Module 8 - Lesson 1 - transcript

eugen@baeldung.com

00:00: Welcome to the first module of Course 8: Fundamentals of Monitoring with Spring Boot. As you probably know by now, this is not a Spring Boot heavy course, however, we do want to use the best tool for the job and Spring Boot does bring a lot of great support to the story of metrics and monitoring for a Spring API. So we are definitely going to leverage that. And with that in mind, let's get into the agenda.

00:26: So, first we're going to have a look at Spring Boot actuators and we're going to see how they change monitoring and how they essentially just make things a lot easier. We're then going to get a solid base and a solid understanding of the info endpoint and we're going to see how we can use application properties to really configure that endpoint. We are then going to do the exact same thing for health and for metrics. These are the other two critical foundational endpoints when it comes to actuators.

00:56: All right, let's get into the code. So, the very first thing that we need to do here is, of course, we need to add the actuator dependency into our POM so that we can actually use the actuators. As you can see the Spring Boot Starter Actuator is not defining a version and that's because, of course, it picks up the version from the parent.

01:16: Okay, now that we've added the dependency with the system running but nothing else changed, we're going to hit some of these actuator endpoints and we're going to have a look at the output. So, first thing we're hitting here is this info endpoint and as you can see there is no specific information but the endpoint is working, there's just simply no data and we will add some data later on, but for the time being, we can see the endpoint working and we can see the 200 Okay being returned. Next endpoint we're going to hit is the health endpoint. We get back the 200 Okay and we get back a status of "Up." And again, this is something we'll be able to configure and tune to our own system, but for the time being, the endpoint is responding and the status of the system is considered up. The next and the final endpoint that we're going to hit without any sort of extra configuration is the metrics endpoint. And as you can immediately see, this has a lot more information than the previous two. This returns low level JVM data, but it also returns a few other super interesting things. We'll have information about HTTP sessions here and we're also getting completely out of the box some super interesting HTTP info.

02:32: We're getting the 401 here meaning that the API actually returned a 401 at some point and we're also getting this 200 here, and so we can immediately see how this might be helpful. Getting this low level granular data about our own API and about the types of HTTP responses that the API is serving back to the client, that is a critical type of information that we want to track. We want to know when error rate spike, we want to know when something goes wrong, and all of a sudden the API starts returning status codes in the 400 or 500 range. So, we want to know all of that and this is a great simple way to do it. We are going to look at more complex ways, but this is a super simple and highly useful way to get this information exposed easily and quickly. Because as I mentioned, all I did is I added the actuators dependency, I did not configure anything else. Now that we've seen exactly what the out-of-the-box experience is with this actuator endpoints, we're going to see how we can configure them. Ultimately, it's here, it's in this configuration where things actually start

becoming useful, although, of course, the metrics endpoint is actually quite useful even without any configuration.

03:43: Let's start with configuring the endpoints themselves. So, you can start seeing the syntax. We have endpoints, then we have the name of the endpoint that we're trying to configure and then we have three properties that we can set. We have enabled, which, of course, determines if the endpoint is enabled or not. We have the ID property just to give this endpoint another name. And we have the sensitive property which essentially decides if the endpoint needs to be secured or not. So, for example, let's say that we want to close off the info endpoint. We're going to do this and we're going to say this is no longer enabled. And so now, let's restart the system and let's actually hit this endpoint again and see what happens. Okay, so the endpoint is now disabled on the backend, so we are hitting it and as expected we are getting back the 404 not found. So as expected, this endpoint is no longer available. Okay, so the first thing we need to do is to put this back, or we could have just removed this line because by default the info endpoint is actually enabled, but let's now continue with the configuration.

04:50: Another quick change I always like to do is, I like to change the default context path of these endpoints. So, this is going to mean that the endpoints are now going to be accessible at /management/the name of the endpoint, so no longer at the root level. And the simple reason for that is just to make things a little bit harder to guess if someone from outside the system is trying to hit the defaults, however, of course, that's not really security so keep in mind that any information that you choose to expose via these endpoints should be secured if it's sensitive information. And finally, let's have a couple of configurations specifically for the info endpoint. So, the syntax here is actually quite interesting. We have the name of the endpoint, but after that we just have JSON data that's going to be included in the response of that endpoint. So, this doesn't have any sort of specific syntax, it's just any format that you need in order to expose some information about your app and in this case, we chose to say that the app name is something.

06:04: And this last property is certainly an interesting one because we are using information out of the POM of the project. We are accessing it via the syntax and this actually comes from the package phase of the Maven life cycle. So, this is going to be pulled out of the POM and it's going to be added into the response of our info endpoint. So, let's check this out. First, let's make sure that we're hitting the right endpoint and for that we need to change the path here. So, we're now getting the 200 Okay as expected and let's now check out the body.

06:39: Okay, so right off the bat, you can see the new information. You can now see this app.name equals something presented as JSON information exactly like we were expecting. And you can also see the build version properly resolved to the version of our POM. So we're now starting to get some cool information, some useful information in this info endpoint. And of course, what's interesting here is that a whole lot more is possible with this endpoint. A lot of new information could be added here, of course, depending on what you actually want to expose and depending on what you find useful. But definitely, a flexible and quite interesting endpoint here.

07:17: The next endpoint on the list is the health endpoint. And we're going to only have a very quick look at this one because it's only going to be in the next module where we're going to actually hook into this endpoint and provide some custom feedback. For the time being, we're getting back

the 200 Okay and we're getting back a status of "Up." So, pretty straightforward, but as I said, there is a lot more we can do with the health endpoint and a lot more customization that it allows us to do, but we're going to have a look at that in the next module. So, for the time being, let's keep to the metrics endpoint and let's actually see how the data changes when we start to heed the system a little bit more.

08:00: Okay, so now we have all of this data here, but for now what we really care about is we care about these counters and gauges here related to the HTTP info and what we're going to do is we're going to very quickly run our entire suite of live tests against the API and then we're going to see how this data changes. All right, so now that we've run the full suite of live tests a couple of times, let's actually see what the results are in terms of API metrics.

08:29: And we can clearly see all of that data coming in. We can now see all of these types of responses that the API has served back to the client. We can see successful responses, we can see client errors, we can see a lot of types of responses. Of course, the test suite itself is meant to verify some of these failures, so that's why we're seeing so many errors, but the point is we're now seeing those broken down by status code and by operation. So, very useful information to have in order for us to track what's going on with the API and in order to really be able to assess the overall health of the API.

09:05: Okay, so what did we learn in this first module? We started by looking at Spring Boot actuators and really understanding how this can help in our operations work for the API. We then looked at these three major endpoints and we did some very light customization. Now, of course, there's a lot more customization to be done and that is what we're gonna do in the next module, but this certainly provides a good base to build on. All right. Hope you're excited. See you in the next one.