**Spring Boot automatic restart using Spring Boot DevTools**

One of Spring Boot’s key features is creating a Java web application as a standalone, executable JAR file with embedded server like Tomcat. However, this feature comes with a drawback in development time: every changes you made to your project requires complete application restart to take effect. That really slows down your productivity as your time wasted on waiting for the application restart.

To overcome that bottleneck in Spring Boot development, the Spring team created Spring DevTools that can automatically restart the application so you don’t have to do restart manually – thus you will regain your productivity.

**1. What is Spring Boot automatic restart?**

When you make changes to Java code, Spring Boot DevTools will restart the embedded server to reflect the changes. It is full, automatic restart but quick enough – much faster than “cold” restart – because the things do not change (e.g. jars) are already loaded in a separate classloader.

Note that automatic restart is different than automatic reload (or [live reload](https://www.codejava.net/frameworks/spring-boot/spring-boot-auto-reload-changes-using-livereload-and-devtools)) which reloads only changes for static content (e.g. Thymeleaf templates) in the browser. And this post focuses on Spring Boot automatic restart.

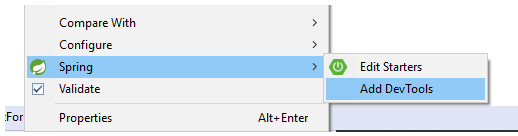
**2. Enable Automatic Restart for Spring Boot project**

To use Spring Boot DevTools, you must declare the following dependency into your project’s pom.xml file:

|  |  |
| --- | --- |
| 1  2  3  4  5 | <dependency>      <groupId>org.springframework.boot</groupId>      <artifactId>spring-boot-devtools</artifactId>      <optional>true</optional>  </dependency> |

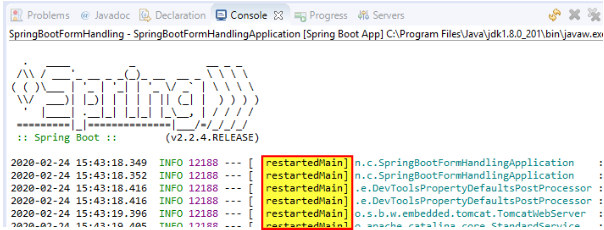
This dependency is specified as optional so the Spring DevTools will not be used in projects/modules that depend on yours. If you have only one project, you don’t have to use the optional attribute.

If you use [Spring Tool Suite IDE](https://www.youtube.com/watch?v=px5rfwi4yYk), enable Spring Boot DevTools is easy: right-click on the project, then click **Spring > Add DevTools** from the context menu:



Then Maven will download and add the spring-boot-devtools-*version*.jar to the project.

Now run your Spring boot project in the IDE, and you will see the main thread (before using DevTools) is replaced by the restartedMain thread:



Now, try to make some changes to a Java class in your project and press Ctrl + S to save the changes (Eclipse or STS), you can see in the Console view, the Spring Boot application restarts automatically but much faster than the first run – because the DevTools caches the whole Spring Boot’s jars and updates only changes in your project.

**3. What changes will trigger automatic restart?**

By default, any changes (create, update or delete) you made to any files in the project’s classpath (typically files under src/main directory) will trigger Spring DevTools to restart the application to update the changes. Actually, Spring Boot DevTools monitors changes for all entries under the project’s classpath.

If you want to include additional paths (files that are not on the classpath) to be monitored for changes, specify the following property in the application.properties file:

**spring.devtools.restart.additional-paths**=g:/Java/WebServices,data

The example above, this will add two directories outside the project’s classpath to be monitored for changes by Spring DevTools. The first path points to an absolute path on the computer, and the second one points to the data directory under the project’s root.

Note that changes to tests code (files under src/test/java) will not trigger automatic restart because testing code should be executed on request - not auto.

**4. Exclude paths from automatic restart**

By default, changes to resources in the following paths won’t trigger an automatic restart: /META-INF/maven, /META-INF/resources, /resources, /static, /public, and /templates.

If you want to override this exclusion, specify the spring.devtools.restart.exclude property in the application.properties file. For example:

spring.devtools.restart.exclude=static/\*\*

Then changes to any files under src/main/resources/static will not trigger automatic restart.

In case you want to keep the default exclusion paths and add additional paths for exclusion, use the spring.devtools.restart.additional-exclude property instead.

**5. Remove Spring Boot DevTools**

By default, the jar files of Spring Boot DevTools are not included in your project’s package (JAR/WAR). If you no longer want to use automatic restart, just remove the spring-boot-devtools dependency in the pom.xml file. In case you use Spring Tool Suite, a quick way is right-click on the project, then click **Spring > Remove DevTools**:

