**MVC Hibernate, JSTL, Bootstrap**

**Introduction:**

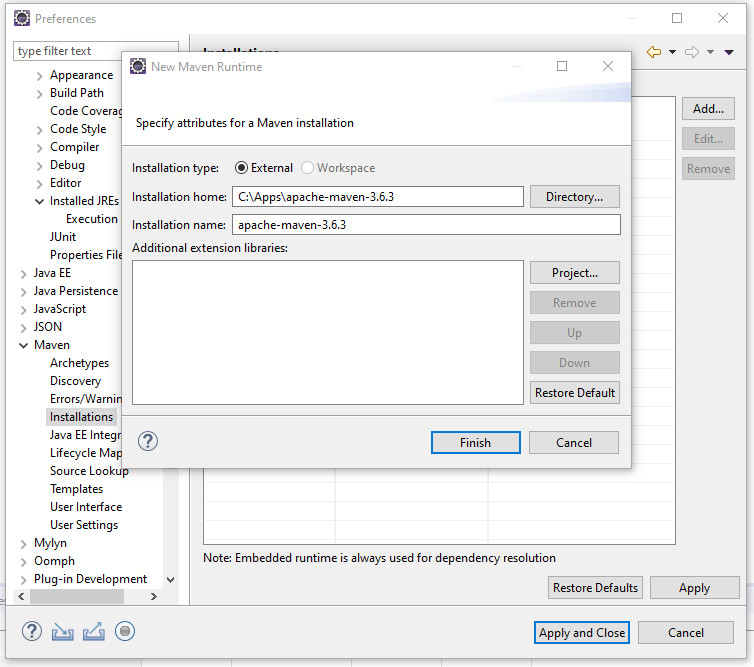
This project archetype is a template for creating a fully functional MVC web application using Hibernate, JSTL and Bootstrap. It has an automatic database creation, auto initial load of the data, with different variety of users. It also has a checkstyle to check the proper coding of your project immediately right after you enter the code.

**Requirements:**

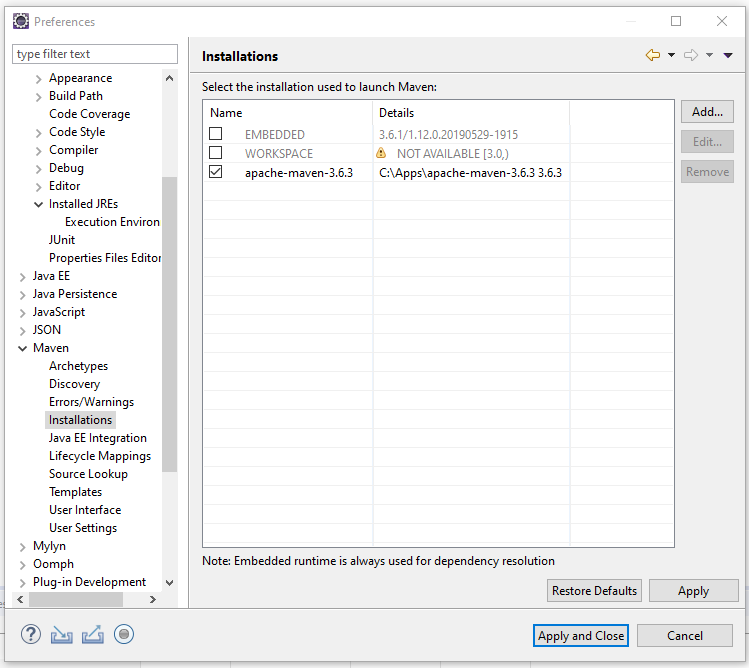
* Java v11+
* Maven v3+
* Apache Tomcat v9+
* Your database server of choice (Any DB server. i.e. PostgreSQL, SQLServer, Oracle, DB2, etc.)
* IDE of choice (Eclipse or IntelliJ)

