Online Auction

Tomcat Deployment Manual

**Slippery Rock University of Pennsylvania**

Contributions by:

**Fall 2023**

Timothy Holtz – tth1003@sru.edu

Douglas Maxwell – dsm1015@sru.edu

Luke McElligott – lpm1006@sru.edu

# Table of Contents

[Table of Contents 2](#_Toc150977431)

[Tomcat Server Deployment 3](#_Toc150977432)

[Robotics Lab Specifications 3](#_Toc150977433)

[Spring project Specifications 3](#_Toc150977434)

[1. Java Compatibility: 3](#_Toc150977435)

[2. Tomcat Version: 3](#_Toc150977436)

[3. Spring Boot Version 3](#_Toc150977437)

[4. Application Packaging: 3](#_Toc150977438)

[5. Build WAR file 4](#_Toc150977439)

[6. Deploy the WAR File to Tomcat 4](#_Toc150977440)

[7. Access Your Deployed Application 5](#_Toc150977441)

[8. Monitor and Troubleshoot 5](#_Toc150977442)

# 

# Tomcat Server Deployment

This document outlines the requirements and steps for deploying a Java web application to a server running Apache Tomcat 10.0.x and using Java Runtime Environment (JRE) 1.8. It covers key considerations to ensure compatibility and successful deployment.

## Robotics Lab Specifications

Tomcat Version: 10.0

Java: 1.8.0\_352

## Spring project Specifications

Tomcat Version: 10.0

Java: 1.8

Spring Boot: 3.1.x

## 1. Java Compatibility:

Java Version: Ensure that your application is compiled and compatible with Java 1.8 (also known as Java 8). Tomcat 10.0.x is a modern servlet container that supports Java versions 8 and higher, but for this specific configuration, Java 1.8 is required.

## 2. Tomcat Version:

Tomcat 10.0.x: Make sure you are using Apache Tomcat 10.0.x, as earlier versions of Tomcat may not fully support Java 8.

## 3. Spring Boot Version

Since the server is running Tomcat 10, we need to support jakarta file structures as opposed to javax (see <https://tomcat.apache.org/migration-10.html>).

## 4. Application Packaging:

**WAR File:** Package your Java web application as a Web Application Archive (WAR) file. Tomcat deploys web applications from WAR files.

**Maven Configuration:** Using Maven, ensure that your pom.xml file specifies Java 1.8 as the source and target versions, like this:

<properties>

<java.version>1.8</java.version>

</properties>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

</plugins>

</build>

## 5. Build WAR file

Open a terminal or command prompt. Navigate to the root directory of your project where the pom.xml file is located.

Execute the following command:

mvn clean package

This command will compile your project, run tests (if any), and package your application into a WAR file. The WAR file will be created in the target directory of your project.

## 6. Deploy the WAR File to Tomcat

Once you have generated the WAR file, you can deploy it to your Tomcat server:

Locate the generated WAR file in the target directory of your project. The WAR file will have a name like your-application-name.war.

Copy the WAR file to the Tomcat webapps directory. By default, this directory is located in the Tomcat installation folder.

Start or restart Tomcat to deploy the application. You can use the provided startup scripts or commands depending on your operating system:

* On Windows, you can use **startup.bat** in the **TOMCAT\_HOME/bin** directory.
* On Unix-based systems (Linux, macOS), you can use **startup.sh** in the same directory.

## 7. Access Your Deployed Application

Once Tomcat has started and deployed your application, you can access it in a web browser by entering the following URL:

http://localhost:8080/your-application-name

## 8. Monitor and Troubleshoot

During and after deployment, monitor the Tomcat logs located in the **TOMCAT\_HOME/logs** directory for any errors or issues.