_1t3h5sf)

[MySQL Database](#_4kanmcfz8pc1)

[Install Oracle Database](#_2s8eyo1)

[Install SQL Server Database](#_17dp8vu)

[Uninstall TE](#_rt1ad59z4z3u)

[Install TE](#_3rdcrjn)

[1) Execute installer](#_26in1rg)

[2) Configuration steps](#_lnxbz9)

[3) TE License Configuration](#_35nkun2)

[Certificate configuration](#_1ksv4uv)

[Configure TE properties](#_44sinio)

[Configuration of remote debugger](#_wei22q5aqws2)

[Add grant permissions into JAAS Policy](#_stdkp7301mb7)

[Cloning the project repositories from GitHub](#_2xcytpi)

[Artifacts](#_1ci93xb)

[Build Projects](#_3whwml4)

[Intellij installation](#_qsh70q)

[Import projects to IntelliJ](#_3as4poj)

[Data population - Setup Automation Data](#_1pxezwc)

[Gradle tasks](#_49x2ik5)

[Project Structure](#_2p2csry)

[Files of the project](#_147n2zr)

[Features (features/rest-api and features/client)](#_3o7alnk)

[Steps (features/steps)](#_23ckvvd)

[Environment (features/environment.py)](#_ihv636)

[Mapping (features/mapping.py)](#_32hioqz)

[Utils (utils)](#_1hmsyys)

[Behave (behave.ini)](#_41mghml)

[Configuration file (te-rest-automation.cfg)](#_2grqrue)

[Running the projects](#_nmf14n)

[Automation project:](#_37m2jsg)

[Databases - Oracle SQL Developer](#_1mrcu09)

[Te-Rest-Api](#_46r0co2)

[Set Up the Debugger for Te-Rest-Api](#_2lwamvv)

[Update TE from unsupported versions](#_111kx3o)

[Fixes for detected errors and tips](#_3l18frh)

[Links](#_206ipza)

[Globant](#_4k668n3)

[Tripwire public](#_2zbgiuw)

[Tripwire private (only accessed from Citrix or VDI)](#_1egqt2p)

[Technologies](#_cz18h3h4pgj5)

# Access the VDI

*VDI: Virtual desktop infrastructure where our Development environments are.*

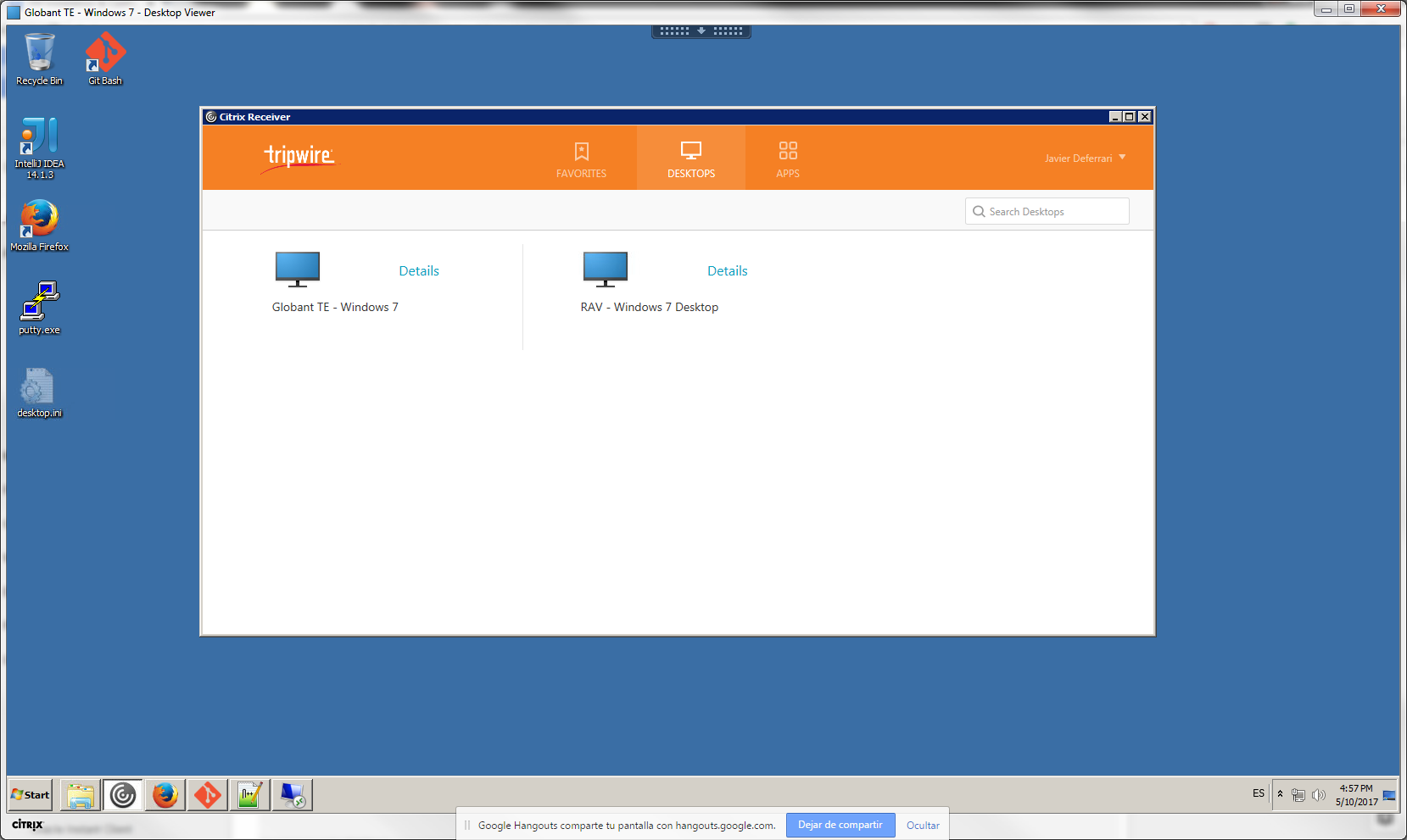
To access to any Tripwire site, the following mobile application is needed to authenticate the user account:

**DUO:** <https://play.google.com/store/apps/details?id=com.duosecurity.duomobile&hl=en>  
(\**Advise: Add your desktop/soft phone in case you lose your cellphone*)

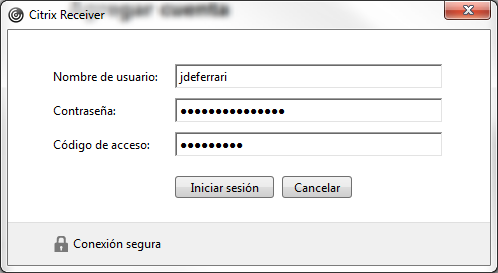
## Connect using Citrix

In order to connect to the VDI we need to use citrix.

1. Go to <https://citrix.tripwire.com> and login using the tripwire account and duo authentication
2. Go to applications and on the user’s menu click ‘activate’. This will detect the installed citrix and activate it.
   1. The first time it will download a .cr file. Open it using citrix.



1. Log into citrix with the user name, pwd and access code. Citrix will ask for this after adding the account or trying to access an application
   1. To get the access code ask the DUO Mobile app for a code (hit the key icon to get the numeric code)



1. Close the browser page (won’t need the browser anymore)
2. Open citrix
3. Connect to the personal VDI from the desktop tab.

### VDI’s Information

The actual environment settings can be found at: <https://docs.google.com/spreadsheets/d/1zv55BAzeOirxQ5njV9N6X7b1GigBQzcd8LUfsoYvTNs/edit#gid=1406666474>

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **VM** | **IP** | **OS** | **CPUs** | **RAM** | **Disk** | **Owner** | **Comments** |
| [glob-terav01.dev.lab.tripwire.com](http://glob-terav01.dev.lab.tripwire.com) | 10.248.169.31 | Windows |  |  |  |  |  |
| [globw2k8r2qa-07.dev.lab.tripwire.com](http://globw2k8r2qa-07.dev.lab.tripwire.com) | 10.248.169.16 | Windows |  |  | 200/200 GB |  | SQL Server, TE 8.5.3 |
| [globw2k8r2qa-08.dev.lab.tripwire.com](http://globw2k8r2qa-08.dev.lab.tripwire.com) | 10.248.169.17 | Windows |  |  | 50/100 GB |  | SQL Server, TE 8.5.6 |
| [globw2k8r2qa-09.dev.lab.tripwire.com](http://globw2k8r2qa-09.dev.lab.tripwire.com) | 10.248.169.18 | Windows |  |  | 131/200 GB |  | Oracle, TE 8.6.1 |
| [globw2k8r2qa-10.dev.lab.tripwire.com](http://globw2k8r2qa-10.dev.lab.tripwire.com) | 10.248.169.19 | Windows |  |  | 33/100 GB |  | Axon Agent, TE 8.5.4b |
| [globw2k8r2qa-13-rename.dev.lab.tripwire.com](http://globw2k8r2qa-13-rename.dev.lab.tripwire.com) | unknown | Windows |  |  |  |  |  |
| [globw7temp01.dev.lab.tirpwire.com](http://globw7temp01.dev.lab.tirpwire.com) | unknown | Windows |  |  |  |  |  |
| [globw7x64-01.dev.lab.tripwire.com](http://globw7x64-01.dev.lab.tripwire.com) | 10.248.169.10 | Windows |  |  | 68/80 GB | ? |  |
| [globw7x64-02.dev.lab.tripwire.com](http://globw7x64-02.dev.lab.tripwire.com) | 10.248.169.11 | Windows |  |  | 75/80 GB | Fer | Oracle, SQL Server, MySQL |
| [globw7x64-03.dev.lab.tripwire.com](http://globw7x64-03.dev.lab.tripwire.com) | 10.248.169.12 | Windows |  |  | 80 GB | Ire | MySQL |
| [globw7x64-04.dev.lab.tripwire.com](http://globw7x64-04.dev.lab.tripwire.com) | 10.248.169.13 | Windows |  |  | 73/80 GB | Flor | MySQL |
| [globw7x64-05.dev.lab.tripwire.com](http://globw7x64-05.dev.lab.tripwire.com) | 10.248.169.14 | Windows |  |  | 80 GB |  | SQL Server |
| [globw7x64-06.dev.lab.tripwire.com](http://globw7x64-06.dev.lab.tripwire.com) | 10.248.169.15 | Windows |  |  | 80 GB | Meri | MySQL |
|  | 10.248.169.20 | CentOS |  |  | 25 GB |  | TE 8.6. SSH login: mvillar:NnypeAmectrall#, use sudo -i |
|  | 10.248.169.21 | CentOS |  |  |  |  |  |
| [glob-terhel01.dev.lab.tripwire.com](http://glob-terhel01.dev.lab.tripwire.com) | 10.248.169.50 | CentOS |  |  | 800 GB (180 GB partitions) |  |  |
| GlobWin7TE01 | 10.2.2.52 | Windows VDI |  |  |  | Fer |  |
| GlobWin7TE02 | 10.2.2.50 | Windows VDI |  |  |  | Flor |  |
| GlobWin7TE05 | 10.2.2.53 | Windows VDI |  |  |  |  |  |
| GlobWin7TE08 | 10.2.2.54 | Windows VDI |  |  |  |  |  |

# Citrix VM

Virtual machines provided by tripwire to Globant TE project will have the following programs already installed:

* Git & Git Bash
* Firefox
* IntelliJ IDE 14.1.3
* Visual C++ 2013 Runtime
* Microsoft .Net Framework 4.5.1
* Python 2.7 & python added to PATH
  + Installed Python libraries can be found at C:\Python27\Lib\site-packages
* 7-zip
* Notepad++
* Java SE
* MySQL - database: **te**; root password: ***Passw0rd***
* Installed deprecated version of TE - passphrase: ***TEServicesPassphrase***

Steps to Configure VDI

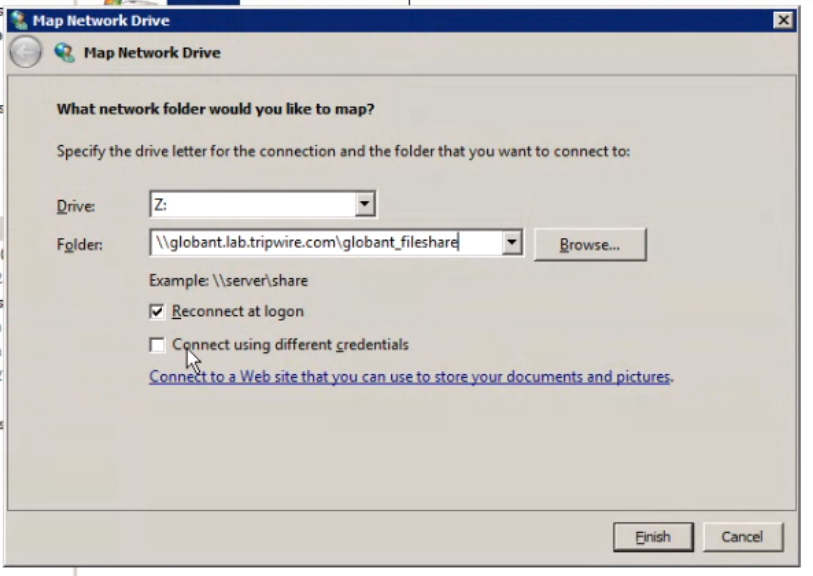
1. Map globant\_fileshare for easy access ([here](#_1g5jim5xsn15))
2. Add C:\Program Files\MySQL\MySQL Server 5.6\bin to PATH
3. Recreate MySQL Database ([here](#_4kanmcfz8pc1))
4. Unistall **TE**, clean Tripwire folder & Install latest version ([here](#_3rdcrjn))
5. Set ***Password1@*** as administrator password
6. Install CA Root certificates ([here](#_1ksv4uv))
7. Configure TE Server properties ([here](#_44sinio))
8. Without uninstalling python, run the installer and select option “Install for all”
9. Run MySQL installer and install MySQL Python Connector
10. Copy testdata folders from shared/TE-REST-API/automation to your C:/ drive
11. Install Oracle connector ([here](#_1t3h5sf))
12. Clone projects ([here](#_2xcytpi))
13. Configure connection to Artifactory ([here](#_1ci93xb))
14. Build projects ([here](#_3whwml4))
15. Import projects to IntelliJ ([here](#_3as4poj))
16. Populate Database ([here](#_1pxezwc))

### Map Fileshare

**\\globant.lab.tripwire.com\globant\_fileshare**

In this location is where all the data relevant to Tripwire projects is shared between the client and Globant.