**Setup and Configuration:**

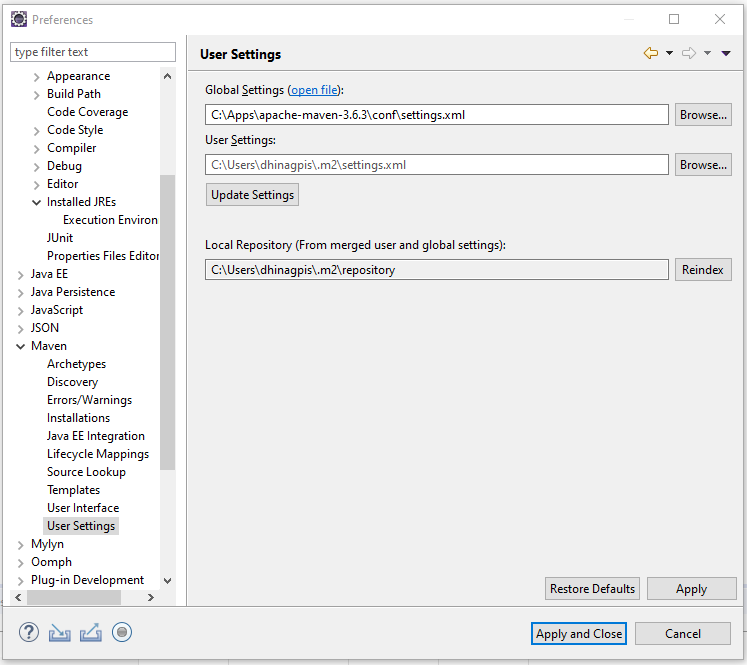
1. Download Java JDK.
2. Download Maven.
3. Download and configure database.
4. Download Apache Tomcat
5. Download and configure IDE
6. Download the “MVC Hibernate JSTL Bootstrap” maven archetype
7. Configure you default maven Installation and maven settings. Go to Windows 🡪 Preferences 🡪 Maven 🡪 Installations



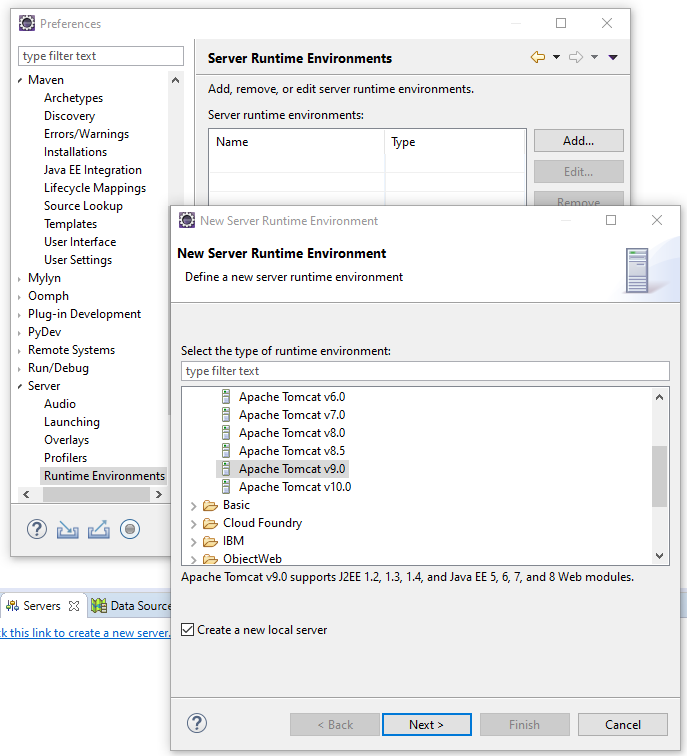
1. Set the maven as default maven installation.



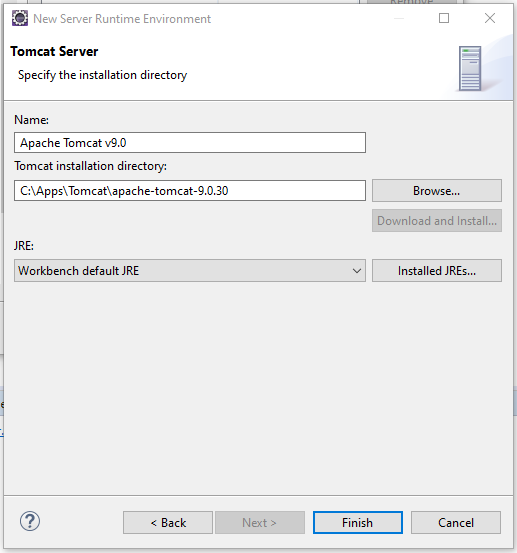
1. Select the settings.xml from the maven installation folder as the default global settings



