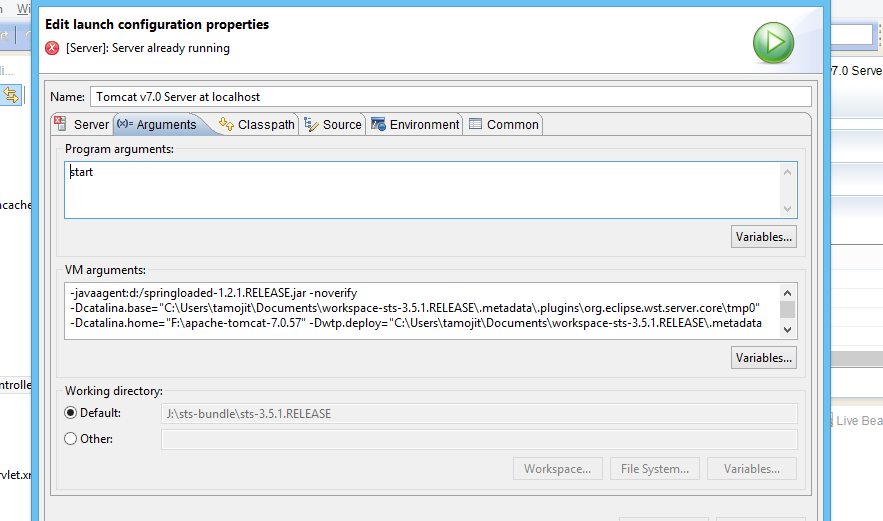
**Activate Spring loaded in tomcat:**

Step 1:

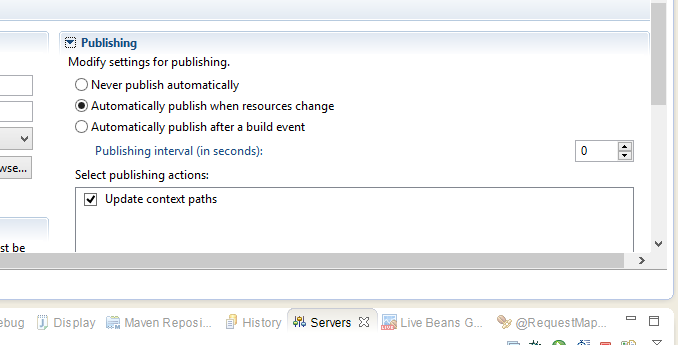
Go to tomcat launch configuration and add path to your spring loaded jar file

eg. “-javaagent:d:/springloaded-1.2.1.RELEASE.jar -noverify”



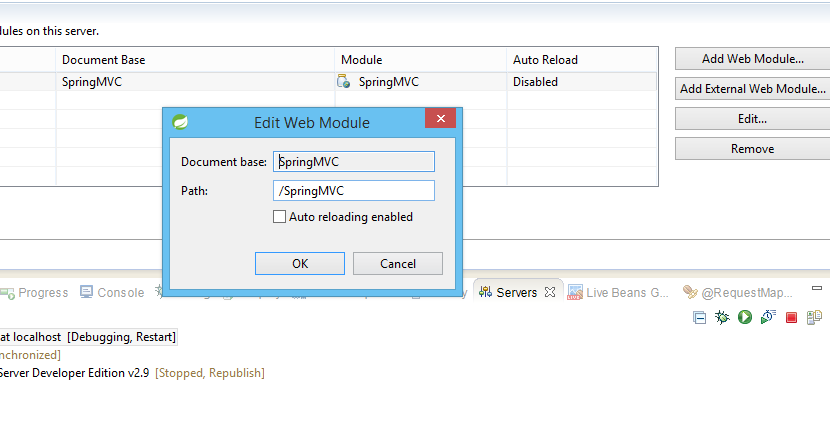
Step 2:

In tomcat configuration enable Automatically publish as shown below



Step 3:

Go to module tab and uncheck “auto reloading enable” in your project as shown below



And make sure auto build is on.

Note:

You DO need "Automatic Publishing" in Tomcat, because the changed .class files are not copied to the temporary folder where Tomcat has the application deployed to.

For example, in my local instance, the temporary Tomcat deployment folder is [WORKSPACE\_FOLDER]\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps. In this folder I have my Eclipse web app resources (classes and other files) copied over and this is where Tomcat is picking them up for deployment. While my Eclipse web app has its .class files compiled in [WORKSPACE\_FOLDER]\[MY\_WEB\_APP\_FOLDER]\target when I change one class source code, the class is recompiled and its .class file placed in target folder. With "Automatic Publishing" enabled the .class files from target folder above ARE copied over to wtpwebapps folder, whereas with that option disabled there is no copying over.

"Automatic Publishing" doesn't also mean that the application is redeployed on Tomcat, its updated .class files and other files are updated in the wtpwebapps folder, as well.

What you do need to disable in Tomcat, though, is the "Auto Reload" option for your web module. Double-click on the Tomcat Server created in Eclipse, go to "Modules" tab, click on your web app web module, then click on "Edit..." and uncheck "Auto reloading enabled". Save and restart your Tomcat.