Workstation Prerequisites

Workstation Characteristic	Recommended Minimum
CPU	Intel x86 64-bit (or equivalent)
RAM	8 GB
Operating System	Windows 7

Our recommendation is to install all programs in your Public Users directory (usually C:\Users\Public\) and all workspaces within your own User directory (usually C:\Users\login-name\).

Java Virtual Machine Setup

1. Remove any Java Runtime Environment (JRE) installs. These will conflict with the JDK.
2. Download the Java Development Kit (JDK) for your operating system and processor architecture.
    a. Currently Java 8 is widely used, but some projects may require an additional version.
3. Run the installer with the **Public JRE option disabled**.
4. Create or update the environment variable JAVA_HOME, using the location of the JDK install as its value.
5. Add "%JAVA_HOME%\jre\bin;%JAVA_HOME%\bin" to the PATH environment variable.
6. Scrub references to any other Java JDK or JRE from PATH.
7. Log off, log back on, open a Command Prompt and type "echo %PATH%". You should see your entries from Step 5 expanded to full paths. And there should be no other references to a Java install in the Path.

Apache Tomcat Server Setup

1. Download the Apache Tomcat for your operating system and processor architecture.
    a. Currently Tomcat 8.0 is widely used, but some projects may require another version.
    b. The Windows Service installer is not needed; you will be running Tomcat from within Eclipse.
2. Extract the contents of the downloaded archive file to your workstation. It is recommended that you put it alongside your Java install so it is easy to locate.

Development Environment Setup

1. Download the Eclipse Integrated Development Environment (IDE) for your operating system and processor architecture.
    a. Currently Eclipse 4.9 (2018-09) is widely used.
    b. Extract the contents of the downloaded archive file to your workstation.
    c. Edit the **eclipse.ini** file, replacing lines '-Xms40m' and '-Xmx512m' with '-Xms1g' and '-Xmx2g'.
    d. [*If using Eclipse IDE for Java EE Developers skip to Step* 2] Within Eclipse, go to Help > Install New Software... > Work with: '2018-09 - http://download.eclipse.org/releases/2018-09'.
    e. Uncheck 'Group items by category', check 'Buildship: Eclipse Plug-ins for Gradle', 'EclEmma Java Code Coverage', 'Eclipse Faceted Project Framework', 'Eclipse Faceted Project Framework JDT Enablement', 'Eclipse Java Development Tools', 'Eclipse Java EE Developer Tools', 'Eclipse Java Web Developer Tools', 'Eclipse JSON Editors and Tools', 'Git integration for Eclipse', 'JST Server Adapters Extensions', 'JST Server UI' then click Next, Next, I accept..., Finish.
2. Install the Apache IvyDE plug-in. IvyDE is an Eclipse plug-in that integrates Apache Ivy support into Eclipse. It provides dependency (i.e., library) management using configuration files and repositories instead of manual jar file copying.
    a. Go to Help > Install New Software... > Add... For 'Name:' put 'Apache Ivy', for 'Location:' put 'http://www.apache.org/dist/ant/ivyde/updatesite' and click OK.
    b. Uncheck 'Show only the latest versions of available software' and 'Group items by category', check 'Apache Ivy (2.4.0.final)' and 'Apache IvyDE (2.2.0.final)' then click Next, Next, I accept..., Finish.
    c. After restarting Eclipse, go to Window > Preferences > Ivy > Classpath Container and check 'Resolve dependencies in workspace'. This will allow IvyDE to make links between dependent projects that you have open in your workspace.
3. Install the Checkstyle plug-in.
    a. Go to Help > Install New Software... > Add... For 'Name:' put 'Checkstyle', for 'Location:' put 'https://checkstyle.org/eclipse-cs/update' and click OK.
    b. Uncheck 'Group items by category', check 'Eclipse Checkstyle Plug-in' then click Next, Next, I accept..., Finish.
4. Install the FindBugs plug-in.
    a. Go to Help > Install New Software... > Add... For 'Name:' put 'FindBugs', for 'Location:' put 'http://findbugs.cs.umd.edu/eclipse' and click OK.
    b. Uncheck 'Group items by category', check 'FindBugs Feature' then click Next, Next, I accept..., Finish.
5. Install the Groovy plug-in.
    a. Go to Help > Install New Software... > Add... For 'Name' put 'Groovy', for 'Location:'' put 'http://dist.springsource.org/snapshot/GRECLIPSE/e4.9' and click OK.
    b. Uncheck 'Group items by category', check 'Eclipse Groovy Development Tools' and 'Groovy Compiler 2.5' then click Next, Next, I accept..., Finish.
    c. Go to Window > Preferences > Groovy
        i. > DSLD: Check 'Disable DSLD support in your workspace'.