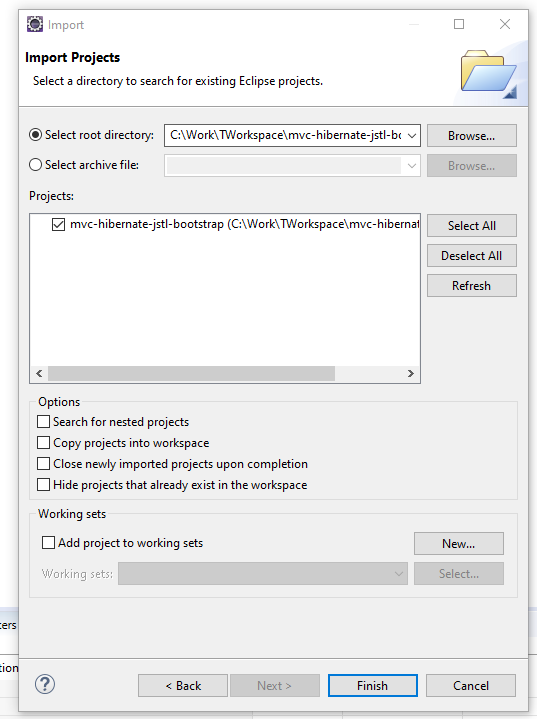
1. Configure Apache Tomcat Server, Under Windows 🡪 Preferences 🡪 Server 🡪 Runtime Environments, click Add button.
2. Select Apache Tomcat v9.0 and check the “Create a new local server”.



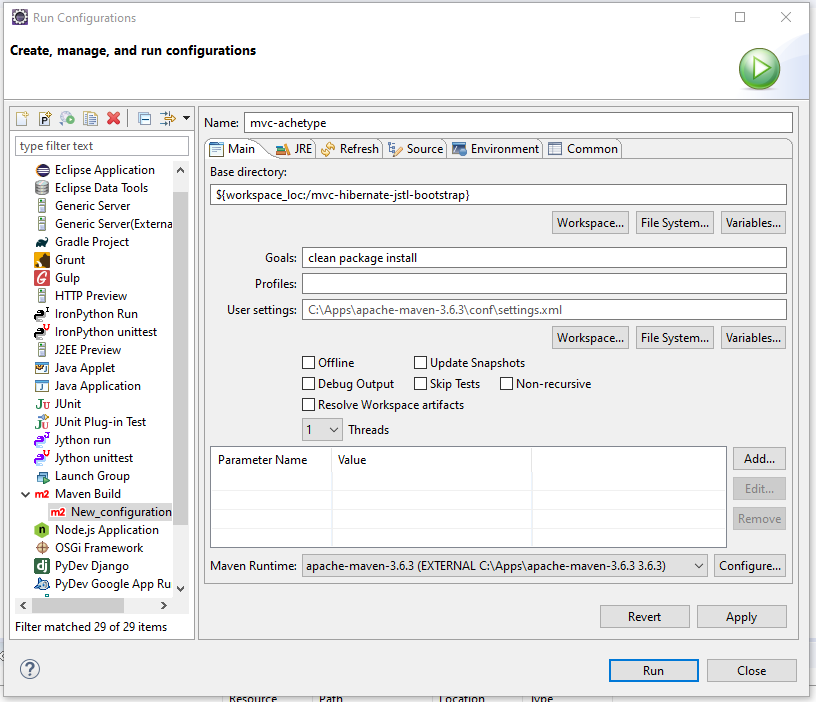
1. Select your tomcat installation folder and hit Finish button.



1. Open your IDE and import “MVC Hibernate JSTL Bootstrap” archetype maven project



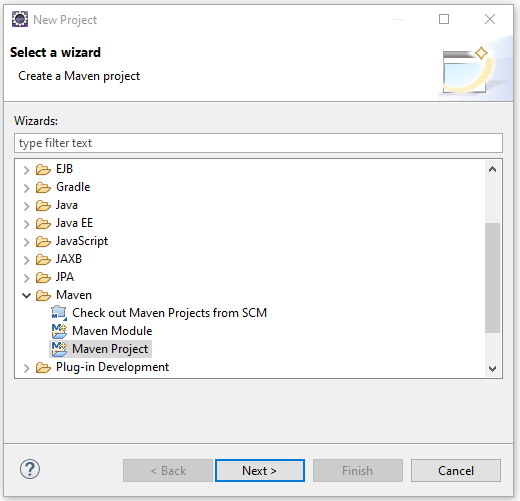
1. Create new run configuration. Under Run 🡪 Run Configurations, Create new Maven Build.
2. Set the name of the configuration, select the mvc-hibernate-jstl-bootstrap project and add the goal “clean package install”, then hit Apply and Run.



