# **Instructions for setting up the environment for java web PRJ301**

To develop and deploy a java web application on **WINDOW**, you need to follow 6 below instructions:

**Contents**

[**1> Installing JDK 8u321** 2](#_Toc102595746)

[**2> Installing Apache Tomcat 10.0.18** 3](#_Toc102595747)

[**3> Installing Apache NetBeans IDE 13.** 8](#_Toc102595748)

[**4> Integrating Apache NetBeans IDE 13 with Apache Tomcat.** 9](#_Toc102595749)

[**5> Create Java Web project, Servlet and Debug.** 11](#_Toc102595750)

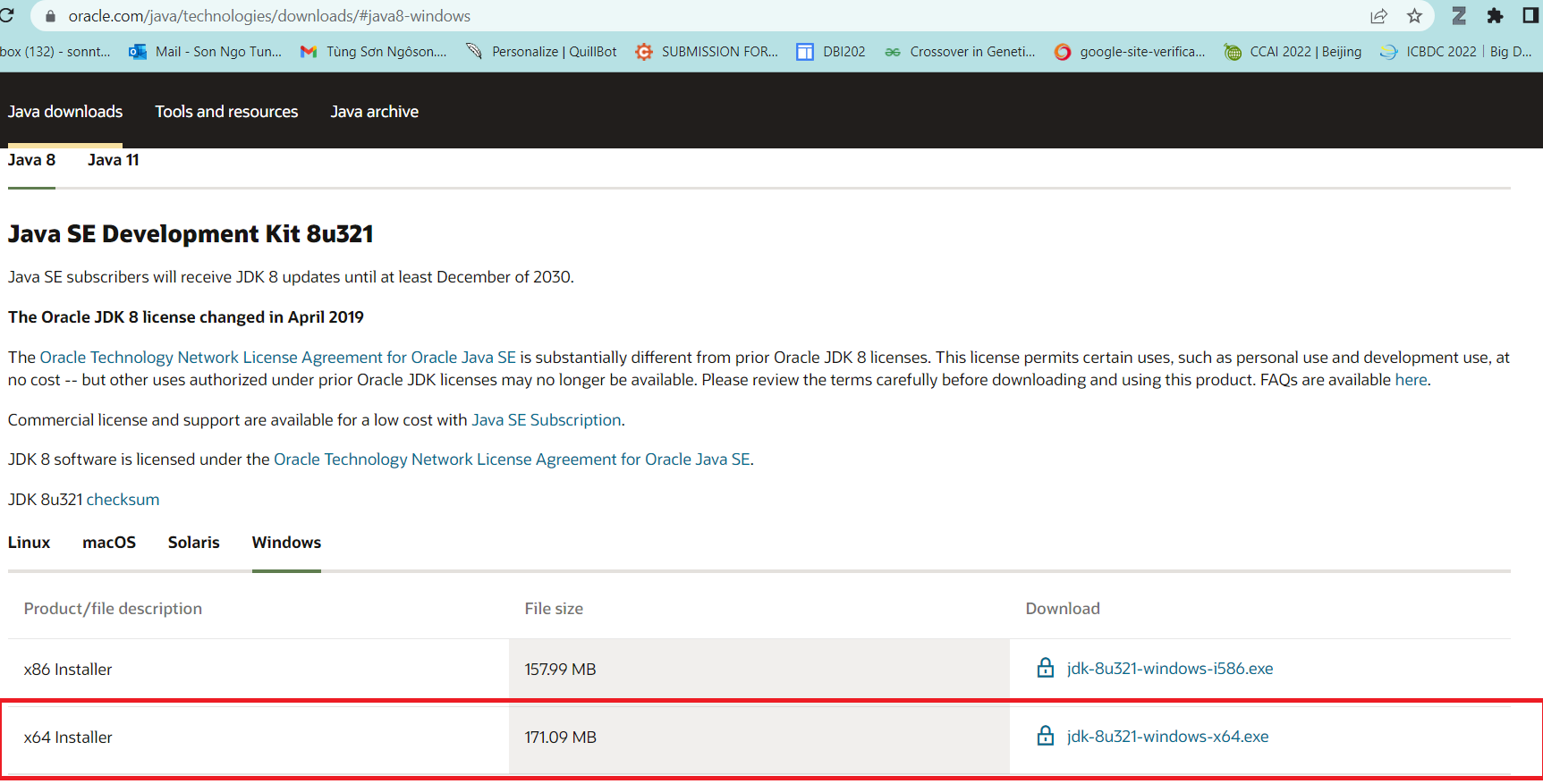
[**6> Working with JSTL.** 15](#_Toc102595751)

