Property Precedence

In Spring Boot, property sources are considered in a specific order to allow for sensible overriding of property values. This order ensures that later property sources can override values defined in earlier ones. The property source order is as follows:

1. **Command Line Arguments:** Property values specified as command-line arguments when starting the Spring Boot application.
2. **Properties Files:** Property values defined in properties files located in specific locations on the classpath. These properties files can include `application.properties`, `application.yml`, and others.
3. **Environment Variables:** Property values provided through environment variables.
4. **System Properties:** Property values specified as JVM system properties.

This property source order enables flexibility in configuring Spring Boot applications. Properties can be overridden easily by specifying them in higher-priority sources, such as command-line arguments or environment variables. This allows for customization of application behavior without modifying the application code or properties files.

Yes, that's correct. In the property source order used by Spring Boot, system properties specified as JVM system properties will be overridden by any corresponding properties provided as command-line arguments when starting the Spring Boot application.

This behavior allows for easy customization and configuration of the application at runtime by providing command-line arguments to override default or previously set system properties.