

[\(/\)](#)[Start Here \(/start-here\)](#)[Courses ▾](#)[Guides ▾](#)[About ▾](#)[\(/feed\)](#)

Tomcat Configuration In Eclipse

Last modified: December 8, 2019

by [baeldung](https://www.baeldung.com/author/baeldung/) (<https://www.baeldung.com/author/baeldung/>)

[IDE](https://www.baeldung.com/category/ide/) (<https://www.baeldung.com/category/ide/>) [Jakarta EE](https://www.baeldung.com/category/java-ee/) (<https://www.baeldung.com/category/java-ee/>)

[Eclipse](https://www.baeldung.com/tag/eclipse/) (<https://www.baeldung.com/tag/eclipse/>)

I just announced the new *Learn Spring* course, focused on the fundamentals of Spring 5 and Spring Boot 2:

>> CHECK OUT THE COURSE ([/ls-course-start](#))

1. Overview

One of the key challenges of web development is being able to debug effectively when running on web servers. Since building, packaging, and deploying consumes a lot of time, this might be difficult to achieve.

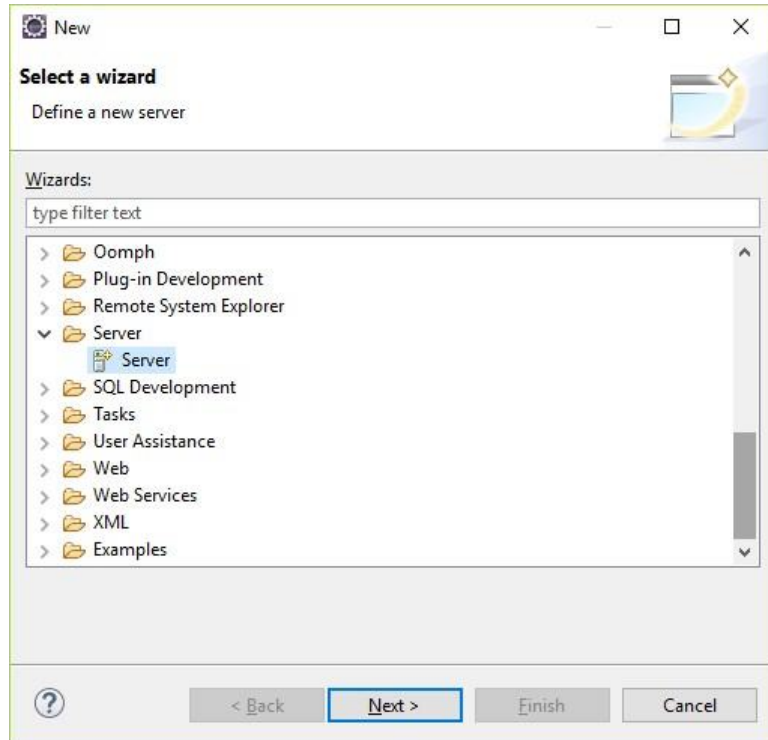
Luckily, Eclipse allows us to start a server in the IDE itself, saving build and packaging time. Also, helping us to investigate issues by starting the server in debug mode.

In this quick tutorial, **we'll see how to configure a Tomcat server in Eclipse** to achieve this.