## 

## **1> Installing JDK 8u321**

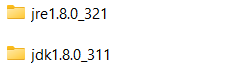
Page:

Scroll down to find the version you need.



After installing, let check the installation folder:

Normally it will be located at: C:\Program Files\Java. It contains 2 folders JDK and JRE.

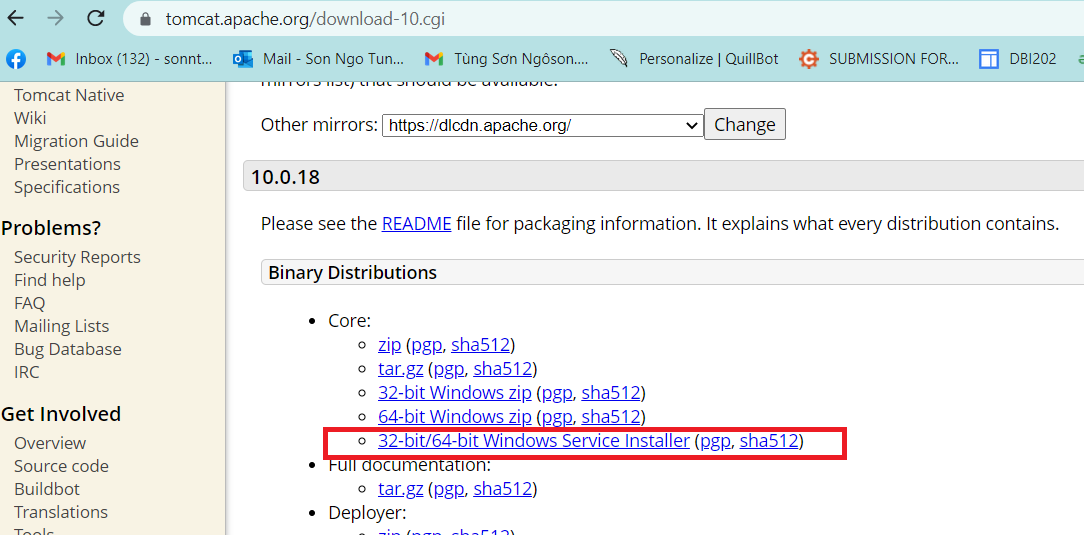


If you see these folders, it is ok for now.

For more details, you can read <https://www.oracle.com/java/technologies/javase-jdk8-doc-downloads.html>

## **2> Installing Apache Tomcat** [**10.0.18**](https://tomcat.apache.org/download-10.cgi#10.0.18)

Download page: <https://tomcat.apache.org/download-10.cgi>



Open the installer file then click “Next” button, until you see the below screen:

Graphical user interface, application

Description automatically generated

By default, the HTTP port is set as 8080, but I prefer to change it to 9999. You can keep it as 8080 or change it as you desire. Same for Window Service Name.

You can also set username and password at this step, but it is an optional and you can do it later.

Click “Next”, then the installer will ask you for JRE folder. Basically, the installer can automatically detect JRE folder, but in the case it cannot, you can add it manually.

Graphical user interface, application

Description automatically generated

Click “Next” …. Again then “Finish”.

