**Logging**

Spring Boot uses [Commons Logging](https://commons.apache.org/logging) for all internal logging but leaves the underlying log implementation open. Default configurations are provided for [Java Util Logging](https://docs.oracle.com/en/java/javase/17/docs/api/java.logging/java/util/logging/package-summary.html), [Log4j2](https://logging.apache.org/log4j/2.x/), and [Logback](https://logback.qos.ch/" \t "_blank). In each case, loggers are pre-configured to use console output with optional file output also available.

By default, if you use the starters, Logback is used for logging. Appropriate Logback routing is also included to ensure that dependent libraries that use Java Util Logging, Commons Logging, Log4J, or SLF4J all work correctly.

|  |  |
| --- | --- |
|  | There are a lot of logging frameworks available for Java. Do not worry if the above list seems confusing. Generally, you do not need to change your logging dependencies and the Spring Boot defaults work just fine. |
|  | When you deploy your application to a servlet container or application server, logging performed with the Java Util Logging API is not routed into your application’s logs. This prevents logging performed by the container or other applications that have been deployed to it from appearing in your application’s logs. |

**Log Format**

The default log output from Spring Boot resembles the following example:

2025-05-22T20:15:56.181Z INFO 135387 --- [myapp] [ main] o.s.b.d.f.logexample.MyApplication : Starting MyApplication using Java 17.0.15 with PID 135387 (/opt/apps/myapp.jar started by myuser in /opt/apps/)

2025-05-22T20:15:56.218Z INFO 135387 --- [myapp] [ main] o.s.b.d.f.logexample.MyApplication : No active profile set, falling back to 1 default profile: "default"

2025-05-22T20:16:00.978Z INFO 135387 --- [myapp] [ main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)

2025-05-22T20:16:01.006Z INFO 135387 --- [myapp] [ main] o.apache.catalina.core.StandardService : Starting service [Tomcat]

2025-05-22T20:16:01.016Z INFO 135387 --- [myapp] [ main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.41]

2025-05-22T20:16:01.233Z INFO 135387 --- [myapp] [ main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext

2025-05-22T20:16:01.263Z INFO 135387 --- [myapp] [ main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 4702 ms

2025-05-22T20:16:02.752Z INFO 135387 --- [myapp] [ main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path '/'

2025-05-22T20:16:02.828Z INFO 135387 --- [myapp] [ main] o.s.b.d.f.logexample.MyApplication : Started MyApplication in 10.053 seconds (process running for 11.546)

2025-05-22T20:16:02.878Z INFO 135387 --- [myapp] [ionShutdownHook] o.s.b.w.e.tomcat.GracefulShutdown : Commencing graceful shutdown. Waiting for active requests to complete

2025-05-22T20:16:02.917Z INFO 135387 --- [myapp] [tomcat-shutdown] o.s.b.w.e.tomcat.GracefulShutdown : Graceful shutdown complete

Copied!

The following items are output:

* Date and Time: Millisecond precision and easily sortable.
* Log Level: ERROR, WARN, INFO, DEBUG, or TRACE.
* Process ID.
* A --- separator to distinguish the start of actual log messages.
* Application name: Enclosed in square brackets (logged by default only if spring.application.name is set)
* Application group: Enclosed in square brackets (logged by default only if spring.application.group is set)
* Thread name: Enclosed in square brackets (may be truncated for console output).
* Correlation ID: If tracing is enabled (not shown in the sample above)
* Logger name: This is usually the source class name (often abbreviated).
* The log message.

|  |  |
| --- | --- |
|  | Logback does not have a FATAL level. It is mapped to ERROR. |
|  | If you have a spring.application.name property but don’t want it logged you can set logging.include-application-name to false. |

|  |  |
| --- | --- |
|  | If you have a spring.application.group property but don’t want it logged you can set logging.include-application-group to false. |