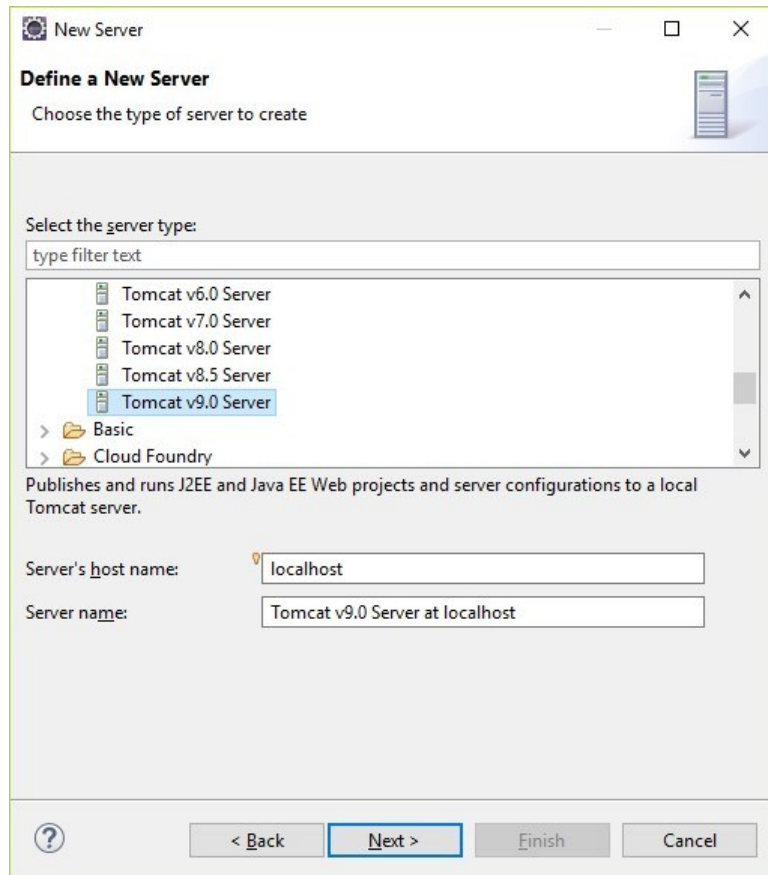
2. Defining a Server in Eclipse

Before configuring Tomcat in Eclipse, we'll have to install (<https://tomcat.apache.org>) it first.

Now, let's start by invoking the *New Server* wizard in Eclipse using *File > New > Other*:

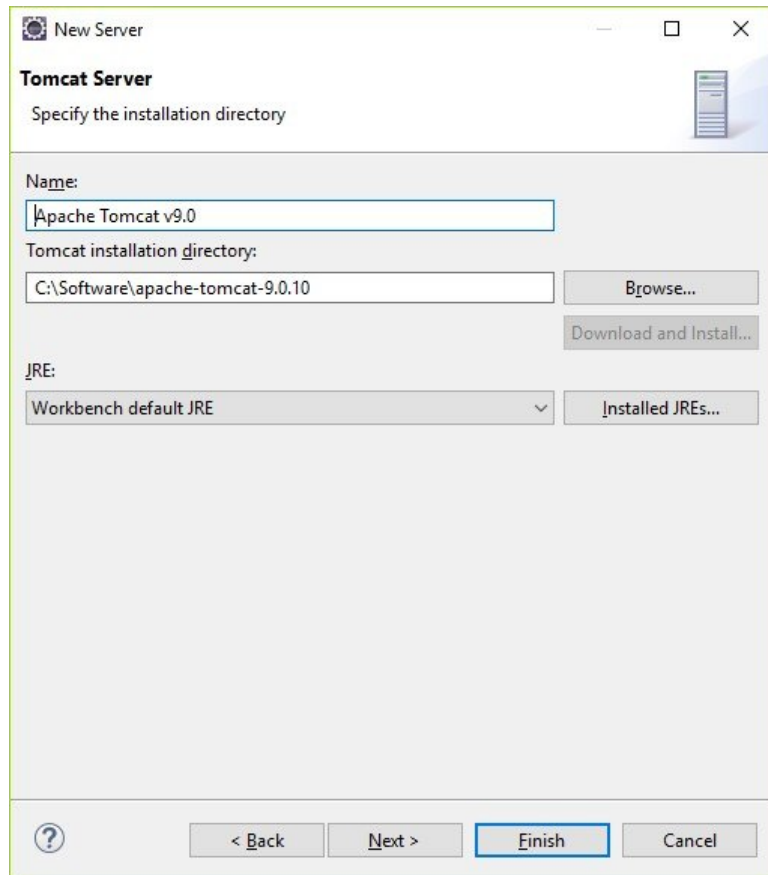


Clicking on *Next* will take us to the window where we can select the version of Tomcat. Here, we have selected *version 9.0*:



The wizard will default the name of the server to *localhost* and the server name to *Tomcat v9.0 Server at localhost*.