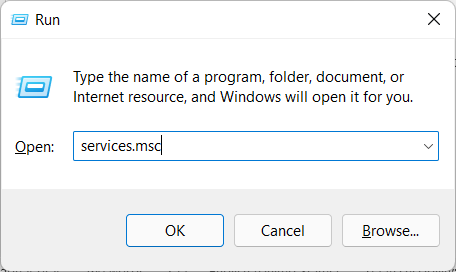
Normally, you are going to see the below window app displayed:

Graphical user interface, text, application, email

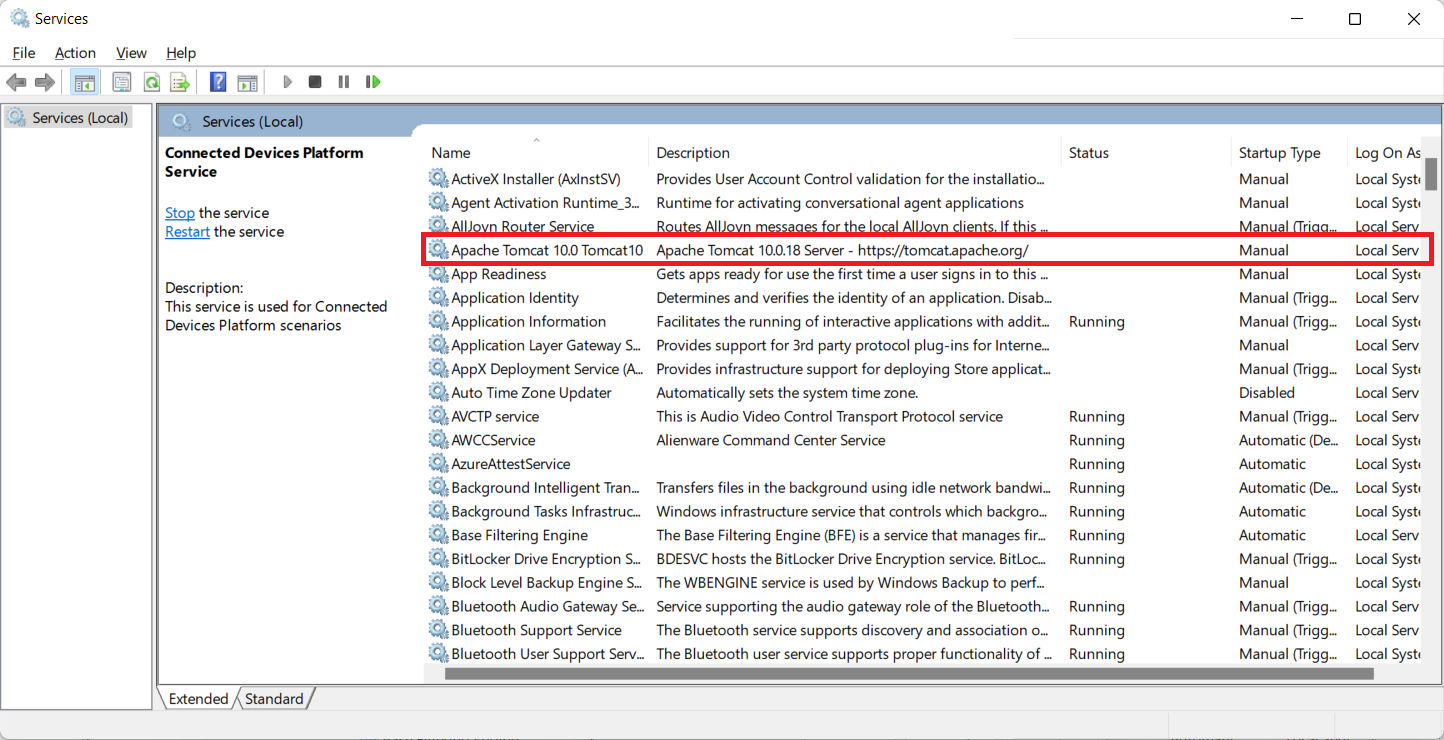
Description automatically generated

It is ok, this application used to control your TOMCAT webserver, but if you do not see it, do not worry. I recommend you follow below instructions:

1. TOMCAT now is installed as a Window Service (It is a software, but it does not have a graphic user interface).
2. To work with Window Service, press window+R, fill service.msc



1. Find the TOMCAT service in the list:



Right click -> properties ->

Graphical user interface, text, application, email

Description automatically generated

Now you can start/stop the TOMCAT service. Just make sure it is running before use.

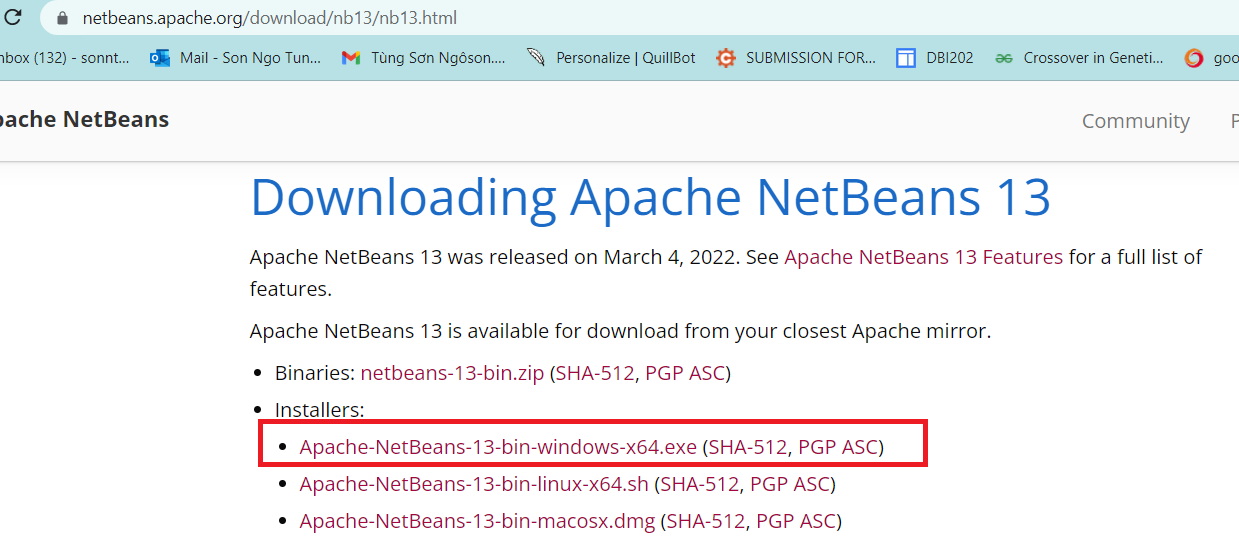
You can grant permission for TOMCAT to run with Local System account on the same dialog.

Graphical user interface, application

Description automatically generated

## **3> Installing Apache NetBeans IDE 13.**

Download: <https://netbeans.apache.org/download/nb13/nb13.html>



During your installation, remember to select JDK 1.8 (just in case, you have more than one version of Java).

After installation, run netbeans 13, File > New Project > Java with Ant > Select Web Application (It is disabled by default, just select it, netbeans will automatically enable the project template).

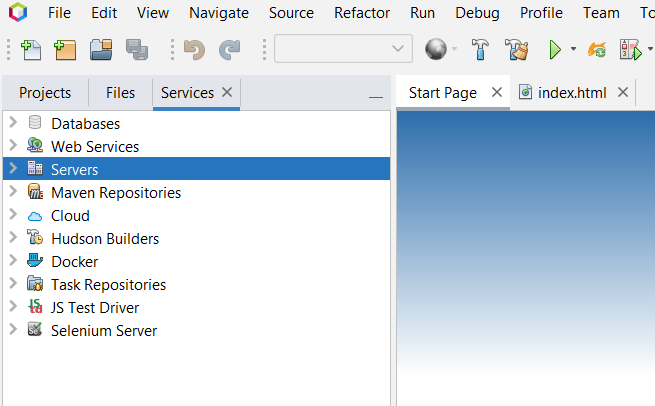
## **4> Integrating Apache NetBeans IDE 13 with Apache Tomcat.**

+ First, open the Apache Tomcat folder:

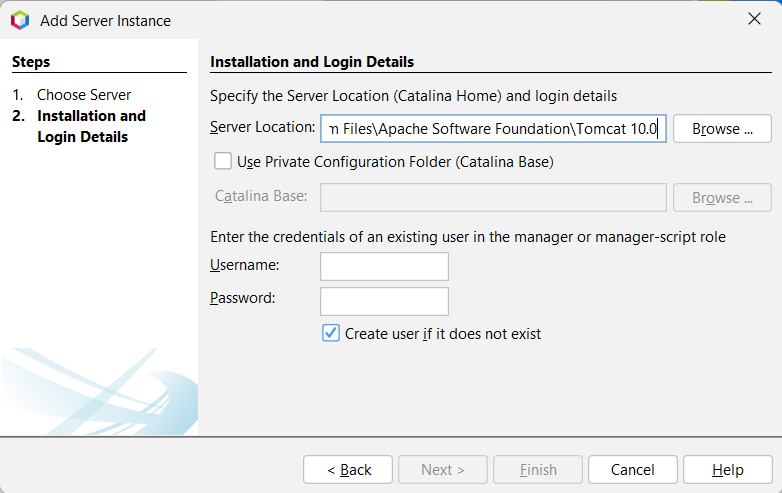
By default, it is located at: C:\Program Files\Apache Software Foundation\Tomcat 10.0

The system may ask you to grant permission, just click “Yes” button. This step aims to allows your account to access the Apache Tomcat folder for further tasks.

**+** Open netbeans, on the menu bar, Window > Service (ctrl + 5) > select Service panel as following:



Server > Right click > Add Server > Apache Tomcat and TomEE >



Here you can create your username and password, then finish. Now you can work with Java Web project in netbeans 13.

If you get an error related to “CATALINA\_base is Invalid”, check the first step.

Errors related to Tomcat will be logged to “C:\Program Files\Apache Software Foundation\Tomcat 10.0\logs”, check the catalina\*\*\*\* file for the detail logs.

## 

## **5> Create Java Web project, Servlet and Debug.**

To create a new java web project: File > New Project > Java with Ant> Java Web > Web Application (See the below figure)

Graphical user interface, text, application

Description automatically generated

/// LET DO IT, IN THE SESSION OF SERVLET

Once you create new Servlet via Netbeans template file. By default, netbeans using javax, which is not included in Jarkata EE platform equipped by Tomcat 10. Just need to replace all javax by Jakarta package, then everything will be compiled.

|  |  |
| --- | --- |
| Graphical user interface, text, application  Description automatically generated |  |

Refer to <https://tomcat.apache.org/> for more detail

Graphical user interface, text, application

Description automatically generated

you can also edit the template file for further tasks.