**Installers** and necessary packages to configure the VM can be found here.



\*Tip: Run all the installers with As Administrator, as not to be bother many times for installation being asked for permissions.

## cx\_Oracle

1. Make sure that python was added to the registry (regedit.exe) in HK\_LOCAL\_MACHINE/Software/Python
   1. Otherwise run **\\globant.lab.tripwire.com\globant\_fileshare\Python\python27.reg** to add it
2. Install the Oracle Instant Client ODBC libraries (shouldn’t be necessary if the VM already has a full Oracle installation)

Install the needed version of the instant client. If Python is 32 bits, then install the 32 bits instant client.

The installer is on **\\globant.lab.tripwire.com\globant\_fileshare\Oracle11g**

1. Unzip the basic and odbc zip files on c:\Oracle (should create a **c:\Oracle\instantclient\_11\_2** folder)
2. Open a cmd as administrator on the instantclient\_11\_2 folder
3. Run odbc\_install.exe
4. Add **;c:\Oracle\instantclient\_11\_2** to the end of the Path environment variable
5. Install cx\_Oracle: **\\globant.lab.tripwire.com\globant\_fileshare\Oracle11g\*version\*cx\_Oracle-5.2.1-11g.win32-py2.7**
   1. Run the exe, it should detect the python install from the registry.

# MySQL Database

*Database:* ***te***

*User:* ***root***

*Password:* ***Passw0rd***

Run the following query to regenerate TE database:

**SET GLOBAL max\_allowed\_packet=1073741824;**

DROP DATABASE te;

CREATE DATABASE te CHARACTER SET utf8 COLLATE utf8\_bin;

-- Grant privileges to root user

GRANT ALL PRIVILEGES ON te.\* TO 'root'@'localhost';

Known issues:

After restarting a VDI, an maxAllowedPackages error could be thrown by TE server.

There was an error connecting to the Tripwire backend database: Invalid max\_allowed\_packet value detected. Expected: '1073741824', found: '4194304'.

Then, Rerun the following query in Mysql

**SET GLOBAL max\_allowed\_packet=1073741824;**

# Install Oracle Database

/globant.lab.tripwire.com/globant\_fileshare/TE\_installer\_guides/te\_install\_and\_maint\_guide.pdf (page 28)

Open SQL Plus

Connect system/tripwire

Create a tablespace:

CREATE TABLESPACE <tablespace name>

DATAFILE 'C:\app\Administrator\oradata\orcl\<tablespace name>.DBF'

SIZE 2000M

AUTOEXTEND ON

EXTENT MANAGEMENT LOCAL;

Create a user:

CREATE USER <user name>

IDENTIFIED BY <user password>

DEFAULT TABLESPACE <tablespace name>

QUOTA UNLIMITED ON <tablespace name>;

Grant user privileges:

GRANT CREATE VIEW, CREATE TABLE, ALTER SESSION, CREATE SESSION, CREATE SYNONYM,CREATE SEQUENCE, CREATE TRIGGER, CREATE PROCEDURE TO <user name>;

Tips for Oracle databases:

*Connect to a tablespace:*

connect <user name>;

password: <user password>

*Display all tables:*

select tablespace\_name, table\_name from user\_tables;

*Delete tablespace:*

DROP TABLESPACE <tablespace name> INCLUDING CONTENTS;

*Close SQL Plus:*

exit

# Install SQL Server Database

/globant.lab.tripwire.com/globant\_fileshare/TE\_installer\_guides/te\_install\_and\_maint\_guide.pdf (page 16)

Open SQL Server Management Studio

Connect with the following user:

User: sa

Password: B3ld3n2015

TraceName (Trace1)

Create database:

Right click in database

Select new database

Complete Database name (Is case sensitive)

Initial size:

1st row: 2000M

2nd row: 500M

Select options page in the left menu

Collation: Latin1\_General\_CS\_AI

Recovery Model: simple

Compatibility level:SQL Server 2008 (100)

Miscellaneous ANSI NULL DEFAULT: True

Click OK to finish creating database

Right click on the new database

Select new query: A new query window is opened at the right

ALTER DATABASE <database name> SET READ\_COMMITTED\_SNAPSHOT ON;

SELECT name, is\_read\_committed\_snapshot\_on FROM sys.databases WHERE name='<database name>';

Right click on Security page in the left menu

Select New login

Complete Login Name: <user name>

Select SQL Server authentication with as password (**Store the user and password for TE installation step**)

Complete Default database: <database name>

Complete Default language: english

Select Server Roles page in the right menu

Select public role

Click OK to finish user creation

User Permissions in Database:

Right click on the new database

Select Properties

Select Permissions page in the left menu

Select the following Explicit permissions:

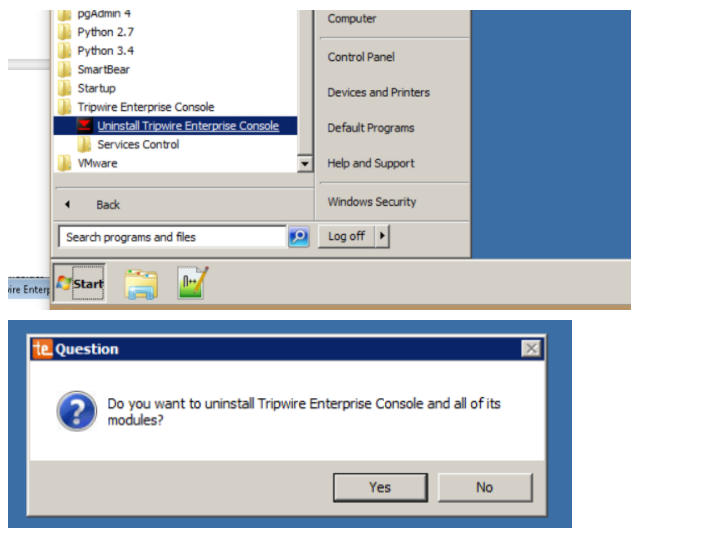
\*Connect \*Delete

\*Create Procedure \*Insert

\*Create Table \*Select

\*Create View \*Update

# Uninstall TE

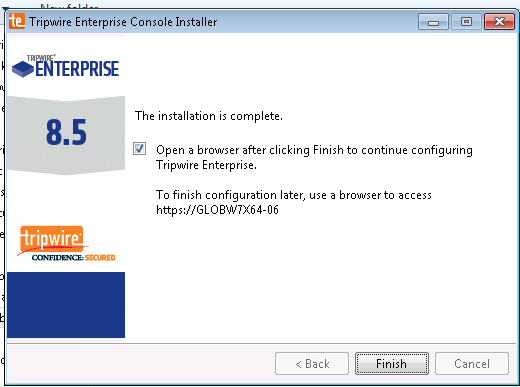
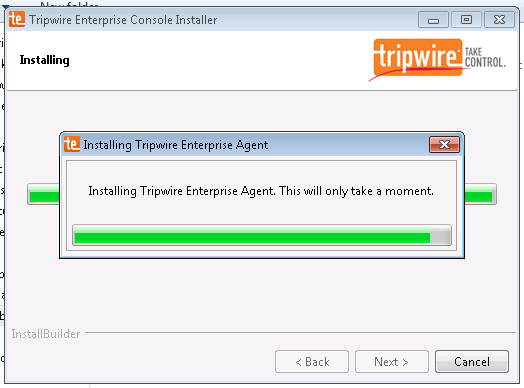
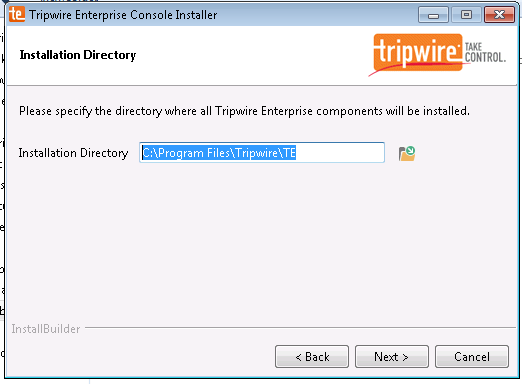
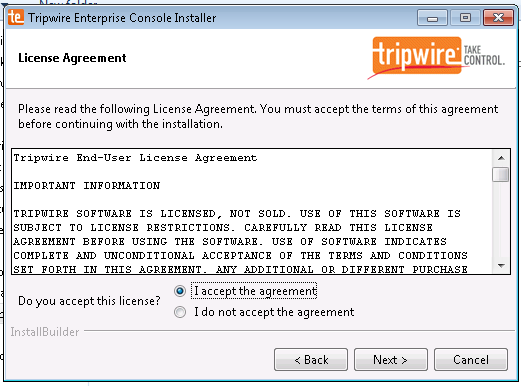


# 

# Install TE

## **1) Execute installer**

Use the TE installer to install the latest version of TE



Linux environment:

**Use sudo for permissions before any command**

Use mounts to copy the installation file to Linux: sudo mount -t cifs <installation file directory path> -o username=<installation server user name>,password=<installation server password> <Linux installation directory path>

e.g.: sudo mount -t cifs //globant.lab.tripwire.com/globant\_fileshare/ForTato -o username=mvillar,password=clave43l# /home/mvillar/etc

Copy the installation file from the mounts directory in Linux to the local installation directory in Linux (The directory contains the installation script file and java)

e.g.: cp /home/mvillar/mounts/fileshare/install-server-linux-amd64.bin /usr/local/msplett/install-server-linux-amd64.bin

In the same installation directory install Java and create the script

Installation Script:

Create an install.sh with the following information:

#!/bin/bash

export JAVA\_HOME=/usr/local/msplett/jdk1.8.0\_45

Execute the script: ./<installation file name.bin> --ignoreOSVersion true e.g. ./install-server-linux-amd64.bin --ignoreOSVersion true

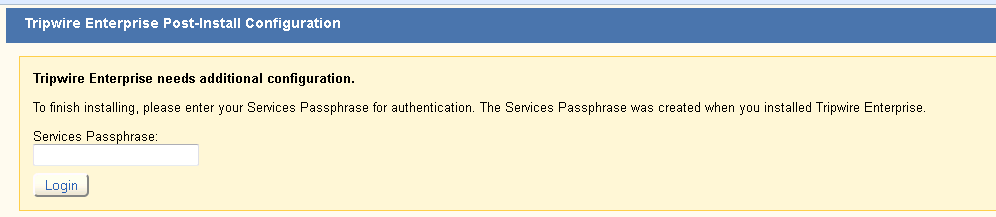
Choose option [2] and then accept all defaults

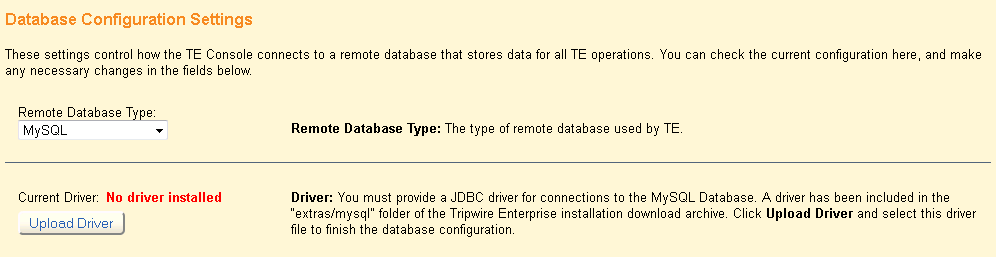
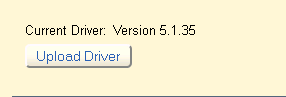
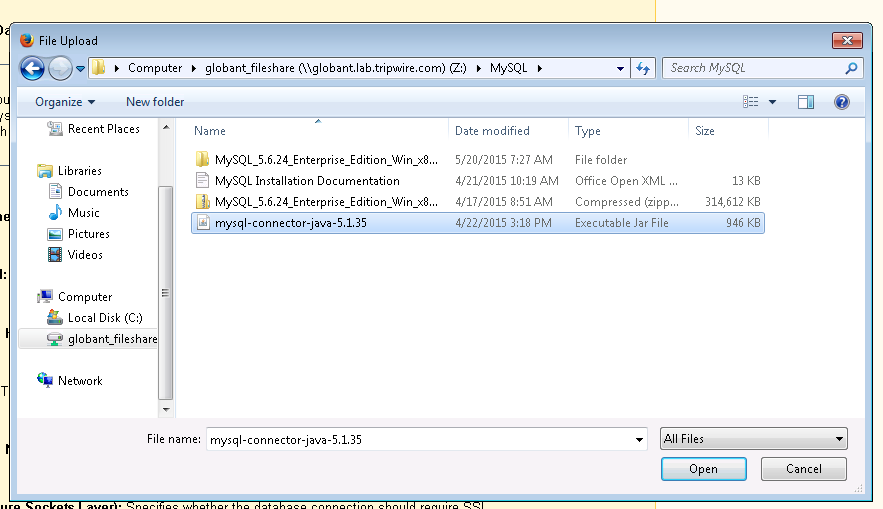
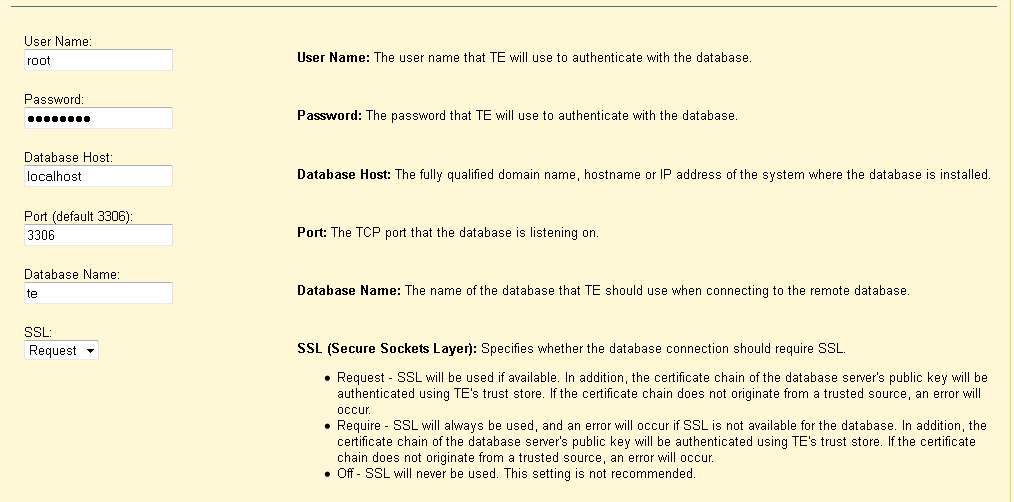
Installation Directory: /usr/local/tripwire/te

