52. Monitoring and Management over JMX

Java Management Extensions (JMX) provide a standard mechanism to monitor and manage applications. By default, Spring Boot exposes management endpoints as JMX MBeans under the `org.springframework.boot` domain.

52.1 Customizing MBean Names

The name of the MBean is usually generated from the `id` of the endpoint. For example, the `health` endpoint is exposed as `org.springframework.boot:type=Endpoint,name=Health`.

If your application contains more than one Spring `ApplicationContext`, you may find that names clash. To solve this problem, you can set the `management.endpoints.jmx.unique-names` property to `true` so that MBean names are always unique.

You can also customize the JMX domain under which endpoints are exposed. The following settings show an example of doing so in `application.properties`:

```
management.endpoints.jmx.domain=com.example.myapp
management.endpoints.jmx.unique-names=true
```

52.2 Disabling JMX Endpoints

If you do not want to expose endpoints over JMX, you can set the `management.endpoints.jmx.exposure.exclude` property to `*`, as shown in the following example:

```
management.endpoints.jmx.exposure.exclude=*
```

52.3 Using Jolokia for JMX over HTTP

Jolokia is a JMX-HTTP bridge that provides an alternative method of accessing JMX beans. To use Jolokia, include a dependency to `org.jolokia:jolokia-core`. For example, with Maven, you would add the following dependency:

```
<dependency>
        <groupId>org.jolokia</groupId>
        <artifactId>jolokia-core</artifactId>
        </dependency>
```

The Jolokia endpoint can then be exposed by adding `jolokia` or `*` to the `management.endpoints.web.exposure.include` property. You can then access it by using `/actuator/jolokia` on your management HTTP server.

52.3.1 Customizing Jolokia

Jolokia has a number of settings that you would traditionally configure by setting servlet parameters. With Spring Boot, you can use your `application.properties` file. To do so, prefix the parameter with `management.endpoint.jolokia.config.`, as shown in the following example:

```
management.endpoint.jolokia.config.debug=true
```

52.3.2 Disabling Jolokia

If you use Jolokia but do not want Spring Boot to configure it, set the `management.endpoint.jolokia.enabled` property to `false`, as follows:

```
management.endpoint.jolokia.enabled=false
```


Prev	Up	Next
51. Monitoring and Management over HTTP	Home	53. Loggers