

How to Install Tomcat 8 and run Java Servlet Programming

STEP 1: Download and Install Tomcat

Goto <http://tomcat.apache.org> ⇒ Downloads ⇒ Tomcat 8.0 ⇒ "8.0.{xx}" (where {xx} is the latest upgrade number) ⇒ Binary Distributions ⇒ Core ⇒ "ZIP" package (e.g., "apache-tomcat-8.0.{xx}.zip", about 8 MB).

Create your project directory, say "d:\myProject" or "c:\myProject". UNZIP the downloaded file into your project directory. Tomcat will be unzipped into directory "d:\myProject\apache-tomcat-8.0.{xx}".

For ease of use, we shall shorten and rename this directory to "d:\myProject\tomcat".

Take note of Your Tomcat Installed Directory. Hereafter, I shall refer to the Tomcat installed directory as <TOMCAT_HOME>.

Tomcat's Directories

Take a quick look at the Tomcat installed directory. It contains the following sub-directories:

bin: contains the binaries; and startup script (startup.bat for Windows and startup.sh for Unixes and Mac OS X), shutdown script (shutdown.bat for Windows and shutdown.sh for Unix and Mac OS X), and other binaries and scripts.

conf: contains the system-wide configuration files, such as server.xml, web.xml, context.xml, and tomcat-users.xml.

lib: contains the Tomcat's system-wide JAR files, accessible by all webapps. You could also place external JAR file (such as MySQL JDBC Driver) here.

logs: contains Tomcat's log files. You may need to check for error messages here.

webapps: contains the webapps to be deployed. You can also place the WAR (Webapp Archive) file for deployment here.

work: Tomcat's working directory used by JSP, for JSP-to-Servlet conversion.

temp: Temporary files.

STEP 2: Create an Environment Variable JAVA_HOME

(For Windows)

You need to create an environment variable called "JAVA_HOME" and set it to your JDK installed directory.

First, find your JDK installed directory. The default is "c:\Program Files\Java\jdk1.8.0_{xx}", where {xx} is the upgrade number. Take note of your JDK installed directory.

To set the environment variable JAVA_HOME in Windows 7/8/10:

Start "Control Panel" ⇒ System ⇒ Advanced system settings ⇒ Switch to "Advanced" tab ⇒ Environment Variables ⇒ System Variables ⇒ "New" ⇒ In "Variable Name", enter "JAVA_HOME" ⇒ In "Variable Value", enter your JDK installed directory as noted.

To verify, RE-START a CMD shell (restart needed to refresh the environment) and issue:

```
SET JAVA_HOME
```

```
JAVA_HOME=c:\Program Files\Java\jdk1.8.0_{xx} <== Verify that this is YOUR JDK installed directory
```

STEP 3: Configure Tomcat Server

The Tomcat configuration files are located in the "conf" sub-directory of your Tomcat installed directory, e.g. "d:\myProject\tomcat\conf" (for Windows) or "/Applications/tomcat/conf" (for Mac OS X). There are 4 configuration XML files:

server.xml

web.xml

context.xml

tomcat-users.xml

Make a BACKUP of the configuration files before you proceed.

(a) "conf\server.xml" - Set the TCP Port Number

Use a programming text editor (e.g., NotePad++, TextPad for Windows; or gEdit, jEdit for Mac OS X) to open the configuration file "server.xml", under the "conf" sub-directory of Tomcat installed directory.

The default TCP port number configured in Tomcat is 8080, you may choose any number between 1024 and 65535, which is not used by an existing application. We shall choose 9999.

Locate the following lines (around Line 69) that define the HTTP connector, and change port="8080" to port="9999".

```
<!-- A "Connector" represents an endpoint by which requests are received
```

```
and responses are returned. Documentation at :
```

```
Java HTTP Connector: /docs/config/http.html (blocking & non-blocking)
```

```
Java AJP Connector: /docs/config/ajp.html
```

```
APR (HTTP/AJP) Connector: /docs/apr.html
```

```
Define a non-SSL HTTP/1.1 Connector on port 8080
```

-->

```
<Connector port="9999" protocol="HTTP/1.1"  
    connectionTimeout="20000"  
    redirectPort="8443" />
```

(b) "conf\web.xml" - Enabling Directory Listing

Again, use a programming text editor to open the configuration file "web.xml", under the "conf" sub-directory of Tomcat installed directory.

We shall enable directory listing by changing "listings" from "false" to "true" for the "default" servlet. This is handy for test system, but not for production system for security reasons.

