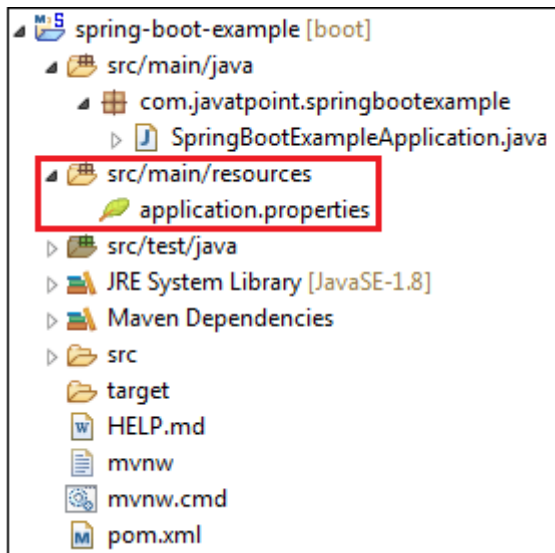


Spring Boot Application Properties

Spring Boot Framework comes with a built-in mechanism for application configuration using a file called **application.properties**. It is located inside the **src/main/resources** folder, as shown in the following figure.



Spring Boot provides various properties that can be configured in the **application.properties** file. The properties have default values. We can set a property(s) for the Spring Boot application. Spring Boot also allows us to define our own property if required.

The application.properties file allows us to run an application in a **different environment**. In short, we can use the application.properties file to:

- Configure the Spring Boot framework
- define our application custom configuration properties

Example of application.properties

1. #configuring application name
2. spring.application.name = demoApplication
3. #configuring port
4. server.port = 8081

In the above example, we have configured the **application name** and **port**. The port 8081 denotes that the application runs on port 8081.

Note: The lines started with # are comments.

YAML Properties File

Spring Boot provides another file to configure the properties is called **yml** file. The Yaml file works because the **Snake YAML** jar is present in the classpath. Instead of using the application.properties file, we can also use the application.yml file, but the **Yml** file should be present in the classpath.

Example of application.yml

1. spring:
2. application:
3. name: demoApplication
4. server:
5. port: **8081**

In the above example, we have configured the **application name** and **port**. The port 8081 denotes that the application runs on port **8081**.

Spring Boot Property Categories

There are **sixteen** categories of Spring Boot Property are as follows:

1. Core Properties
2. Cache Properties
3. Mail Properties
4. JSON Properties
5. Data Properties
6. Transaction Properties
7. Data Migration Properties
8. Integration Properties
9. Web Properties
10. Templating Properties
11. Server Properties
12. Security Properties
13. RSocket Properties
14. Actuator Properties

15. DevTools Properties

16. Testing Properties

Application Properties Table

The following tables provide a list of common Spring Boot properties:

Property	Default Values	Description
Debug	false	It enables debug logs.
spring.application.name		It is used to set the application name.
spring.application.admin.enabled	false	It is used to enable admin features of the application.
spring.config.name	application	It is used to set config file name.
spring.config.location		It is used to config the file name.
server.port	8080	Configures the HTTP server port
server.servlet.context-path		It configures the context path of the application.
logging.file.path		It configures the location of the log file.
spring.banner.charset	UTF-8	Banner file encoding.
spring.banner.location	classpath:banner.txt	It is used to set banner file location.

logging.file		It is used to set log file name. For example, data.log.
spring.application.index		It is used to set application index.
spring.application.name		It is used to set the application name.
spring.application.admin.enabled	false	It is used to enable admin features for the application.
spring.config.location		It is used to config the file locations.
spring.config.name	application	It is used to set config the file name.
spring.mail.default-encoding	UTF-8	It is used to set default MimeMessage encoding.
spring.mail.host		It is used to set SMTP server host. For example, smtp.example.com .
spring.mail.password		It is used to set login password of the SMTP server.
spring.mail.port		It is used to set SMTP server port.
spring.mail.test-connection	false	It is used to test that the mail server is available on startup.

spring.mail.username		It is used to set login user of the SMTP server.
spring.main.sources		It is used to set sources for the application.
server.address		It is used to set network address to which the server should bind to.
server.connection-timeout		It is used to set time in milliseconds that connectors will wait for another HTTP request before closing the connection.
server.context-path		It is used to set context path of the application.
server.port	8080	It is used to set HTTP port.
server.server-header		It is used for the Server response header (no header is sent if empty)
server.servlet-path	/	It is used to set path of the main dispatcher servlet
server.ssl.enabled		It is used to enable SSL support.
spring.http.multipart.enabled	True	It is used to enable support of multi-part uploads.

spring.servlet.multipart.max-file-size	1MB	It is used to set max file size.
spring.mvc.async.request-timeout		It is used to set time in milliseconds.
spring.mvc.date-format		It is used to set date format. For example, dd/MM/yyyy.
spring.mvc.locale		It is used to set locale for the application.
spring.social.facebook.app-id		It is used to set application's Facebook App ID.
spring.social.linkedin.app-id		It is used to set application's LinkedIn App ID.
spring.social.twitter.app-id		It is used to set application's Twitter App ID.
security.basic.authorize-mode	role	It is used to set security authorize mode to apply.
security.basic.enabled	true	It is used to enable basic authentication.
Spring.test.database.replace	any	Type of existing DataSource to replace.
Spring.test.mockmvc.print	default	MVC Print option
spring.freemarker.content-type	text/html	Content Type value

server.server-header		Value to use for the server response header.
spring.security.filter.dispatcher-type	async, error, request	Security filter chain dispatcher types.
spring.security.filter.order	-100	Security filter chain order.
spring.security.oauth2.client.registration.*		OAuth client registrations.
spring.security.oauth2.client.provider.*		OAuth provider details.