

Eugen: Welcome to this lesson, out of Learn Spring, where we're going to be focusing on the Spring testing framework, and broadly speaking on the testing support in Spring.

Now, up until this point, we've definitely implemented some functionality into our application, but we haven't yet focused on the core aspect of working in Spring, and really doing software development in general. And that is testing, so that's exactly what we're going to be looking at in this lesson.

[00:00:30] If you haven't done a lot of testing up until this point you may have this question sort of in the back of your mind, what's really the difference between unit testing and integration testing? And while that's a nuanced and complex question, and it's certainly beyond the scope of this lesson here, I will focus on the core aspect that will frame your decision as to what direction to focus on more. And very simply put, that is peace of mind.

So, what do I mean by that? [00:01:00] I mean peace of mind that your application is functioning the way it's supposed to be functioning. Now, to be clear you can, certainly, get there with the both a unit test harness as well as an integration test harness, but typically getting to the point with unit tests is very difficult and, simply put, requires a lot of testing experience whereas getting to the same points with integration tests is a lot easier.

[00:01:30] Of course, remember that unit tests do have other critical advantages and uses. For one thing, they're very fast, they're also a fantastic design tool for your system, so they definitely go beyond pure testing. And they do have many other aspects that make them something that you should be focusing on but, if we just hone in on making sure that our application works and keeps working the way we're expecting it to work then integration [00:02:00] testing is a much better tool for that.

And not only that, but while the Spring framework is certainly built with unit testing and even with DDD in mind the actual testing framework, the actual testing support in Spring really starts being more useful as we go into the integration testing, since that's more or less what lasts what it focused on. So, that's exactly what we're going to be focusing on here.

With that said and out of the [00:02:30] way, let's jump right in and let's start writing some more integration tests. So, let's start by adding the dependencies we need here. We're going to open up the [P-OM 00:02:41] and naturally we're going to be using

Springboard here to bring in the testing support. So, we're going to define two dependencies.

We're adding the test starter here, and we're also going [00:03:00] to add JUnit 5. Now, because we're talking about test focused dependencies let's make sure that we adjust the scope of these dependencies here. And we are good to go with the dependencies.

Now, let's get the folder structure and the packages created as [00:03:30] well.

Remember that tests need to go in a separate folder, the source test Java folder 1 of 2