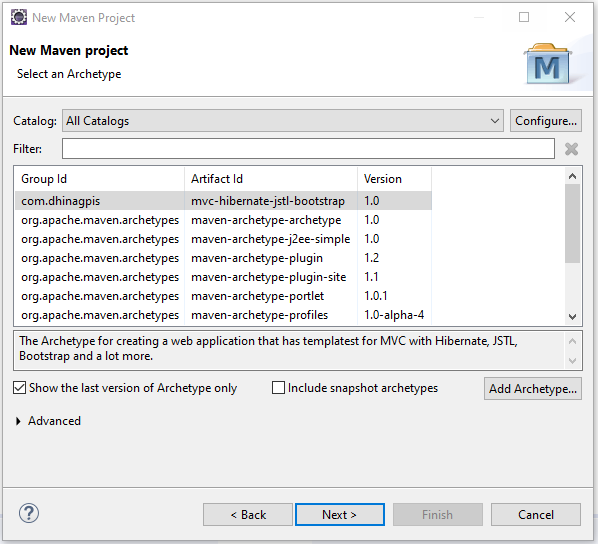
1. Success! You can now create you template project from the new mvc-hibernate-jstl-bootstrap maven archetype.

**Creating a web project:**

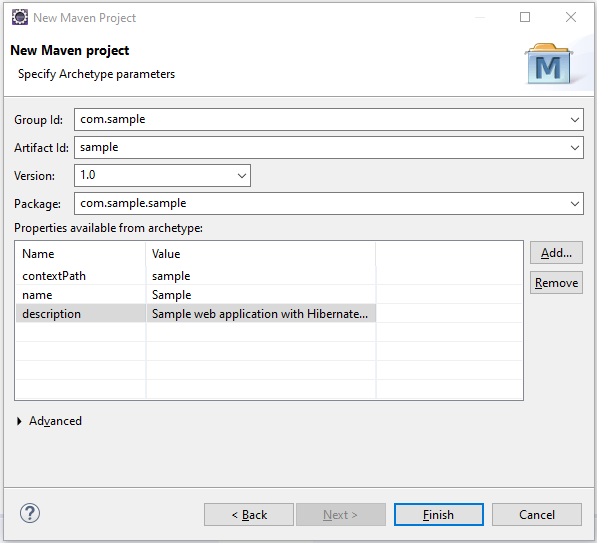
1. Create new maven project.



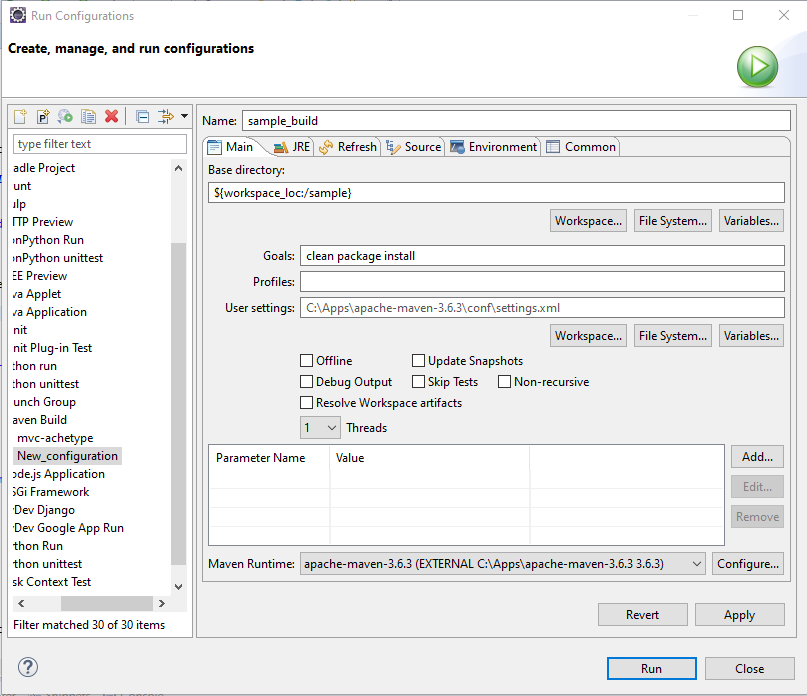
1. Hit Next twice.
2. Select the “mvc-hibernate-jstl-bootstrap” archetype.