Locate the following lines (around Line 103) that define the "default" servlet; and change the "listings" from "false" to "true".

```
<!-- The default servlet for all web applications, that serves static -->  
<!-- resources. It processes all requests that are not mapped to other -->  
<!-- servlets with servlet mappings. -->  
<servlet>  
    <servlet-name>default</servlet-name>  
    <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>  
    <init-param>  
        <param-name>debug</param-name>  
        <param-value>0</param-value>  
    </init-param>  
    <init-param>  
        <param-name>listings</param-name>  
        <param-value>>true</param-value>  
    </init-param>  
    <load-on-startup>1</load-on-startup>  
</servlet>
```

(c) "conf\context.xml" - Enabling Automatic Reload

We shall add the attribute `reloadable="true"` to the `<Context>` element to enable automatic reload after code changes. Again, this is handy for test system but not for production, due to the overhead of detecting changes.

Locate the `<Context>` start element (around Line 19), and change it to `<Context reloadable="true">`.

```
<Context reloadable="true">
```

```
.....
```

```
.....
```

```
</Context>
```

Step 3(d) (Optional) "conf\tomcat-users.xml"

Enable the Tomcat's manager by adding the highlighted lines, inside the `<tomcat-users>` elements:

```
<tomcat-users>
```

```
<role rolename="manager-gui"/>
```

```
<user username="manager" password="xxxx" roles="manager-gui"/>
```

```
</tomcat-users>
```

This enables the manager GUI app for managing Tomcat server.

STEP 4: Start Tomcat Server

The Tomcat's executable programs and scripts are kept in the "bin" sub-directory of the Tomcat installed directory, e.g., "d:\myProject\tomcat\bin" (for Windows) or "/Applications/tomcat/bin" (for Mac OS X).

(a) Start Server

For Windows

Launch a CMD shell. Set the current directory to "`<TOMCAT_HOME>\bin`", and run "startup.bat" as follows:

```
// Change the current directory to Tomcat's "bin"
```

```
// Assume that Tomcat is installed in "d:\myProject\tomcat"
```

```
d:           // Change the current drive
```

```
cd \myProject\tomcat\bin // Change Directory to YOUR Tomcat's "bin" directory
```

```
// Start Tomcat Server
```

```
startup
```

A new Tomcat console window appears. Study the messages on the console. Look out for the Tomcat's port number (double check that Tomcat is running on port 9999). Future error messages will be sent to this console. System.out.println() issued by your Java servlets will also be sent to this console.

```
.....
```

```
.....
```

```
xxx xx, xxxx x:xx:xx xx org.apache.coyote.AbstractProtocol start
```

```
INFO: Starting ProtocolHandler ["http-bio-9999"]
```

```
xxx xx, xxxx x:xx:xx xx org.apache.coyote.AbstractProtocol start
```

```
INFO: Starting ProtocolHandler ["ajp-bio-8009"]
```

```
xxx xx, xxxx x:xx:xx xx org.apache.catalina.startup.Catalina start
```

```
INFO: Server startup in 2477 ms
```

(b) Start a Client to Access the Server

Start a browser (as HTTP client). Issue URL "http://localhost:9999" to access the Tomcat server's welcome page. The hostname "localhost" (with IP address of 127.0.0.1) is meant for local loop-back testing inside the same machine. For users on the other machines over the net, they have to use the server's IP address or DNS domain name or hostname in the format of "http://serverHostnameOrIPAddress:9999".

Apache Tomcat/7.0.30



If you're seeing this, you've successfully installed Tomcat.

Try issuing URL <http://localhost:9999/examples> to view the servlet and JSP examples.

Try running some of the servlet examples.

(Optional) Try issuing URL <http://localhost:9999/manager/html> to run the Tomcat Web Manager. Enter the username and password configured earlier in tomcat-users.xml.

(c) Shutdown Server

For Windows

You can shutdown the tomcat server by running "<TOMCAT_HOME>\bin\shutdown.bat" script.

Open a new "cmd" and issue:

```
// Change the current directory to Tomcat's "bin"
```

```
d: // Change the current drive
```

```
cd \myProject\tomcat\bin // Change Directory to YOUR Tomcat's "bin" directory
```

```
// Shutdown the server
```

```
shutdown
```

Step 5: Configure this tomcat server installed in your system in Eclipse

1. Open the Eclipse workspace
2. Select the option **Window** present in Menu bar -> **Preferences** -> **Server** -> **Installed Runtimes** to create a Tomcat installed runtime.
3. Click on **Add...** to open the **New Server Runtime** dialog, then select your runtime under **Apache** (Apache Tomcat v8.0 in this example):
4. Click **Next** , and fill in your **Tomcat installation directory**
5. Ensure the selected **JRE** is a full JDK and is of a version that will satisfy Apache Tomcat (this scenario was written using SUN JDK 1.4.2_06). If necessary, you can click on **Installed JREs...** to add JDKs to Eclipse.
6. Click **Finish**