Graphical user interface, text, application

Description automatically generated

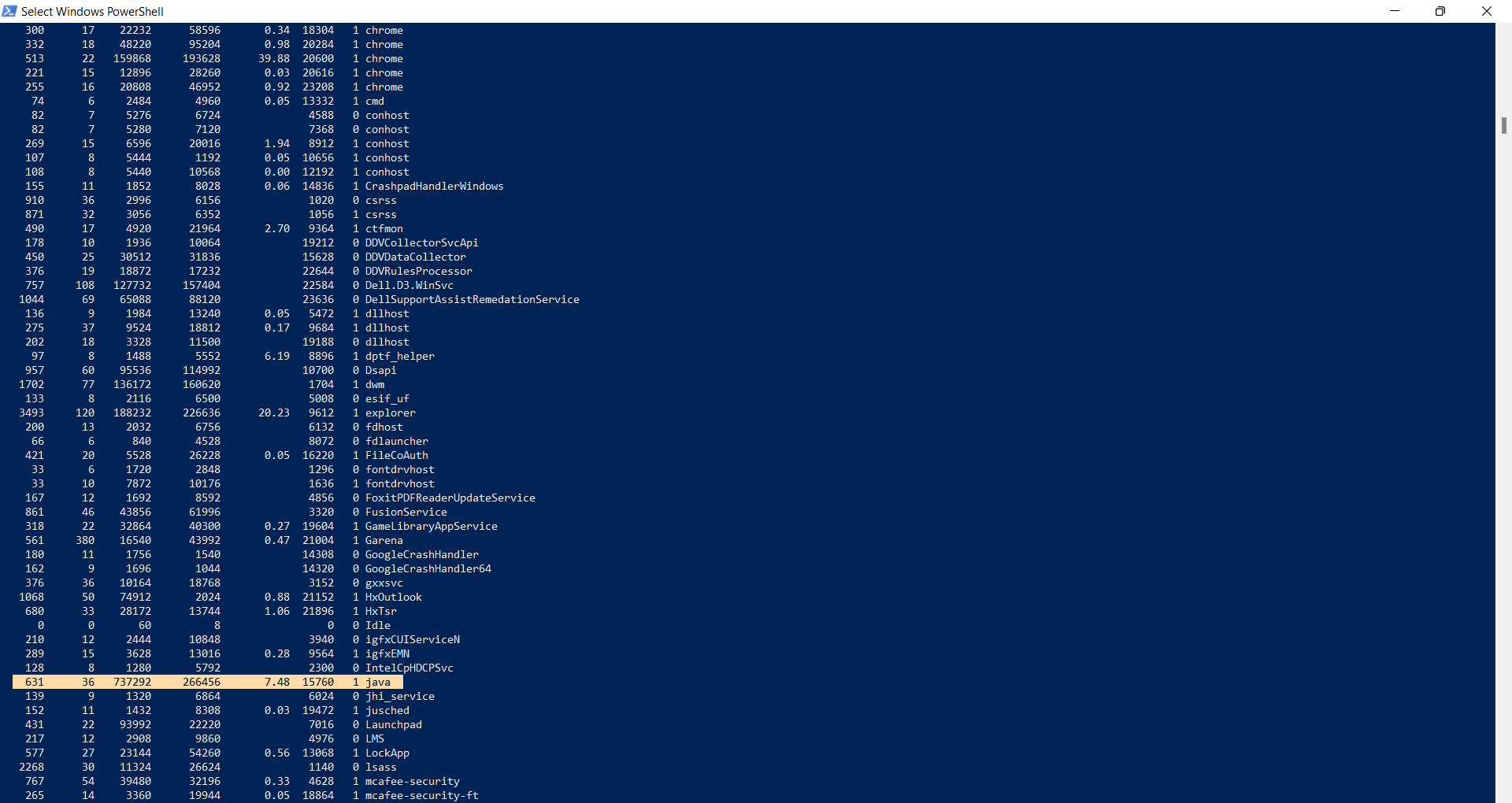
**//** For one who cannot start debugging mode with apache tomcat 10. Open task manager window then kill the process of Java SE.