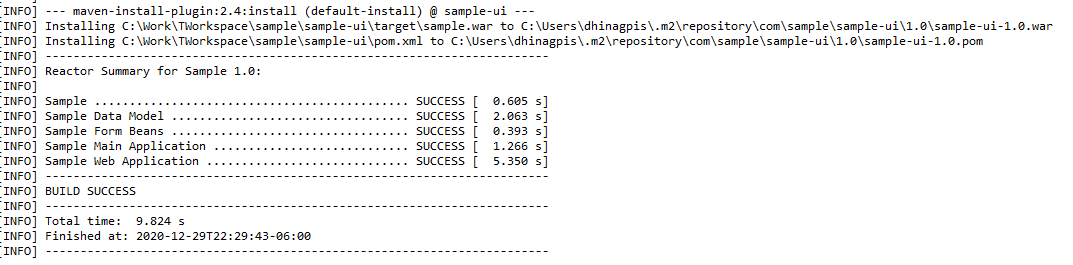
1. Set the following required fields:
   1. Group ID – Company name (i.e. com.sample)
   2. Artifact ID – Your project name (all small letter) (i.e. sample)
   3. Version – The initial version of the project (i.e. 1.0)
   4. contextPath – The default context path of your web project (i.e. sample)
   5. name – Name of the project to be display in the UI (i.e. Sample)
   6. description A short description of this project (i.e. Sample web application with Hibernate, JSTL and Bootstrap)



1. 5 new projects should be created:
   1. sample – The parent project containing all the four projects.
   2. sample-beans – Base data model project, contains all the database table with hibernate annotation. (**Model**)
   3. sample-formbeans – Request/Response data model for the UI, consist of data model representation for web UI request and response. (**View**)
   4. sample-app – Main business logic project. Contains all the logic between the UI and the database.
   5. sample-ui – Web project, consist of servlets and jsp pages for the web application (**Controller**)
2. Create new run configuration, under Run 🡪 Run Configuration, create new Maven Build. Enter the configuration name, Select the main project (in this case the “sample” project), then add “clean package install” as the Goals. Hit the Apply and Run button.



You should get the following logs:



1. To be able to automatically create the tables in the database, you need to create the database and the user that has read/write access to create or drop the database tables and indexes. (If you are using PostgreSQL database, follow the instruction of creating the database below.)
2. Go to Window 🡪 Show View 🡪 Servers, add the UI project in the server by right clicking into the server and click Add and Remove. Select the project then click “Add >”. Then hit Finish.
3. Under the Project Explorer 🡪 Servers 🡪 <Your new local server>, open catalina.properties.
4. At the bottom of the file, add the following:

DB\_CREATED=N

DEBUG\_ENABLE=false

SHOW\_SQL=false

ZONE=America/Winnipeg

#DATABASE\_HOST\_NAME=localhost

#DATABASE\_PORT=5432

#DATABASE\_NAME=sample

#DATABASE\_USERNAME=sampleadm

#DATABASE\_PASSWORD=sampleadm

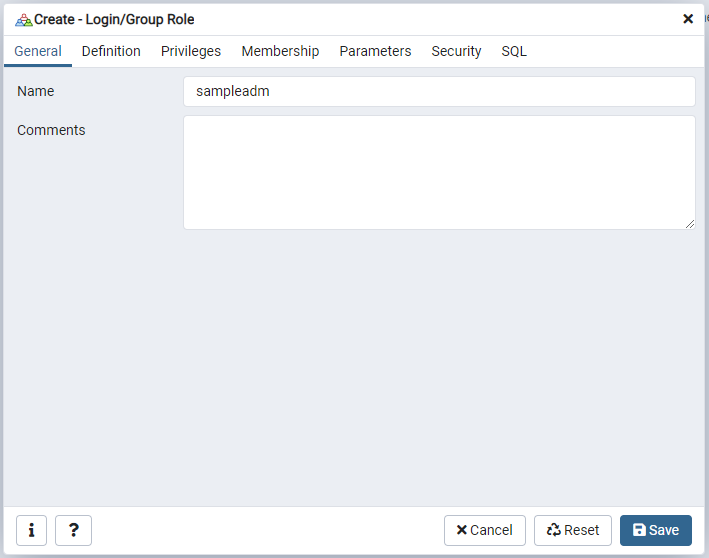
* DB\_CREATED 🡪 Y to create new database or N to update existing database
* DEBUG\_ENABLE 🡪 true or false to enable debug
* SHOW\_SQL 🡪 true or false to show sql queries in the logs
* ZONE 🡪 Your default timezone for localization
* DATABASE\_HOST\_NAME 🡪 the IP or hostname of the database server
* DATABASE\_PORT 🡪 the port of the database server
* DATABASE\_NAME 🡪 the name of the database
* DATABASE\_USERNAME 🡪 the database user
* DATABASE\_PASSWORD 🡪 the database user password

Note: Remove the “#” to uncomment the properties.

1. Right click and start the server. By default, the database tables will be automatically created and an initial set of data will be loaded into the database.
   1. su/su 🡪 the Super User
   2. root/su 🡪 the Regular User
   3. member/su 🡪 the Member of the regular user
2. You can access the website using this url: http://localhost:8080/<contextPath>/ (i.e., <http://localhost:8080/sample/>)
3. Enjoy!!!

**Creating a user and database on PostgreSQL:**

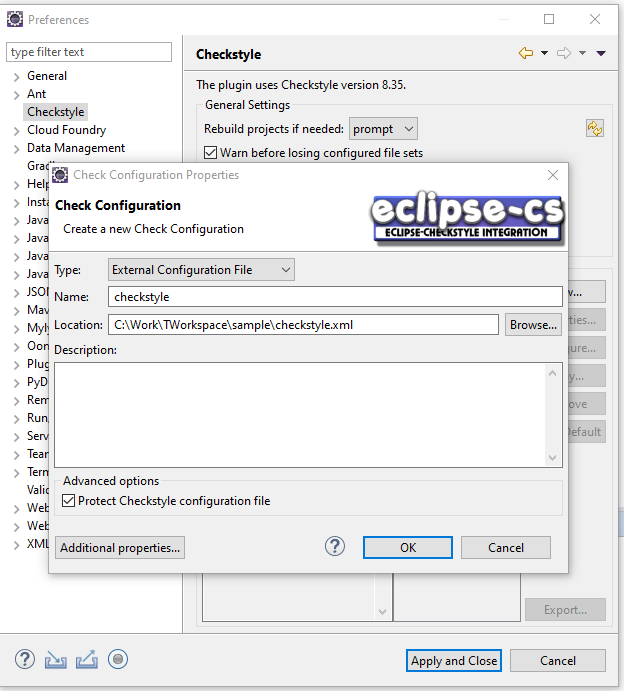
1. Open PGAdmin
2. Under PostgreSQL <Version>, Right click on Login/Group Roles 🡪 Create 🡪 Login/Group Role.
3. Enter the Name under the General tab and the password under the Definition tab. (in this case we use the same username and password for testing purposes, you can put whatever you like.)



1. Under the Priveleges tab, set the following
   1. Can login? 🡪 YES
   2. Superuser? 🡪 NO
   3. Create roles? 🡪 NO
   4. Create databases? 🡪 YES
   5. Update catalogs? 🡪 NO
   6. Inherit rights from the parent roles? 🡪 YES
   7. Can initiate streaming replication and backups? NO
2. Click the “Save” button.
3. Right click on Databases 🡪 Create 🡪 Database
4. Set the database name and select the new created user (i.e. sampleadm) as the owner. Then hist “Save” button.

**Setup Project CheckStyle:**

1. Under Windows 🡪 Preferences, Select Checkstyle. Click the New button. Select External Configuration File for the Type, enter the name of the checkstyle and navigate to your parent project default folder and look for checkstyle.xml file.



1. Select the new checkstyle as your default.
2. On your projects (\*-app, \*.beans, \*.ui), Right click and Properties. Check the “Checkstyle active for this project” and hit “Apply and Close”.