        ii. > Editor > Formatter: Set 'Default indentation for wrapped lines' to 1.
        iii. > Compiler: Uncheck 'Include all jars in ~/groovy/lib on the classpath'.
        iv. > Compiler: If it does not state "You are currently using Groovy Compiler 2.5.x", click 'Switch to 2.5.x' button. Then follow instructions for restart.

6. Install the JavaCC plug-in.
    a. Go to Help > Install New Software... > Add... For 'Name:' put 'JavaCC', for 'Location:' put 'http://eclipse-javacc.sourceforge.net' and click OK.
    b. Uncheck 'Group items by category', check 'JavaCC Eclipse Plug-in' then click Next, Next, I accept..., Finish.
7. Install the MoreUnit plug-in.
    a. Go to Help > Install New Software... > Add... For 'Name:' put 'MoreUnit', for 'Location:' put 'http://moreunit.sourceforge.net/update-site ' and click OK.
    b. Uncheck 'Group items by category', check 'MoreUnit for Java' then click Next, Next, I accept..., Finish.
8. Install the SpotBugs plug-in.
    a. Go to Help > Install New Software... > Add... For 'Name:' put 'SpotBugs', for 'Location:' put 'https://spotbugs.github.io/eclipse' and click OK.
    b. Check 'SpotBugs' then click Next, Next, I accept..., Finish.

Eclipse Workspace Configuration

Each new workspace created by a developer requires some one-time setup, unless it was created by copying settings from an existing workspace. Unfortunately, these settings cannot be embedded in each project and so must be set by each developer.

1. Link your JDK install to Eclipse.
    a. Go to Window > Preferences > Java > Installed JREs. If your JDK is listed, select it, click Edit... and skip to Step D.
    b. Click Add... Select Standard VM and click the Next.
    c. Click Directory... Navigate to your JDK installation and click OK.
    d. For 'Default VM arguments' put '-Xmx1g -XX:+UseG1GC -XX:-OmitStackTraceInFastThrow -Djava.net.preferIPv4Stack=true' then click Finish.
    e. Click OK to add the new JRE definition to Eclipse, then go to Window > Preferences > Java > Installed JREs > Execution Environments.
    f. Click on JavaSE-1.8 in the Execution Environments list. Your newly installed JRE should be listed under Compatible JREs and should be bolded and annotated with [perfect match].
    g. Check the checkbox next to the JRE and click OK.
2. Link your Tomcat install to Eclipse.
    a. Go to Window > Preferences > Server > Runtime Environments and click Add...
    b. Choose the appropriate Tomcat version (e.g. Apache > Apache Tomcat v8.0) and click Next.

    c. Note that some projects may rely on the value in the Name field, so it is best to stick with the default (i.e. Apache Tomcat v8.0).
    d. Click Browse... Navigate to the directory in which you installed Tomcat, select it, click OK, and finally click Finish.
3. Link the JSON content type to the new JSON editor.
    a. Go to Window > Preferences > General > Editors > File Associations and click Add...
    b. For 'File type' put '*.json' and click OK.
    c. For 'Associated editors' select 'JSON Editor' and click Default.
    d. Go to Window > Preferences > General > Content Types > Text > JavaScript Source File. If '*.json' appears in the 'File associations', select it and click Remove.
    e. Go to Window > Preferences > JSON > JSON Files > Editor. For 'Line width' put 80, for 'Indentation size' put 2, select 'Indent using spaces' and click Apply.
4. Disable Eclipse's build before launching preference.
    a. Go to Window > Preferences > Run/Debug > Launching. Uncheck "Build (if required) before launching" and click Apply.
5. Disable Eclipse's build-time validation. The DTD, XML, HTML, JSP, ... validators cause a lot of noise in the Problems view and in text editors for very little value add.
    a. Go to Window > Preferences > Validation and uncheck all checkboxes in the Build column and click OK. *Note:* With this setup, you can always run the validators manually if you want to see their results.