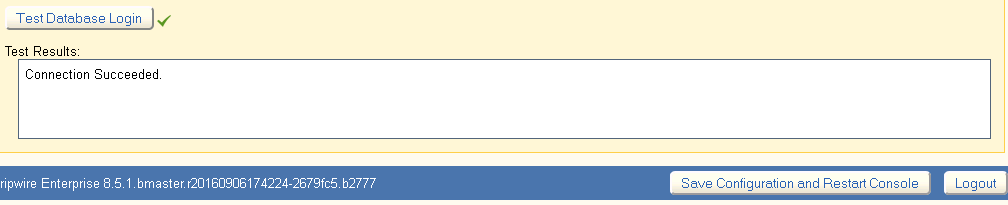
Uninstall directory: /usr/local/tripwire/te/server/uninstall (to run uninstall ./uninstall)

## **2) Configuration steps**

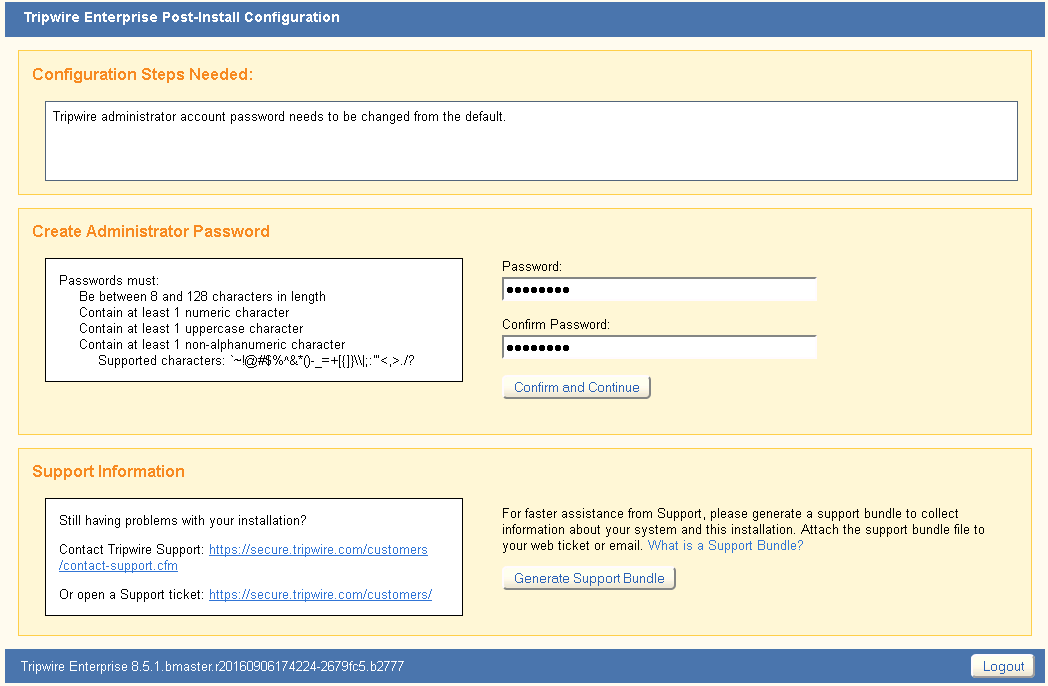
1. Go to <https://localhost>



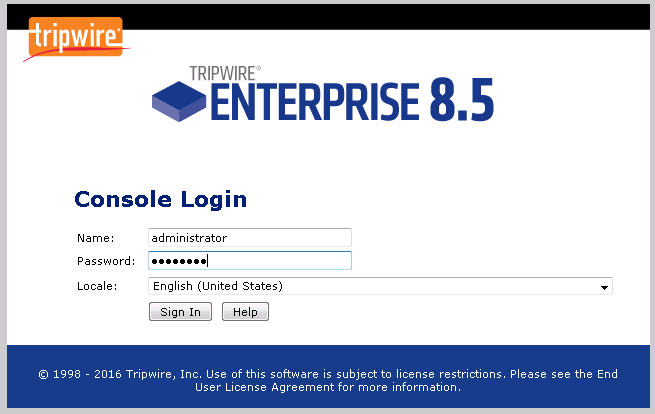
1. Put *Service Passphrase*
   1. Default Passphrase: ***TEServicesPassphrase***
2. MySQL Configuration
   1. Select the JDBC connector found at Z:\MySQLL 
   2. Fill the data necessary to install TE:  
      Password: root password - Passw0rd 
3. Click on “Check Database connection”



1. Save Configuration and Restart Console
2. Wait on “Loading page”
3. Set TE Administrator user password - Password1@



8. Enter as administrator with the password you created



9. FastTrack screen appears

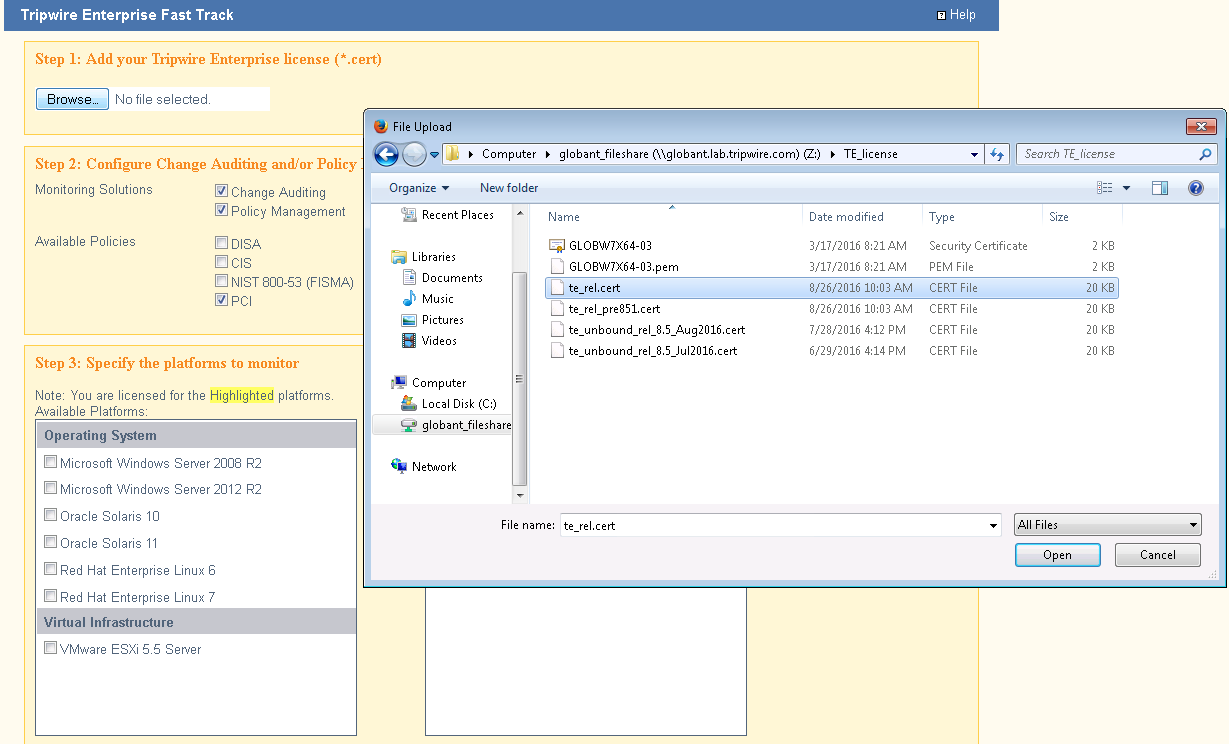


* 1. If you don’t want to add any additional configuration → Cancel → Confirm Cancel
  2. If you want to add additional configuration → Add data → Continue

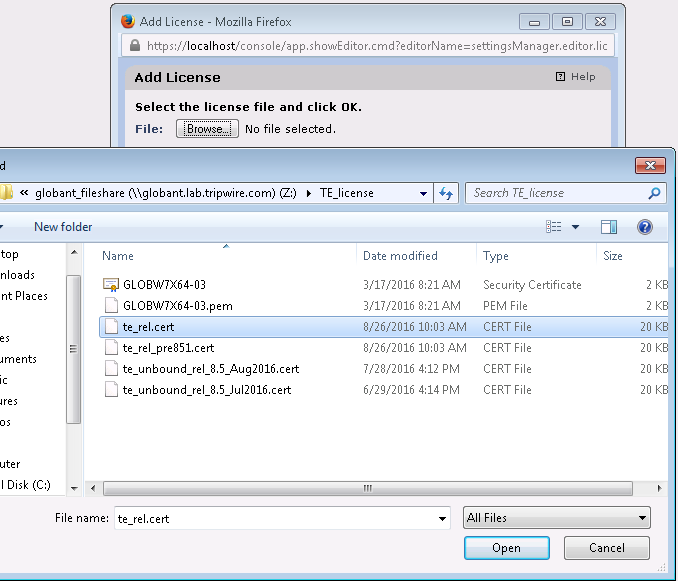
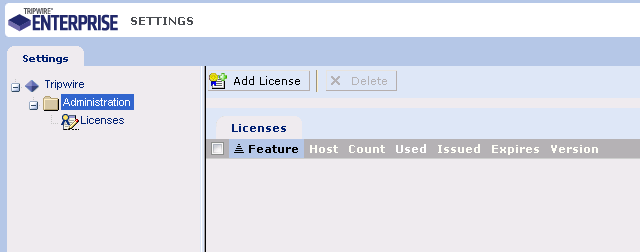
## **3) TE License Configuration**

Location of the licenses: **Z:\TE\_license**.

* If you continued to TestTrack configuration, in Step 1 you need to select TE License:
* If the license is expired (most likely), then cancel



* If you cancelled the configuration, you will be taken to TE Console, where you will need to Add License



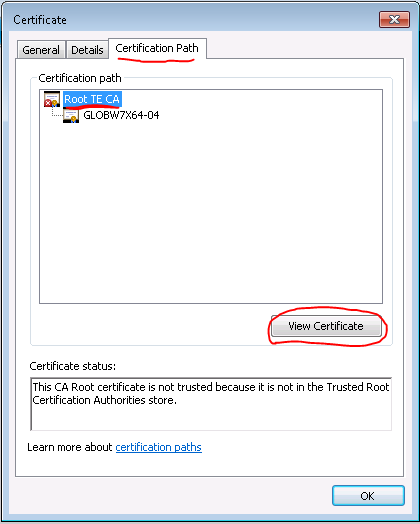
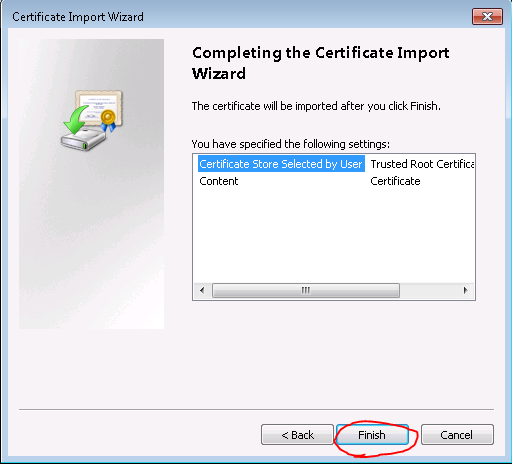
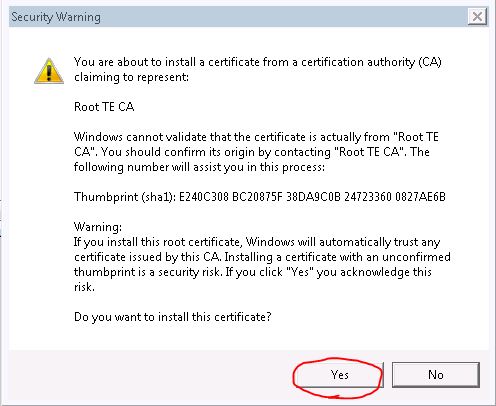
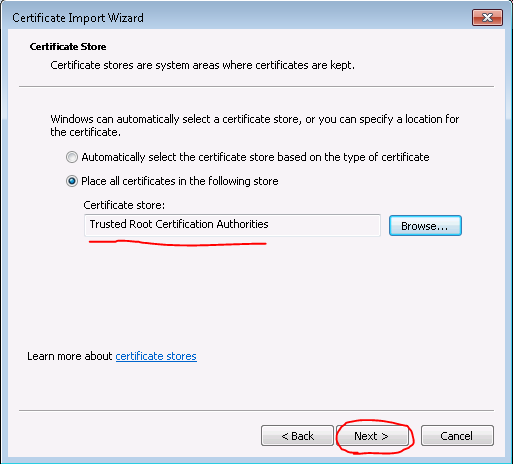
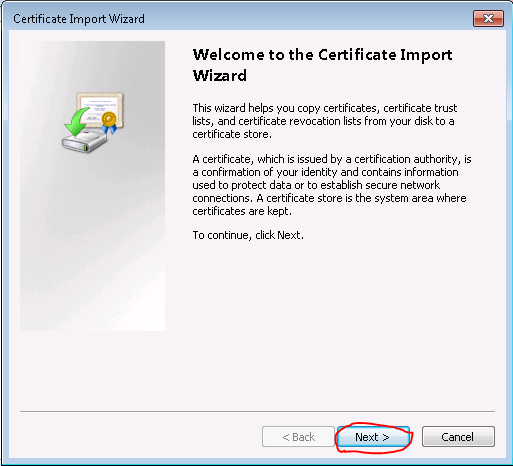
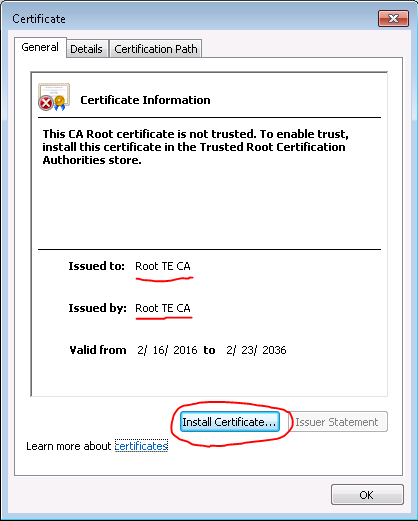
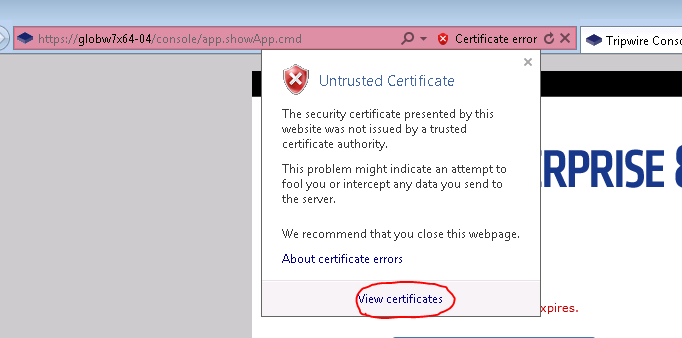
Additionally extend the license by executing on the db and restarting.

***update license set expires = '2030-12-12 01:01:01'***

# Certificate configuration

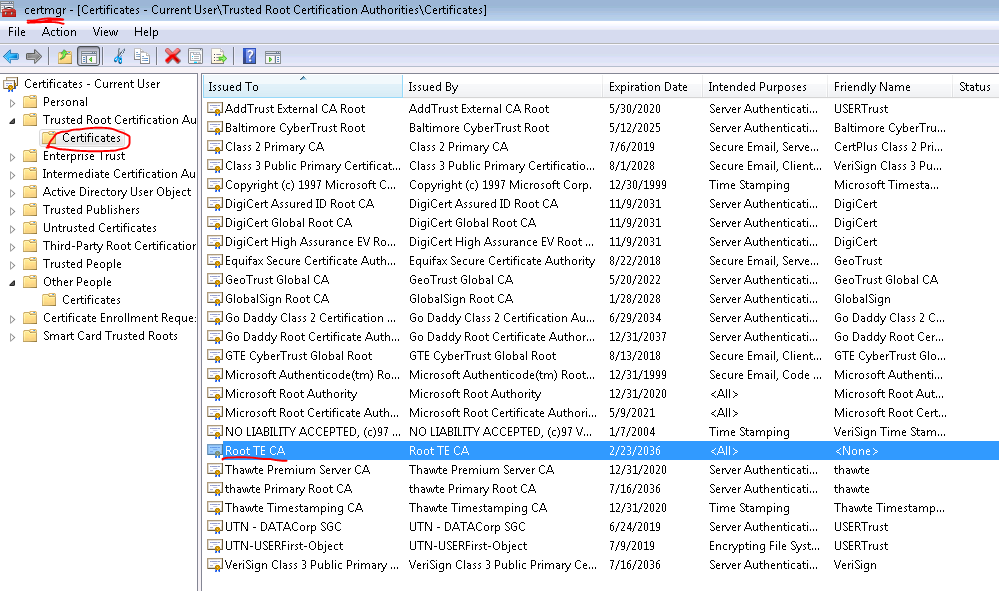
**Add the certificate in the Trusted Root Certification Authorities**

**(En IE)**

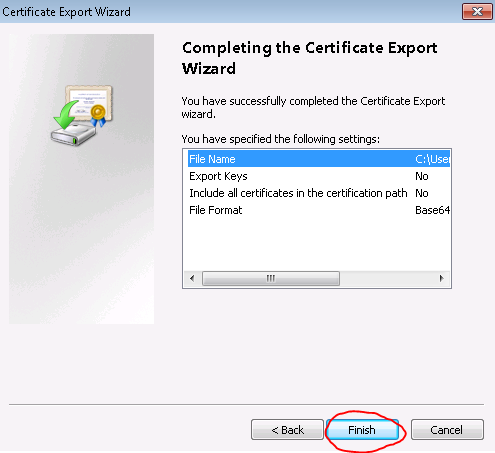
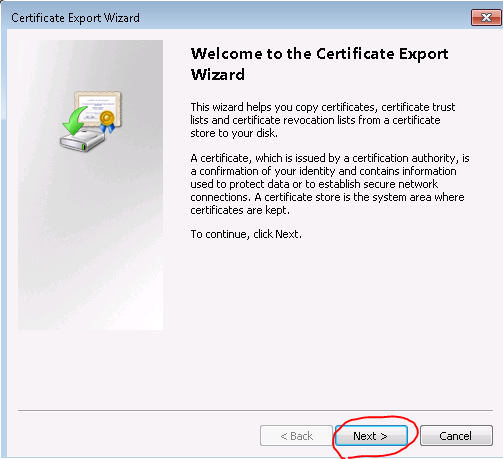
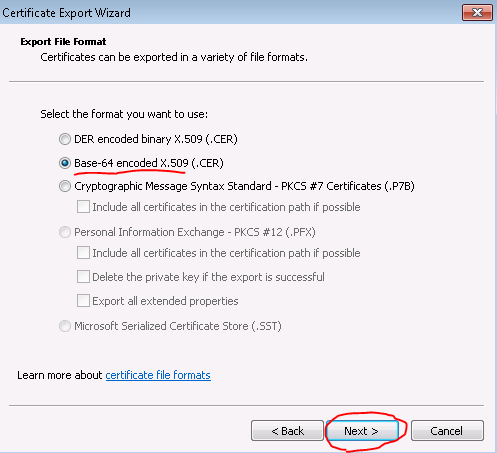
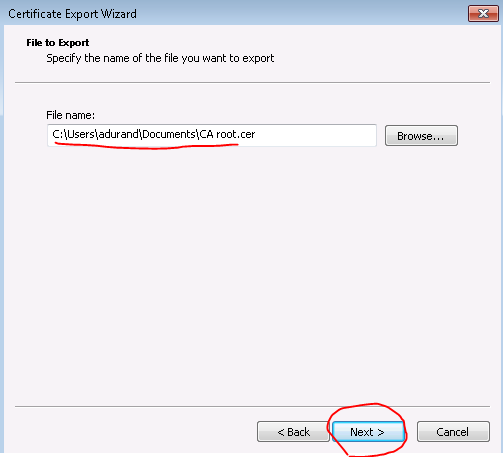


**Get the certificate from Trusted Root Certification Authorities**

In Windows, Start > Search > certmgr.msc.



Right click on **Root TE CA** > All tasks > Export ...



Open the certificate in notepad



Add the certificate content of TE to the **cacert.pem** file of certifi (Python library). Location of certifi: PYTHON\_HOME\Lib\site-packages\certifi\cacert.pem

# Configure TE properties

(Stop the Service before modifying the following files, and Start them when finished)

**Verify** at the end of TE installation that the following is configured:  
In C:\Program Files\Tripwire\TE\Server/data/config/server.properties:

tw.database.type=mysql

tw.database.host=(mysql-host-name)

tw.database.port=3306

tw.database.path=(mysql-te-database-name)

tw.database.user=(mysql-user-name)

tw.database.passphrase=(mysql-user-passphrase)

**Add** the following property into *C:\Program Files\Tripwire\TE\Server/data/config/server.properties* :

# Enables TE-REST-API:

tw.rest.api.enabled=true

#Enables SAL support

tw.securityAuditLog.enabled=true

tw.dev.rest.api.command.requests.purge.time=15000

tw.dev.rest.api.command.requests.purge.timeout=15000

#Edits PageLimit parameter

te.rest.api.maxPageLimit= 1000

## **Configuration of remote debugger**

Set the following options in *C:\Program Files\Tripwire\TE\Server/bin/server.conf*

wrapper.java.additional.**41**=­Xdebug

wrapper.java.additional.**42**=-agentlib:jdwp=transport=dt\_socket,server=y,suspend=n,address=5005

\*(change to the following number according to the pre existing lines)

## **Add grant permissions into JAAS Policy**

Edit file *C:\Program Files\Tripwire\TE\Server\Server\data\security\jaas.policy*

Add the following at the end of the file:

grant

{

permission java.security.AllPermission;

};

# Cloning the project repositories from GitHub

Open **Git Bash** with administrator rights

To set your credentials on Git Bash so it does not ask you for them every time:

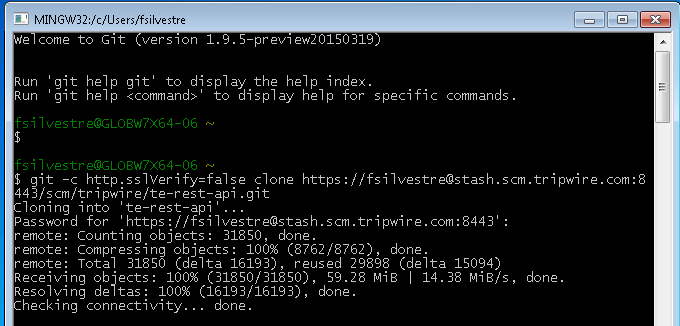
git config --global http.sslVerify false

git config --global user myusername

git config --global password myemail

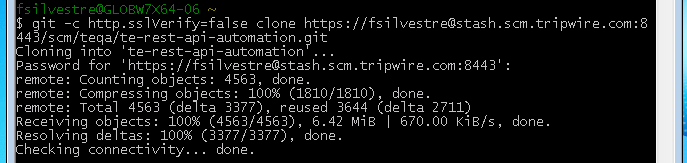
**Te-rest-api:**

git clone https://fsilvestre@stash.scm.tripwire.com:8443/scm/tripwire/te-rest-api.git



**Te-rest-api-automation:**

git clone https://fsilvestre@stash.scm.tripwire.com:8443/scm/teqa/te-rest-api-automation.git



\*(replace with own username)

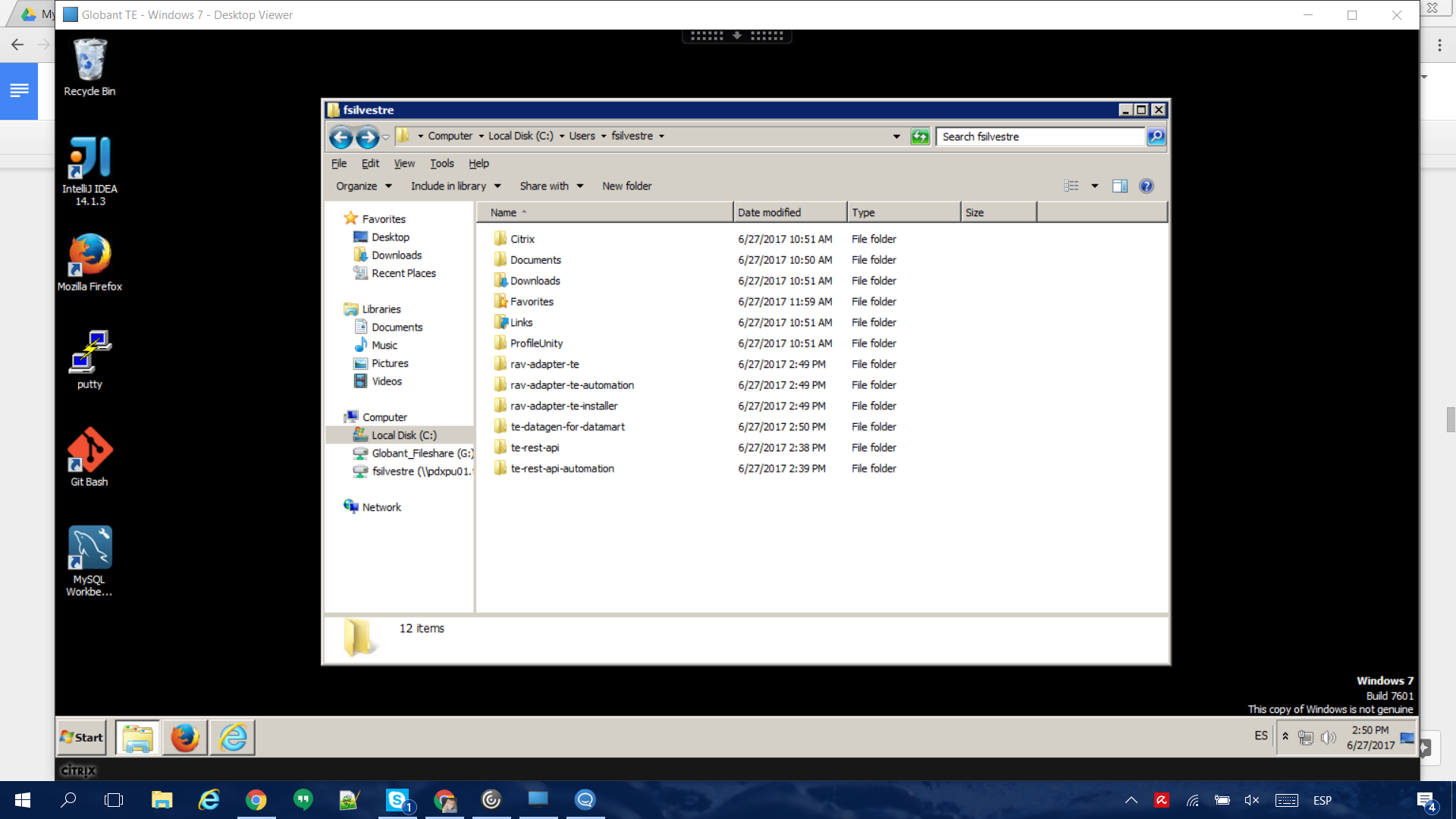
**Rav-adapter-te**: git clone [https://fsilvestre@stash.scm.tripwire.com:8443/scm/tripwire/rav-adapter-te.git](about:blank)