Check the video: <https://youtu.be/9Zcy7yWNAm4>

OR

Open window PowerShell, type command:

get-process



Find java-se process id, then type command:

taskkill /pid 15760 /F

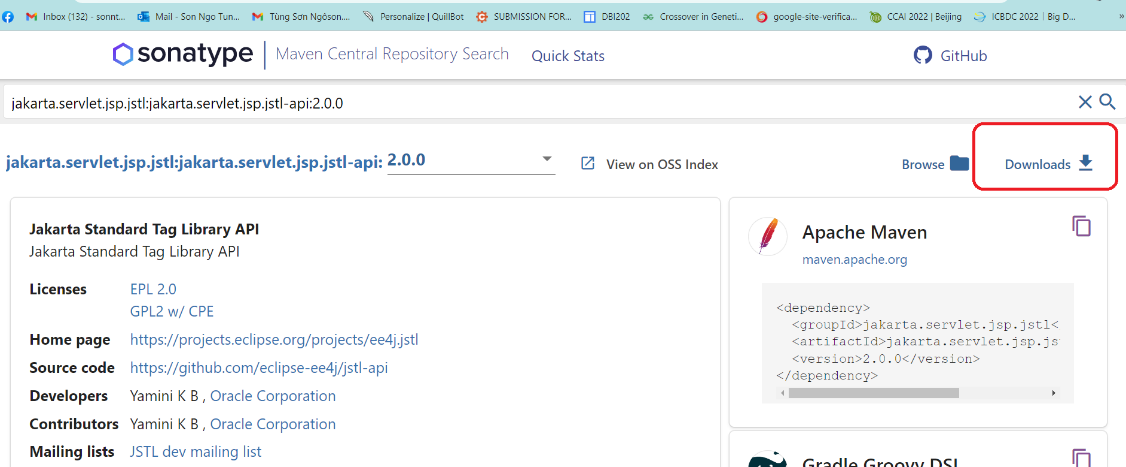
Now you can try debugging again with NetBeans and tomcat.

## **6> Working with JSTL.**

Latest version of JSTL build on J2EE was 1.2x. Since the Tomcat 10 is using Jakartified version (with jakarta.\* package instead of javax.\* package), use JSTL 2.0 to solve this dependency:

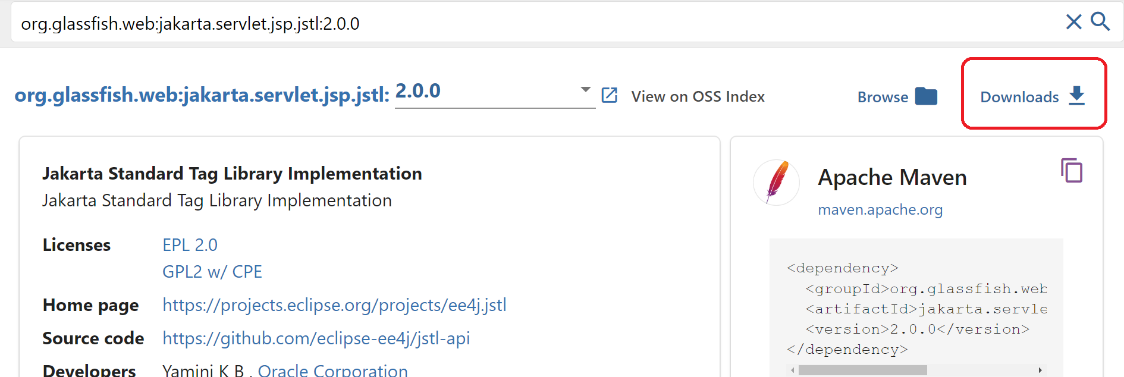
Download: Jakarta Standard Tag Library API

Link: <https://search.maven.org/artifact/jakarta.servlet.jsp.jstl/jakarta.servlet.jsp.jstl-api/2.0.0/jar>



Download: Jakarta Standard Tag Library Implementation

Link: <https://search.maven.org/artifact/org.glassfish.web/jakarta.servlet.jsp.jstl/2.0.0/jar>



Check the document here:

<https://jakarta.ee/specifications/tags/2.0/jakarta-tags-spec-2.0.html>