We'll see that the very first time we add a Tomcat server in Eclipse, the wizard will ask us to configure the server runtime environment:



Here, we'll specify the location of the Tomcat installation directory. Also, we will specify the JRE for the Tomcat server.

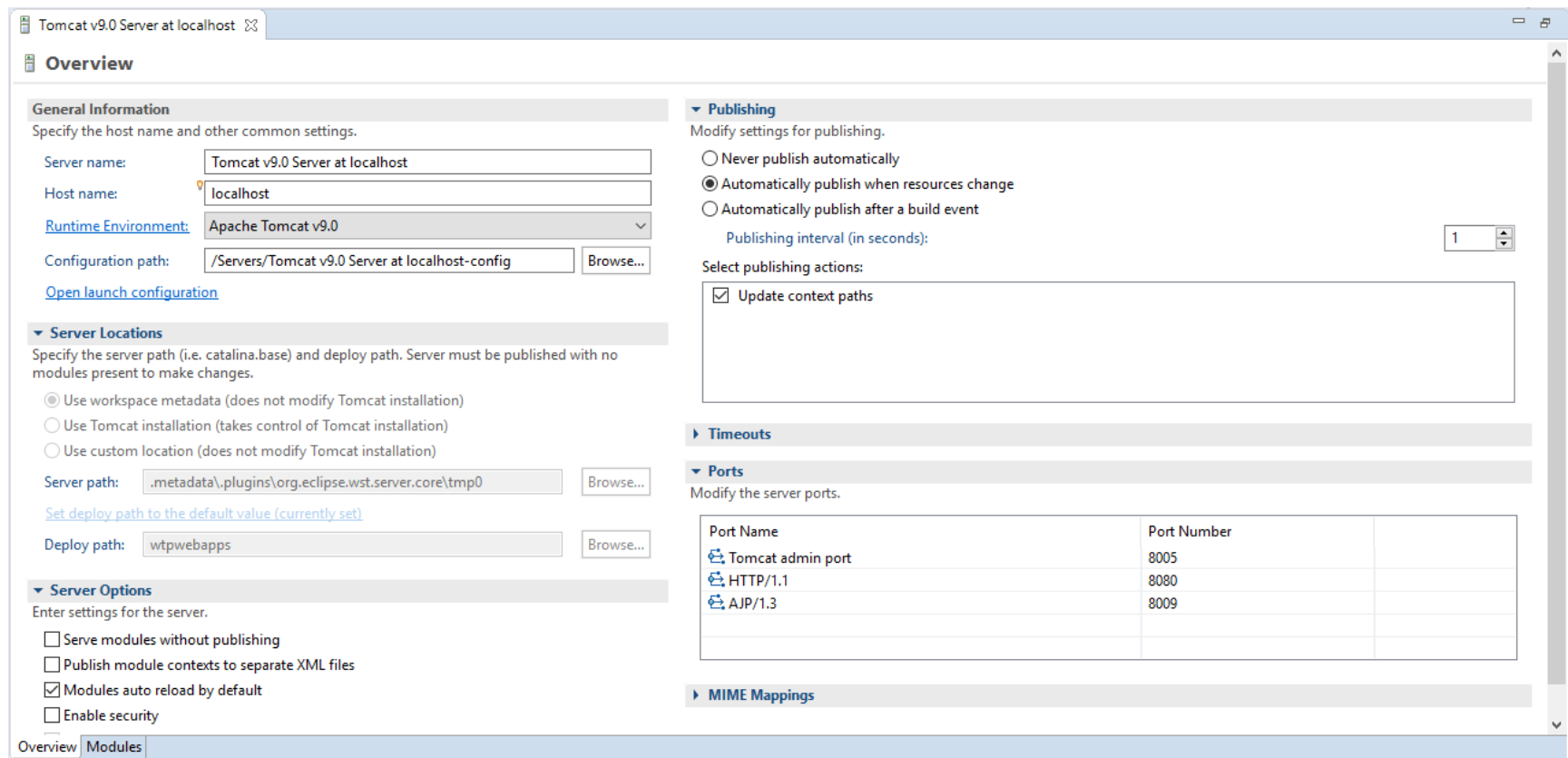
If we click *Next*, Eclipse will allow us to add the web applications to be deployed on the server. But, let's cover that in a later section and click on *Finish* instead.