**Rav-adapter-te-automation**: git clone [https://fsilvestre@stash.scm.tripwire.com:8443/scm/tripwire/rav-adapter-te-automation.git](about:blank)

**Rav-adapter-te-installer**: git clone [https://fsilvestre@stash.scm.tripwire.com:8443/scm/tripwire/rav-adapter-te-installer.git](about:blank)

**Te-datagen-for-datamart**: git clone [https://fsilvestre@stash.scm.tripwire.com:8443/scm/tripwire/te-datagen-for-datamart.git](about:blank)

Now you should have the six projects on your local directory.



# Artifacts

Gradle dependencies and artifacts produced by our build systems are stored in an Artifactory instance. A special instance has been set up for this project.

<https://artifacts-stage.scm.tripwire.com:8443>

Login and browse around to get a feel for what is available. We do not let our build systems pull resources directly from maven-central or other internet locations. Instead we stage our own Maven repository for use by our Gradle based build systems.

Before being able to perform a build, you will need to setup your artifactory password in

C:\Users\%USERPROFILE%\.gradle\gradle.properties

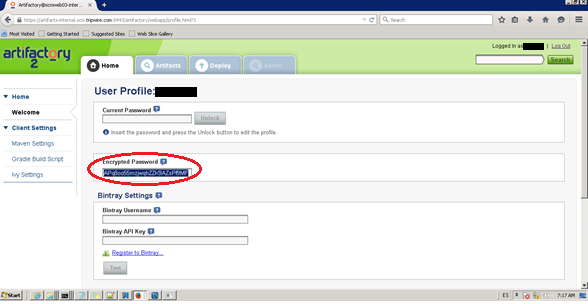
Add the following lines:

**artifactoryUser=<username>**

**artifactoryPass=<encrypted-password>**

The encrypted-password can be fetched from the artifacts-stage server, (<https://artifacts-internal.scm.tripwire.com:8443/>), if you go to your profile by clicking on your username in the upper right. You need to select unlock password and copy it to the gradle.properties file.

**\*Note: you will need to do this every time you change your Tripwires account password.**



## Build Projects

For each of the cloned projects, run

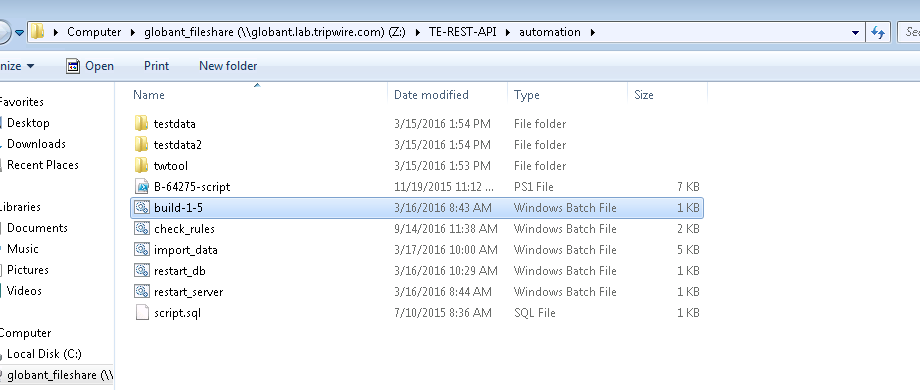
> **gradle clean build**

\* Some of the projects may fail with the command above, try running **gradlew** instead, and as a last resort add **--offline** to the command (this is necessary when the artifactory server is down).

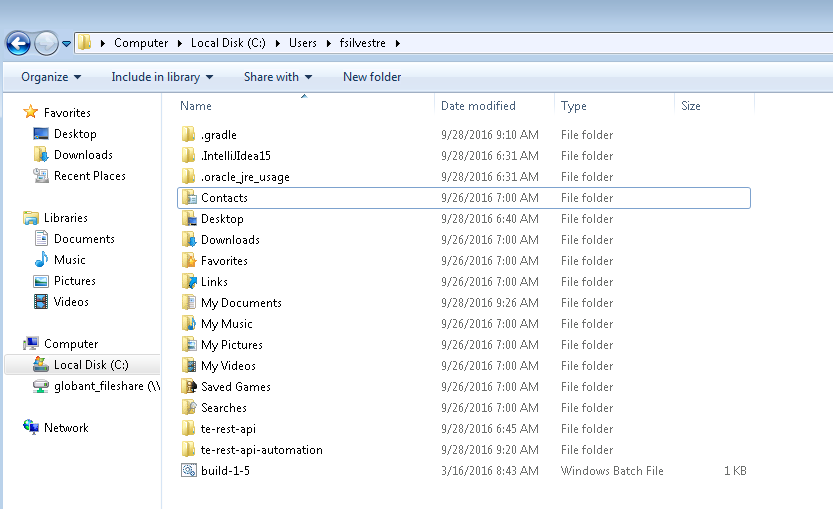
**t**

**Build TE-Api Script**

Go toZ:\TE-REST-API\automation, and copy the following script: **build-1-5**



**Paste it in your Users folder:**



Edit it with Notepad, and update the rest api version, and the war filename (if needed)

**@echo off**

**cd %~dp0\te-rest-api**

**call gradle clean war**

**cd %TW\_HOME%\webapps**

**del "te-rest-api.war"**

**copy "%~dp0\te-rest-api\restapplication\build\libs\te-rest-api-1000.1.11.0.bdevelop.r99999999999999.war" "%TW\_HOME%\webapps\te-rest-api.war"**

**echo "Stopping te server"**

**call "%TW\_HOME%\bin\twservices.cmd" stop**

**echo # Cleaning TE api folder**

**RMDIR /S /Q "%TW\_HOME%\data\tomcat\webapps\api"**

**echo "Starting te server"**

**call "%TW\_HOME%\bin\twservices.cmd" start**

Save, and run build-1-5.

# Intellij installation

**Intellij location:** \\globant.lab.tripwire.com\globant\_fileshare\Intellij\_IDEA\_14.1\_Ultimate

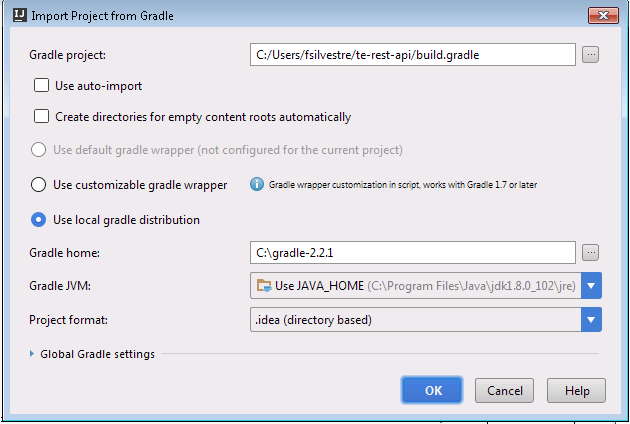
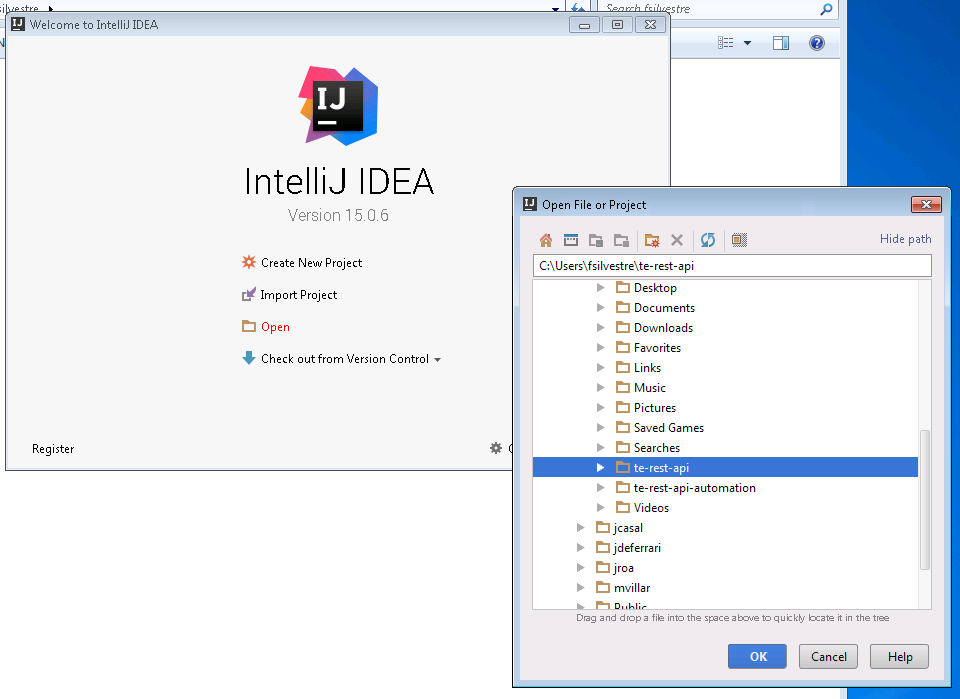
Install this version and not the latests, because it does not support the python plugin we use.

IntelliJ licences location for the dev team: G:\Jetbrains-Intellij\globant-intellij-licenses.txt

## Import projects to IntelliJ

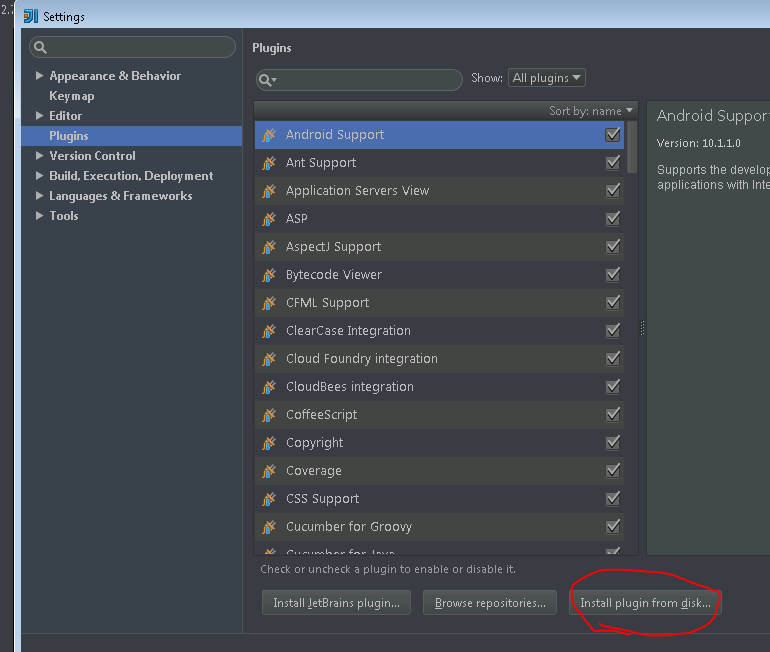
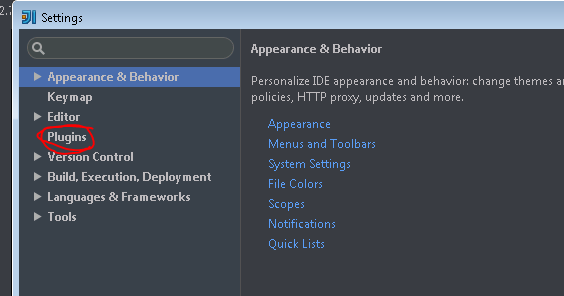
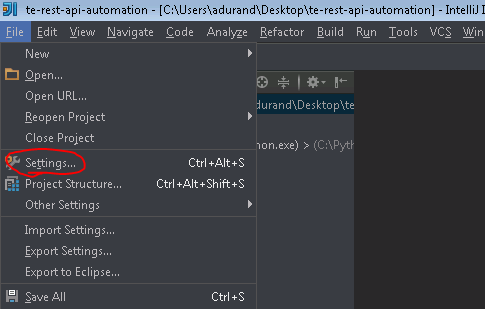
TE-REST-API:

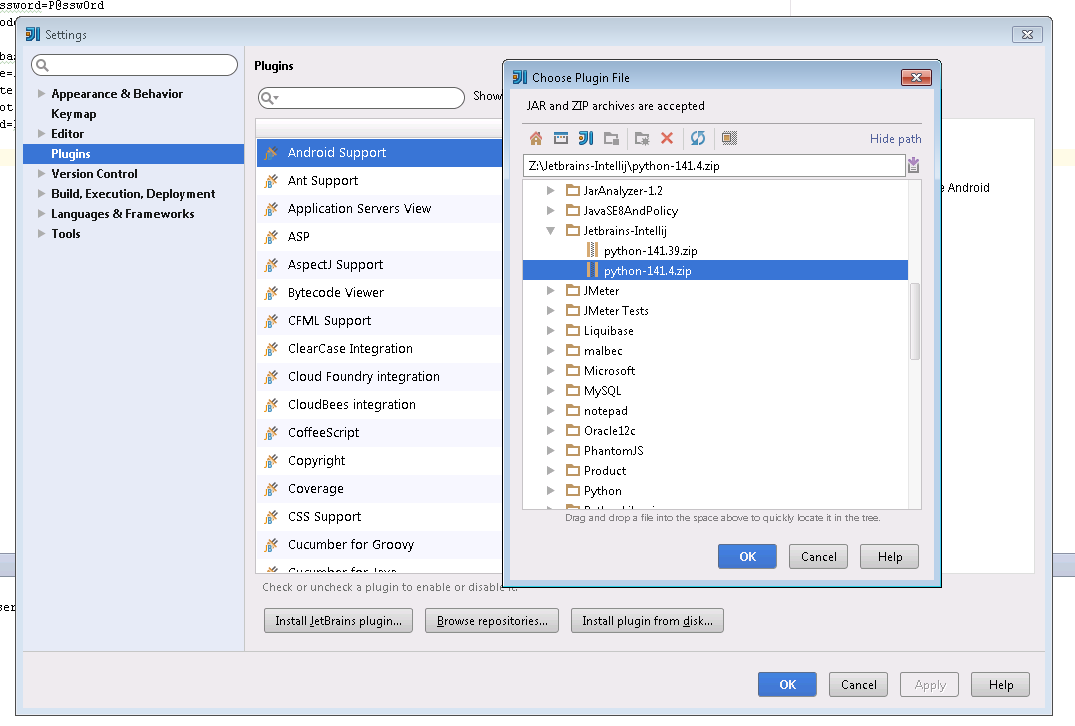
Open Project and select te-rest-api folder



Automation Project:

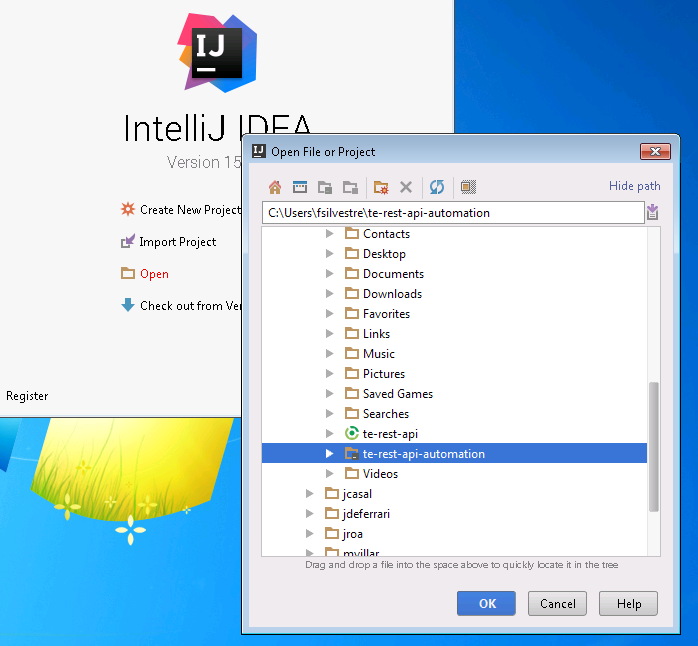
**Install Python plugin:**



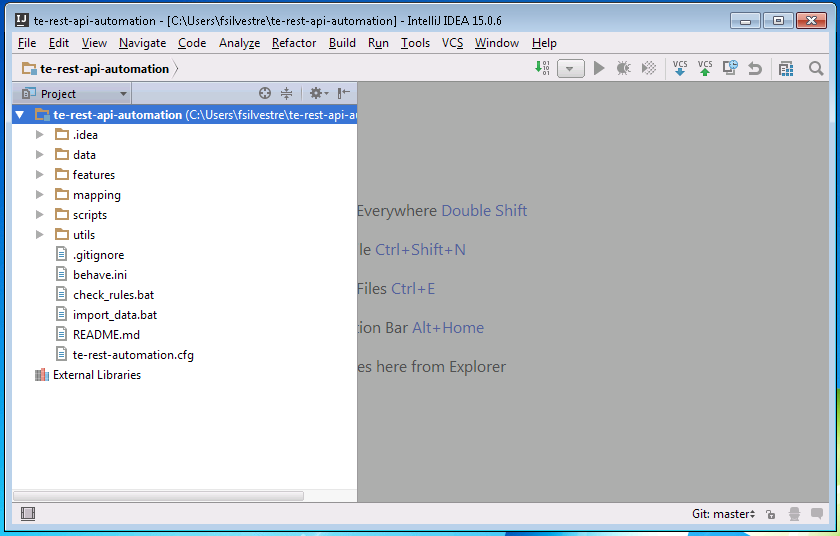


TE-REST-API-AUTOMATION:

Open Project, and select **te-rest-api-automation** project folder:

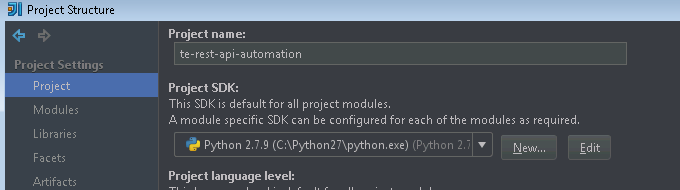


You should see this Workspace: (if you don’t install the python plugin, set the project SDK, and try **importing** the project)

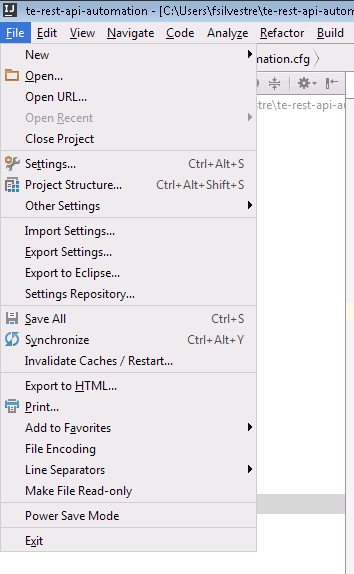


**Project configuration:**

File → Project Structure → Project SDK → Set the location of Python SDK



Import Settings:



Import IntelliJ settings (indentation) from **te-rest-api/codeStyle\_settings.jar**

# Data population - Setup Automation Data

First, update the file **te-rest-automation.cfg** in the automation project with information of your database and TE configuration.

[teserver]

hosturl=https://**GLOBW7X64-04**

tw\_home=C:\Program Files\Tripwire\TE\Server

adminpassword=**Password1@**

local\_node=**GLOBW7X64-04.dev.lab.tripwire.com**

database=mysql

[tedatabase-mysql]

**hostname=localhost**

**dbname=tripwire**

**user=root**

**password=Password1@**

[tedatabase-oracle]

hostname=10.248.169.18

dbname=orcl.dev.lab.tripwire.com

user=ouserte853

password=oclave853

And **gradle.properties** with the administrator password and local node name.

Run:

**Gradle setupdata** - this tasks does the following:

1. Imports data from xml files located on data/toImport folder:
   1. Settings
   2. Rules
   3. Nodes
   4. Actions
   5. Policies and Policy Tests
   6. Tasks

Try importing Tasks manually if the import fails and rerun the gradle setupdata command

1. Checks rules to local node:
   1. element version audit,
   2. element version content,
   3. driver status,
   4. windows compliance configuration.
2. Updates licences.
3. Updates binary file cflag
4. Assign custom tags to nodes
5. Creates element versions
6. Creates waivers

And finally it restarts TE Server.

If your node is **not** listed on Nodes tab under “Root Node Group” on the TE Console as shown below:

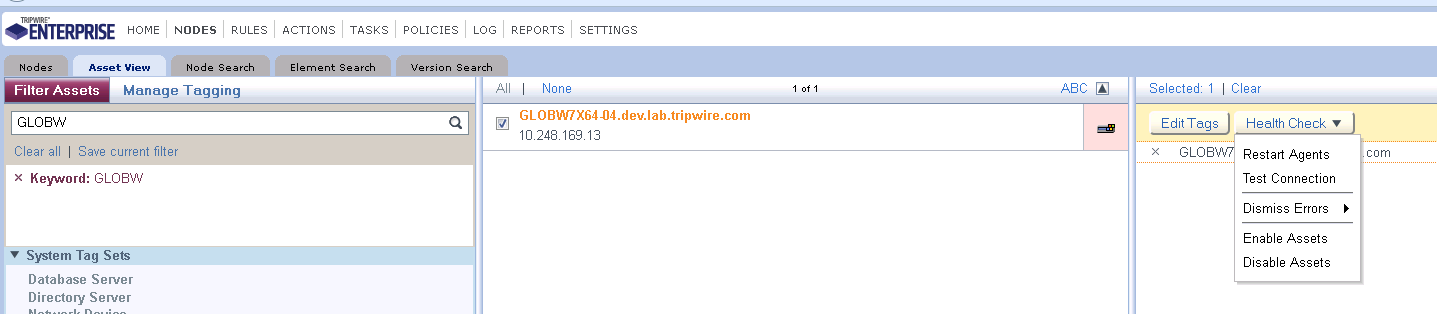
1) Go to **Asset View**

2) Search for “GLOBW”

3) Check your node

4) Under **Health Check**, select **Restart Agent**

5) When it finish, restart TE Server

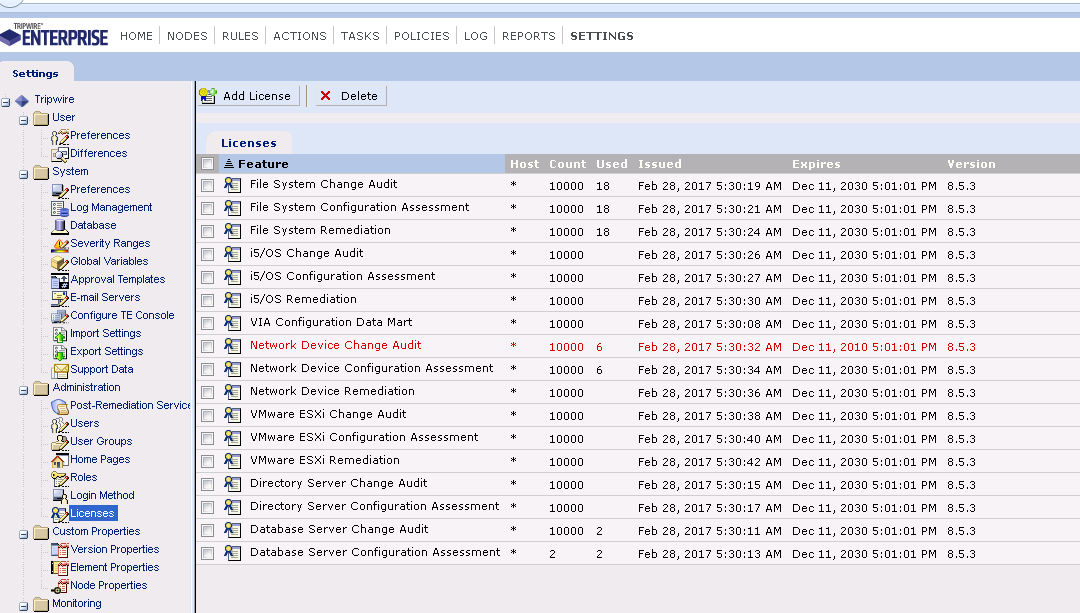


6) Search for your Node, and **Link** it to **Root Node Group**

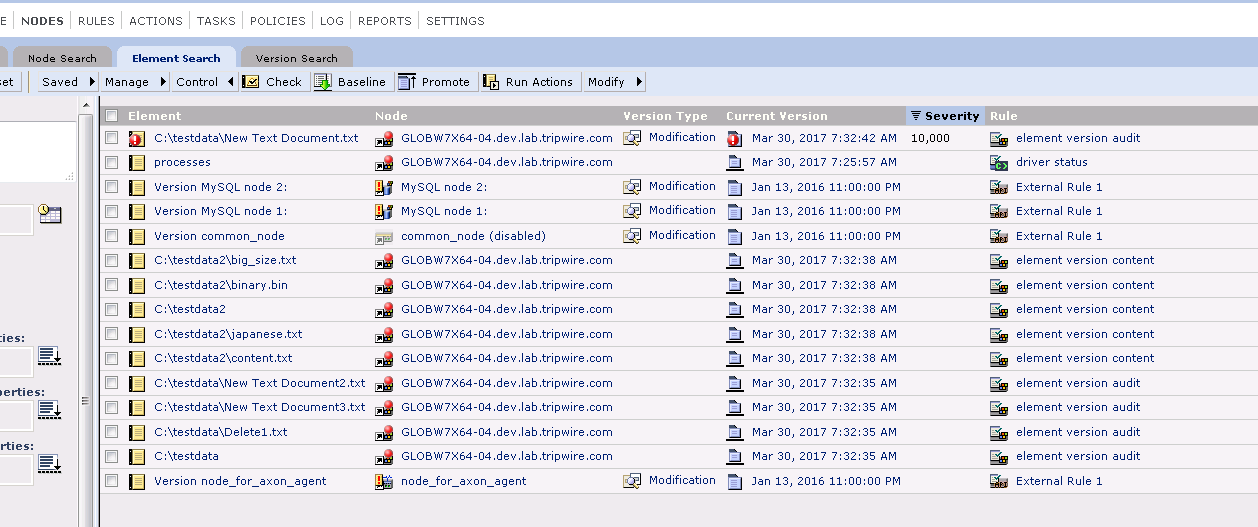
Finally, your TE data should be as follows:

(\*Note: You could have more data if you run some of the scripts more than once)

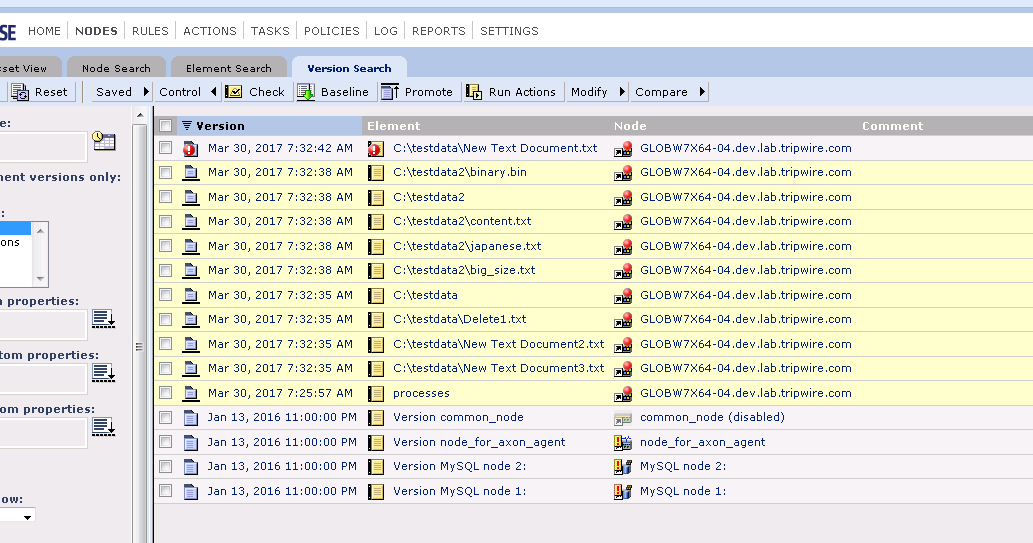
Licenses:



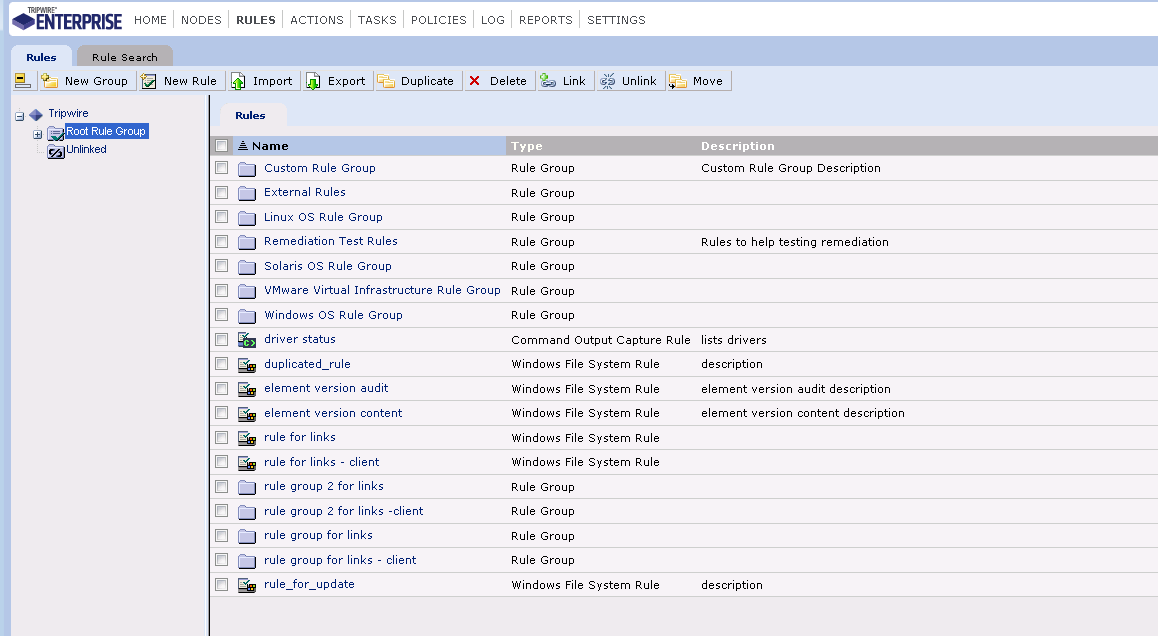
Elements:



Versions:



Rules:



To ensure that your environment has been set up correctly, run the full suite of automations tests and verify that there are no errors.

# Gradle tasks

------------------------------------------------------------

All tasks runnable from root project - The Automation Tests for the REST-API

------------------------------------------------------------

Default tasks: runtests

Automation tasks

----------------

Importdata - Imports necessary data to run Automation Tests

restartserver - Restarts TE Server

runtests - Run automation tests. Properties: -PFEATURE, -PFORMAT, -PTAGS, -PNTAGS, -POTHER

setupdata - Setup all necessary data to run Automation Tests & Restarts TE Server

Build Setup tasks

-----------------

init - Initializes a new Gradle build. [incubating]

Help tasks

----------

buildEnvironment - Displays all buildscript dependencies declared in root project 'te-rest-api-automation'.

components - Displays the components produced by root project 'te-rest-api-automation'. [incubating]

dependencies - Displays all dependencies declared in root project 'te-rest-api-automation'.

dependencyInsight - Displays the insight into a specific dependency in root project 'te-rest-api-automation'.

dependentComponents - Displays the dependent components of components in root project 'te-rest-api-automation'. [incubating]

help - Displays a help message.

model - Displays the configuration model of root project 'te-rest-api-automation'. [incubating]

projects - Displays the sub-projects of root project 'te-rest-api-automation'.

properties - Displays the properties of root project 'te-rest-api-automation'.

tasks - Displays the tasks runnable from root project 'te-rest-api-automation'.

Other tasks

-----------

wrapper - Builds the gradlew wrapper

# Project Structure

* The "data" folder contains all the xml files that stores the data needed to be imported into the TE.
* The "scripts" folder contains all the scripts needed to be executed before to run the tests.
* The "features" folder contains all the files of tests. Into this folder also exists another one called "steps". The "steps" folder contains all the files to map the steps of each feature and execute the appropriate script for each.
* The "mapping" folder contains all the files to map data of each feature. For example: urls, elements of an HTML page, etc.
* The "utils" folder contains utilities used for the tests.

# Files of the project

## Features (features/rest-api and features/client)

The features contains all definition of tests. In the section of sequence diagrams you can see examples of how this works.

## Steps (features/steps)

The project has different type of steps. Each one could be used in different situations. The folder “client\_steps” contains Python files with definitions of steps that only must be used for tests related with the Python client generated by Swagger Codegen. The folder “rest\_steps” contains Python files with definitions of steps that only must be used for tests related with a direct interaction with the REST API. The folder “selenium\_steps” contains Python files with definitions of steps that only must be used for tests related with the interaction with the browser UI. The folder “shared\_steps” contains Python files with definitions of steps that could be used in any of the previous scenarios.

## Environment (features/environment.py)

This file defines code to run before and after certain events during your testing.

## Mapping (features/mapping.py)

This file contains some variables used during the execution of tests. For example: passwords of different users, the name of the local node, the host, etc.

## Utils (utils)

This folder contains utilities used in the project. Here you can see a description of each file:

* **base36.py:** this Python file contains two methods: encode (converts an integer to base36) and decode (converts a string to integer).
* **database.py:** provides three operations open\_conection, execute\_query and close\_conection to interact with a MySQL database.
* **json\_comparator.py:** contains methods to compare json objects.
* **queries.py:** contains MySQL queries to execute during the execution of tests (Mostly to get readable keys).
* **string\_to\_readable\_key.py:** contains only one method that replace a string for readable key. The readable keys should be in the context variable of Behave.

## Behave (behave.ini)

Configuration file for Behave.

## Configuration file (te-rest-automation.cfg)

Configuration file for the project. This file allows to set variables related with the TE Server and the TE database.

**Properties used:**

* **[teserver] hosturl:** url of your TE.
* **[teserver] tw\_home:** directory of the TE server. This property is used to start and stop the server using .bat files located in the bin folder.
* **[teserver] adminpassword:** administrator password of your TE. Is used to make requests using the administrator user.
* **[teserver] local\_node:**  name of your local node.
* **[tedatabase] hostname:** hostname of your database used for TE.
* **[tedatabase] dbname:** name of your database used for TE.
* **[tedatabase] user:** database user (Should have all the permissions for the TE database).
* **[tedatabase] password:** user password.

**Example of configuration file:**



# 

# Running the projects

### Automation project:

To run this project you must be in the root directory and run the following command as administrator (required to start and stop the TE Server in one test):



To run an specific feature you must be in the root directory and run the following command as administrator (required to start and stop the TE Server in one test):



To exclude the slower tests from the execution. This command also exclude the tests where is needed execute the command as administrator:



Running the tests with gradle:

(As runtests is the default task, there is no need to include it on the command)

***Gradle -PFEATURE=homepages/homepages -PNTAGS=slow -PTAGS=Sprint53 -PFORMAT=progress3***

Without tags:

*behave features\rest\_api -f progress3 --tags=~@Tags > C:\Users\fsilvestre\desktop\features.txt*

With tags:

*behave features\rest\_api -f progress3 --tags=@Tags > C:\Users\fsilvestre\desktop\features.txt*

## Databases - **Oracle SQL Developer**

Automation Tests can be runned on Oracle and MySql databases.

Change Tripwire\TE\Server\data\config\**server.prop** to change between dbs:

*tw.database.type=****mysql***

*tw.database.host=localhost*

*webserver.hostname=GLOBW7X64-04.dev.lab.tripwire.com*

*space.bootstrapables=*

*tw.server.host=GLOBW7X64-04.dev.lab.tripwire.com*

*java.rmi.server.hostname=GLOBW7X64-04.dev.lab.tripwire.com*

*tw.scp.enabled=false*

*tw.database.path = te*

*tw.database.user = root*

*tw.database.port = 3306*

*tw.database.mysql.ssl = request*

*------------------------------------------------------------------------------------*

*tw.database.type=****oracle***

*tw.database.host=10.248.169.18*

*tw.database.path=orcl*

*tw.database.user=ouserte853*

*tw.database.port=1521*

*webserver.hostname=GLOBW7X64-04.dev.lab.tripwire.com*

*space.bootstrapables=*

*tw.server.host=GLOBW7X64-04.dev.lab.tripwire.com*

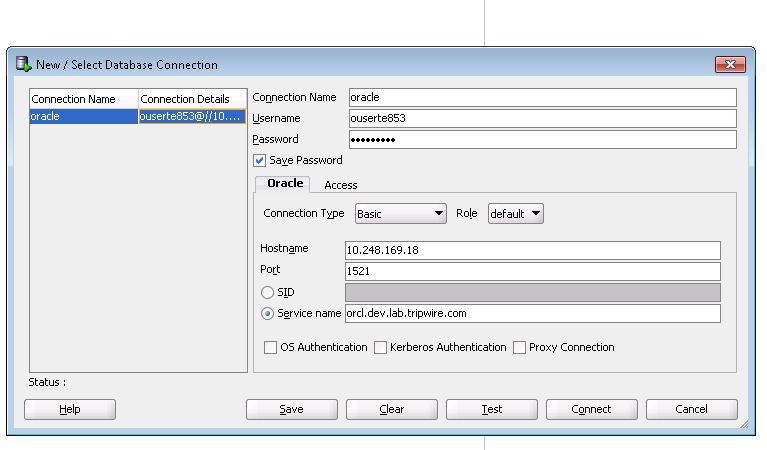
*java.rmi.server.hostname=GLOBW7X64-04.dev.lab.tripwire.com*

*tw.scp.enabled=false*

*Te-rest-automation.cfg* contains a variable **database** to set where tests will run.

