User Guide

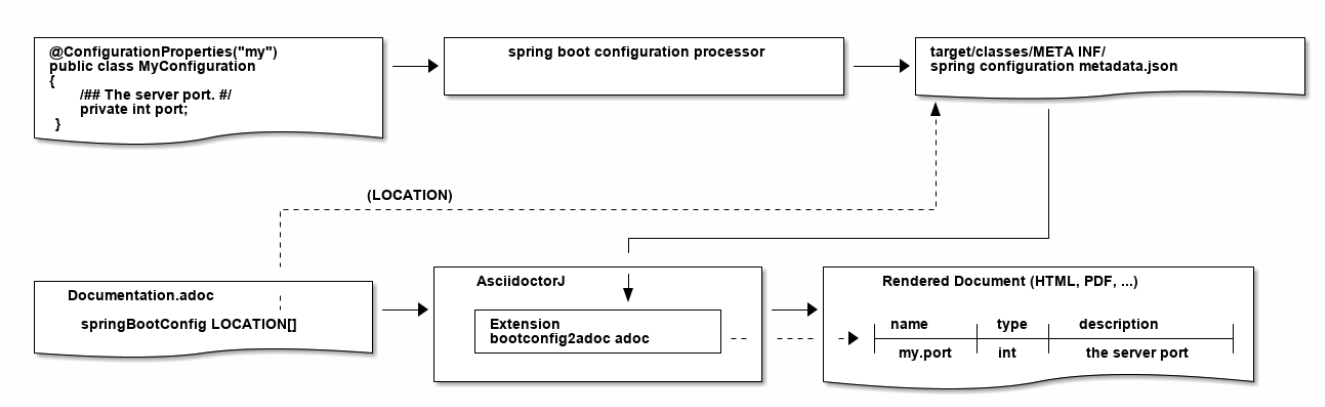
Introduction

This AsciidoctorJ extension inserts the configuration metadata description generated by spring-boot-configuration-processor into the AsciiDoc document. The inserted description is formatted for readability.

The process works as follows. There is an `@ConfigurationProperties` annotated Spring Boot configuration class (`MyConfiguration`). When compiling the source code, the `spring-boot-configuration-processor` processor creates a `spring-configuration-metadata.json` file based on the information from this class.

On the other side there is an Asciidoc document (`Documentation.adoc`). This contains the `springBootConfig::LOCATION[]` statement. Where `LOCATION` (target) indicates the location of the `spring-configuration-metadata.json` file.

AsciidoctorJ is started with extension `bootconfig2adoc-adoc`. This extension takes the `springBootConfig::LOCATION[]` statement, loads the `spring-configuration-metadata.json` file, and generates a table from its content when rendering the document by Asciidoctor.



springBootConfig

The block macro `springBootConfig` is used to insert the configuration.

```
springBootConfig::TARGET[]
```

The target (after the two colons) must be the spring-configuration metadata.json generated by the Spring Processor. It is recommended to use a Asciidoctor attribute (for example `buildTargetDir-config`) that specifies the path to the Maven Target directory in the subdirectory of which is the config-metadata file.

```
springBootConfig::{buildTargetDir-config}/classes/META-INF/spring-configuration-
metadata.json[]
```

This attribute must be set when starting Asciidoctor, for example in the Maven configuration.

```
<plugin>
    <groupId>org.asciidoctor</groupId>
    <artifactId>asciidoctor-maven-plugin</artifactId>
    ...
    <configuration>
       ...
       <attributes>
          ...
          <buildTargetDir-config>${project.build.directory}</buildTargetDir-config>
          ...
       </attributes>
   </configuration>
</plugin>
```

For the extension to be used, the dependency `de.humanfork.asciidoctorj.bootconfig2adoc:bootconfig2adoc-adoc` must be added to the Maven plugin.

```
<plugin>
    <groupId>org.asciidoctor</groupId>
    <artifactId>asciidoctor-maven-plugin</artifactId>
    ...
    <dependencies>
        ...
        <dependency>
            <groupId>de.humanfork.asciidoctorj.bootconfig2adoc</groupId>
            <artifactId>bootconfig2adoc-adoc</artifactId>
            <version>0.1.</version> <!-- or newer -->
        </dependency>
        ...
    </dependencies>
    ...
</plugin>
```

AsciidoctorJ Version

This extension needs AsciidoctorJ Version 1.6.x. At the moment the Maven Plugin `org.asciidoctor:asciidoctor-maven-plugin` (latest version 1.5.7.1) still uses AsciidoctorJ version 1.5. Therefore you have to explicitly specify the version of the AsciidoctorJ implementation used by the plugin.

```
<plugin>
    <groupId>org.asciidoctor</groupId>
    <artifactId>asciidoctor-maven-plugin</artifactId>
    ...
    <dependencies>
        <dependency>
            <groupId>org.asciidoctor</groupId>
            <artifactId>asciidoctorj</artifactId>
            <version>1.6.1</version>
        </dependency>
        ...
    </dependencies>
    ...
</plugin>
```

Example Maven Configuration

This example includes all the above instructions.

AsciiDoc `example.adoc`

```
# Configuration
The application uses the following configuration parameters:

springBootConfig::{buildTargetDir-config}/classes/META-INF/spring-configuration-
metadata.json[]
```

Maven `pom.xml´

```
<plugin>
    <groupId>org.asciidoctor</groupId>
    <artifactId>asciidoctor-maven-plugin</artifactId>

    <dependencies>
        <dependency>
            <groupId>org.asciidoctor</groupId>
            <artifactId>asciidoctorj</artifactId>
            <version>1.6.1</version>
        </dependency>
        <dependency>
            <groupId>de.humanfork.asciidoctorj.bootconfig2adoc</groupId>
            <artifactId>bootconfig2adoc-adoc</artifactId>
            <version>0.1.</version> <!-- or newer -->
        </dependency>
        ...
    </dependencies>

    <configuration>
       ...
       <attributes>
          ...
          <buildTargetDir-config>${project.build.directory}</buildTargetDir-config>
          ...
       </attributes>
   </configuration>
</plugin>
```

A real live example is module `bootconfig2adoc-it`.