Now we can see the new server in the *Project Explorer* and the *Server* views.

3. Configuring the Server

In the *Project Explorer*, we'll see the usual tomcat server configuration files, e.g. *server.xml*, *tomcat-users.xml* etc.

Also, if we double-click on *Tomcat v9.0 Server at localhost*, we can configure the server using the provided UI:



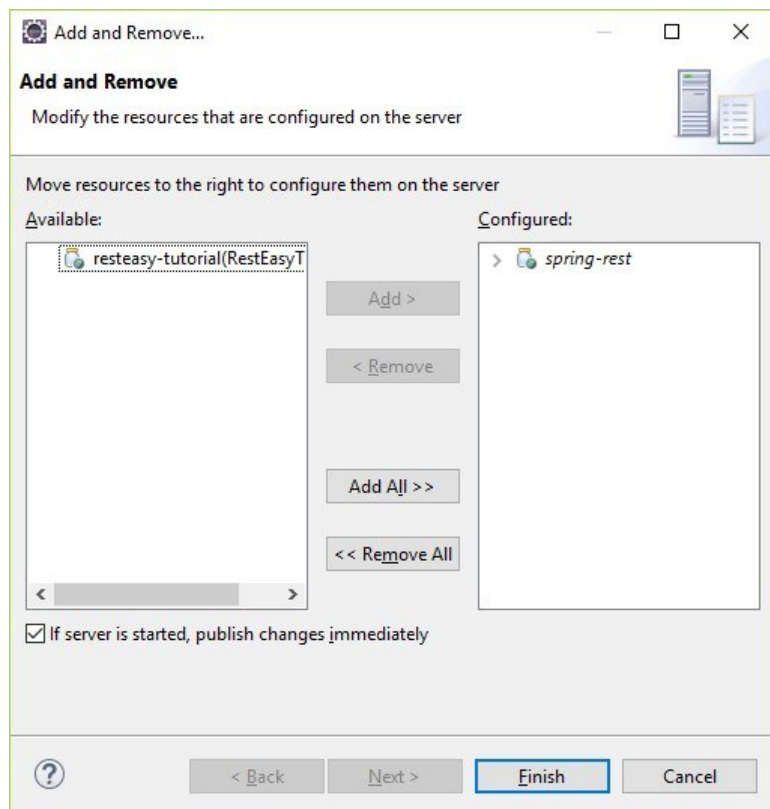
On this screen, we can configure:

- *server name* – this is the name that will appear in the server view
- *configuration path* – this is where the files we see in the *Project Explorer* reside
- *server location* – this is where we configure the location of the server installation. Also, we can set the application deploy location here
- *module publishing* – this is where we configure how the web modules are published
- *timeouts* – these are timeouts for starting/stopping the server
- *ports* – here we can set the various server ports
- *MIME mappings* – these are the various MIME type mappings
- *server launch configuration* – here we can configure the VM arguments, classpath etc.
- *server options* – here we can enable/disable features like security, auto reload of modules by default etc.

4. Adding Applications to the Server

We can now deploy our web applications on this server. Consequently, we have to make sure that the *Dynamic Web Module* facet is enabled for the project before we can add them.

So let's right-click on the tomcat server in the *Servers* view and choose the *Add and Remove...* menu item. Then, on the screen that follows, we'll select the *spring-rest* web module:



Finally, if we now click on *Finish*, we'll see *spring-rest* in the *Servers* view.

5. Running the Server

Now all that remains to do is start the tomcat server. Then we'll see the server logs in the *Console* view, while the server is starting.