Values: **mysql** or **oracle**

Connect to AT Oracle db using **Oracle SQL Developer**



Or using **sqlplus**:

*Sqlplus ouserte853/oclave853@10.248.169.18:1521/orcl.dev.lab.tripwire.com*

### Te-Rest-Api

Build the project:

*Gradlew clean build*

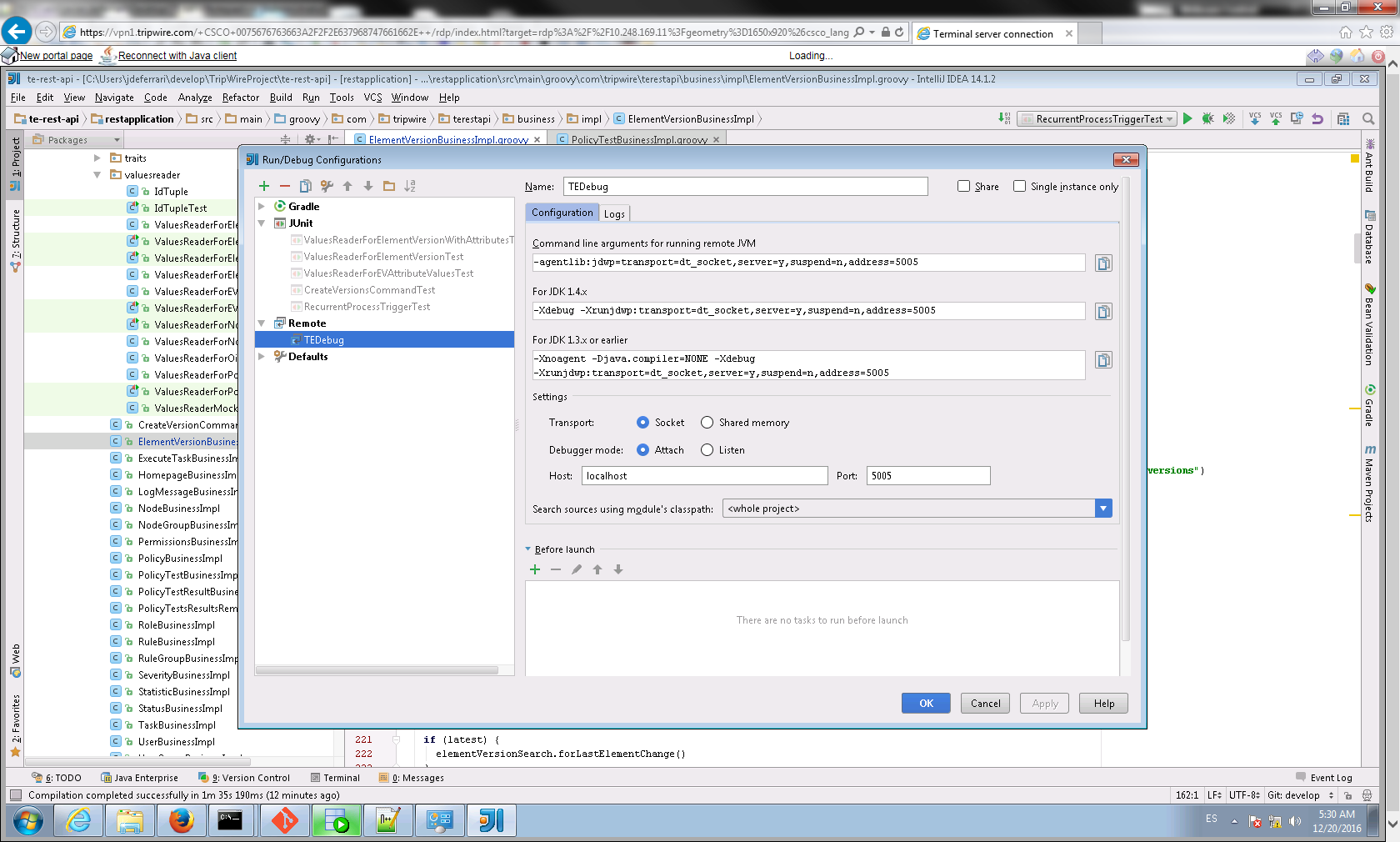
Run swagger codegen:

*gradle asciidoctor*

## Set Up the Debugger for Te-Rest-Api

IntelliJ > Run > Edit Configurations…

* > Remote > Ok



# Update TE from unsupported versions

Use a ‘backup’ & ‘restore’ mechanism instead.

1) Stop the TE services

2) Manually backup this file (if present): .../TE/Server/data/security/site.data

3) Open a shell (shift+click -> Open command window here), and go to the TE/Server/bin directory and run:

tetool.cmd backup —passphrase <services-passphrase> -v C:\config.bak

4) Uninstall old TE

5) Install new TE

6) Stop the TE services

7) Restore the backed up configuration, by again opening a shell, go to the TE/Server/bin directory, and run:

tetool.cmd restore —passphrase <services-passphrase> -v C:\config.bak

8) Restart the TE services

# Fixes for detected errors and tips

(Linux) Create a group: groupadd -g 1000 <group name> e.g.groupadd -g 1000 tripwire

(Linux) Create a user: useradd -M -u 1001 -g <user name> <user password> e.g.useradd -M -u 1001 -g tripwire tripwire

(Linux) To see firewall configurations:

/sbin/iptables -L -v -n | less

/sbin/iptables -L INPUT -v -n | less

(Linux) To disable firewall so that TE can be opened in another VMs:

/etc/init.d/iptables save

/etc/init.d/iptables stop

chkconfig iptables off

LimitPosted content length of 21150812 exceeds limit of 20971520 (Import exceeds limit)

Include tw.webserver.maxUploadSize in TE/server/config/server.properties

Log directories:

C:\Program Files\Tripwire\TE\Server\data\log

(Linux) /usr/local/tripwire/te/server/data/log

Stop/Restart the server:

Access the start menu in windows. Select all programs/Tripwire Enterprise Console/Services Control

Services Stop/Services Start

(Linux) cd /etc/init.d

./twservices stop

./twservices restart

Invalid max\_allowed\_packet value detected. Expected: '1073741824', found: '4194304'.

Copy the following command in MySQL console: SET GLOBAL max\_allowed\_packet=1073741824;

# Links

## Globant

**Confluence para documentación**

<https://confluence.corp.globant.com/pages/viewpage.action?spaceKey=TPWOO1&title=Tripwire+-+Data+Api+Migration>

## Tripwire public

**Version One**

VersionOne is our project planning and tracking system.

<https://www14.v1host.com/Tripwire/>

We will be embarking on project Mulberry. The planning room is at:

<https://www14.v1host.com/Tripwire/PlanningRoom.mvc/Show/532418>

Please verify that you can browse the *Rewrite TED in java* epic and stories.

**Outlook Tripwire**

<https://mail.tripwire.com/owa/>

**MyPassword : para cambiar el password del usuario**

<https://apps.tripwire.com/mypassword/>

## Tripwire private (only accessed from Citrix or VDI)

**TestTrack : Sistema para manejo de defects / bugs**

<http://ttpro.lab.tripwire.com/cgi-bin/ttcgi.exe?command=LoginScreen>

**Artifactory : Repositorio interno de Tripwire para manejo de dependencias**

Gradle dependencies and artifacts produced by our build systems are stored in an Artifactory instance. A special

instance has been set up for this project.

Login and browse around to get a feel for what is available.

We do not let our build systems pull resources directly from maven-central or other internet locations. Instead we

stage our own Maven repository for use by our Gradle based build systems.

<https://artifacts-internal.scm.tripwire.com:8443/artifactory/webapp/home.html?1>

**TeamCity : Servidor de integración continua**

We use a JetBrains TeamCity CI server.

The projects group te-globant contains the te-rest-api project. Find and familiarize yourself with the te-rest-api

project and build results. You can see unit test failures, or build successes from here.

There are also builds for the Tripwire Enterprise product that are dependent on the te-rest-api artifacts.

i. The te-rest-api build is triggered by changes submitted to the Stash master branch.

ii. A TE build is triggered by successful builds of te-rest-api.

iii. Resulting builds of TE are published to the windows network share:

**//BUILDS.scm.tripwire.com/globant**

<https://teamcity.scm.tripwire.com/viewType.html?buildTypeId=TeRestApi_TeRestApi_Master>

**File Share directory**

\\globant.lab.tripwire.com\globant\_fileshare

**Cloudbolt**:

<https://cloudbolt.lab.tripwire.com>

## Technologies

General:

* [GitHub](https://www.atlassian.com/git/tutorials)
* [VersionOne](https://www.versionone.com/product/lifecycle/overview/)
* [Agile Methodology - Scrum](https://www.versionone.com/agile-101/what-is-scrum/)
* [TeamCity](https://www.jetbrains.com/teamcity/documentation/) - Continuous Integration Server
* [Jenkins](https://jenkins.io/) - automation server, Jenkins provides hundreds of plugins to support building, deploying and automating any project.
* [CloudBolt](http://docs.cloudbolt.io/) - Hybrid Cloud Delivery
* [Docker](https://docs.docker.com/get-started/) - Build, ship, and run distributed applications
* [WIX](https://www.firegiant.com/wix/tutorial/) - Windows Installers
* [RPM](https://rpmbuildtut.wordpress.com/) - Linux Installers
* [Bash Scripting Cheatsheet](https://devhints.io/bash)
* [IntelliJ Shortcuts](https://www.jetbrains.com/help/idea/mastering-intellij-idea-keyboard-shortcuts.html)

Data Bases:

* [MySQL](https://dev.mysql.com/doc/refman/5.6/en/)
* [SQL Server](https://docs.microsoft.com/en-us/sql/t-sql/queries/queries)
* [Oracle](http://www.oracle.com/technetwork/database/database-technologies/sql/overview/index.html)
* [PostgreSQL](http://www.postgresqltutorial.com/what-is-postgresql/) [Querying JSON](http://schinckel.net/2014/05/25/querying-json-in-postgres/)
* [Kafka](https://kafka.apache.org/intro) - Streaming Platform

TE Rest API: ([Development Guide](https://docs.google.com/document/d/1ueLeysmtDhthlz6rp5nd59NQCZws4qqMeWP9IbMO-YI/edit))

* [Groovy](http://groovy-lang.org/documentation.html): powerful, optionally typed and dynamic language, with static-typing and static compilation capabilities, for the Java platform aimed at improving developer productivity thanks to a concise, familiar and easy to learn syntax. It integrates smoothly with any Java program, and immediately delivers to your application powerful features, including scripting capabilities, Domain-Specific Language authoring, runtime and compile-time meta-programming and functional programming.
* [Gradle](https://guides.gradle.org/building-groovy-libraries/?_ga=2.100497175.881521126.1521225895-1895500056.1521225895): build automation tool focused on flexibility and performance. Gradle build scripts are written using a Groovy or Kotlin DSL.
* [Swagger UI](https://swagger.io/swagger-ui/): allows visualization and interaction with the API’s resources without having any of the implementation logic in place. It’s automatically generated from your Swagger specification, with the visual documentation making it easy for back end implementation and client side consumption.
* [JDBC](https://docs.oracle.com/javase/tutorial/jdbc/basics/index.html)
* [ESAPI](https://www.javadoc.io/doc/org.owasp.esapi/esapi/2.1.0.1)
* [Jersey](https://jersey.github.io/)
* [Guice](https://github.com/google/guice/wiki/GettingStarted)
* [Jacoco](http://www.jacoco.org/jacoco/trunk/doc/): Code Coverage Library
* [JMX](https://www.wikiwand.com/en/Java_Management_Extensions)
* [Shiro](https://shiro.apache.org/reference.html)
* [Liquibase](http://www.liquibase.org/documentation/index.html)
* [Logback](https://logback.qos.ch/manual/index.html)
* [JMeter](http://jmeter.apache.org/usermanual/index.html)
* [Unit Tests](https://www.tutorialspoint.com/groovy/groovy_unit_testing.htm)

TE Rest API Automation Tests Project: ([Development Guide](https://docs.google.com/document/d/1PcRcPTRuKhfuKyfOd4k_F4I4UcXSe_UGVk5z_kbK1ww/edit))

* [Behave](https://behave.readthedocs.io/en/latest/): Behavior-driven development (or BDD) is an agile software development framework. Uses tests written in a natural language style, backed up by Python code. [Tutorial](https://jenisys.github.io/behave.example/)
* [Gherkins](http://docs.behat.org/en/v2.5/guides/1.gherkin.html): Metalanguage used by BDD that describes expectations for any kind of application in an easily readable format.
* [Python](https://docs.python.org/2/tutorial/index.html): Python is a programming language that lets you work quickly and integrate systems more effectively. The version we use is 2.7.9. [Rec Read](https://learnpythonthehardway.org/book/)

Data Adapter ([Connect Guide](https://docs.google.com/document/d/1652KQxFh-rjmwSaPq240drjLetKjiaItp2qfO3gE_Bk/edit#heading=h.nxzvuzjduvif)) ([Configuration Items](https://docs.google.com/document/d/1GM6OkrE0RX7pzvZShGf7zAm7TZVxyR9agMAgbou800Q/edit)) ([Technical Specs](https://docs.google.com/document/d/1tdsxQ_ZtLEHFa2bhYQctPaE8mN1BwjVdl7OvFl8Vdlk/edit#heading=h.8kqilp9rvlkc))

* [Spring Framework](https://docs.spring.io/spring/docs/current/spring-framework-reference/)
* [Java](https://introcs.cs.princeton.edu/java/11cheatsheet/)
* [Concurrency](https://docs.oracle.com/javase/tutorial/essential/concurrency/) Rec: [Fer’s Tutorial](https://globant.acamica.com/cursos/379/paralelismo-y-concurrencia-en-java)

[Data Adapter Automation Strategy](https://docs.google.com/document/d/1denvE73pi70eCkjK9QSVW3C_mAJmqK5nuOIEWb4ksIo/edit#heading=h.usyzi29efg0n)

[TE Datagen Tool](https://docs.google.com/document/d/1SHPFGgndx4hp9NY_UlM97gkFIZqLo_196hbC-eEMZRA/edit#heading=h.mbjsiz6n6jlo)

Useful tools:

* [Regular Expressions](https://regex101.com/) [Python Regex Docs](https://docs.python.org/3/howto/regex.html)
* [Json Formatter](https://jsonlint.com/)

We are looking to introduce a REST API to Tripwire Enterprise that is easily accessible, can grow over time, and introduces a few new technologies to the TE software stack.

The primary new technology for Tripwire is Swagger:

<http://swagger.io/>

We are particularly interested in offering our integrators the rich explorative interface provided by Swagger UI, client library and documentation generation through swagger-codegen.

I expect the most suitable integration will be through this java server library:

<https://github.com/ROAMSYS/swaggerapi>

You can see a live example of swagger here:

<http://petstore.swagger.io/>

There is a DevCon presentation on Swagger here (follow the youtube video with the slideshare beside it):

<https://www.youtube.com/watch?v=U_lCrFpvXyY>

<http://www.slideshare.net/VictorTrakhtenberg/swagger2?next_slideshow=1>

The build system for our projects is designed on Gradle. The te-rest-api project is a starting point. The te-dataapi project is a source of reference, except that it is written in Clojure. A language rejected by too many in Portland in favor of Java and Groovy. So we will be looking to leverage Groovy for the bulk of the effort in the te-rest-api project.

We also utilize Google Guice for Inversion of Control / Dependency Injection in our software.