Keep in mind, if the server timeout is very low, the server might fail to start. Consequently, we can resolve this by increasing the server start timeout on the configuration screen we saw above.

It is important to note that eclipse will not publish the application to the server's *webapps* folder. It will deploy this web application to a temporary folder. Therefore, leaving the Tomcat installation unmodified. If we do not change the configuration, Eclipse will publish the applications to the workspace folder:

```
1 | <workspace>/ .metadata/.plugins/org.eclipse.wst.server.core/tmp0/wtpwebapps
```

Now, Eclipse will keep monitoring our source code and look for code changes. Then, we can sync these changes with the server so that the latest code is deployed on the server.

6. Conclusion

In this tutorial, we saw how we are able to deploy our web applications in the Eclipse IDE itself.

This helps us avoid having to explicitly build, package and deploy the applications, thus save us precious development time that can be used more effectively.

I just announced the new *Learn Spring* course, focused on the fundamentals of Spring 5 and Spring Boot 2:

>> CHECK OUT THE COURSE (/ls-course-end)

Comments are closed on this article!

CATEGORIES

SPRING ([HTTPS://WWW.BAELDUNG.COM/CATEGORY/SPRING/](https://www.baeldung.com/category/spring/))

REST ([HTTPS://WWW.BAELDUNG.COM/CATEGORY/REST/](https://www.baeldung.com/category/rest/))

JAVA ([HTTPS://WWW.BAELDUNG.COM/CATEGORY/JAVA/](https://www.baeldung.com/category/java/))

SECURITY ([HTTPS://WWW.BAELDUNG.COM/CATEGORY/SECURITY-2/](https://www.baeldung.com/category/security-2/))

[PERSISTENCE \(HTTPS://WWW.BAELDUNG.COM/CATEGORY/PERSISTENCE/\)](https://www.baeldung.com/category/persistence/)

[JACKSON \(HTTPS://WWW.BAELDUNG.COM/CATEGORY/JSON/JACKSON/\)](https://www.baeldung.com/category/json/jackson/)

[HTTP CLIENT-SIDE \(HTTPS://WWW.BAELDUNG.COM/CATEGORY/HTTP/\)](https://www.baeldung.com/category/http/)

[KOTLIN \(HTTPS://WWW.BAELDUNG.COM/CATEGORY/KOTLIN/\)](https://www.baeldung.com/category/kotlin/)

SERIES

[JAVA "BACK TO BASICS" TUTORIAL \(/JAVA-TUTORIAL\)](/java-tutorial/)

[JACKSON JSON TUTORIAL \(/JACKSON\)](/jackson/)

[HTTPCLIENT 4 TUTORIAL \(/HTTPCLIENT-GUIDE\)](/httpclient-guide/)

[REST WITH SPRING TUTORIAL \(/REST-WITH-SPRING-SERIES\)](/rest-with-spring-series/)

[SPRING PERSISTENCE TUTORIAL \(/PERSISTENCE-WITH-SPRING-SERIES\)](/persistence-with-spring-series/)

[SECURITY WITH SPRING \(/SECURITY-SPRING\)](/security-spring/)

ABOUT

[ABOUT BAELDUNG \(/ABOUT\)](/about/)

[THE COURSES \(HTTPS://COURSES.BAELDUNG.COM\)](https://courses.baeldung.com)

[JOBS \(/TAG/ACTIVE-JOB/\)](/tag/active-job/)

[THE FULL ARCHIVE \(/FULL_ARCHIVE\)](/full-archive/)

[EDITORS \(/EDITORS\)](/editors/)

[OUR PARTNERS \(/PARTNERS\)](/partners/)

[ADVERTISE ON BAELDUNG \(/ADVERTISE\)](/advertise/)

[TERMS OF SERVICE \(/TERMS-OF-SERVICE\)](/terms-of-service/)

[PRIVACY POLICY \(/PRIVACY-POLICY\)](/privacy-policy/)

[COMPANY INFO \(/BAELDUNG-COMPANY-INFO\)](/baeldung-company-info/)

[CONTACT \(/CONTACT\)](/contact/)