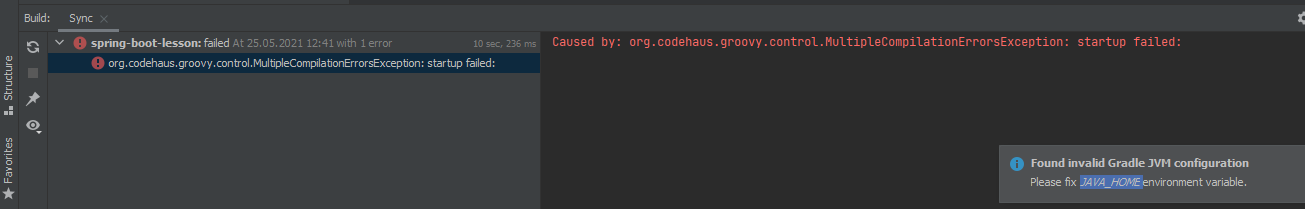
Contents

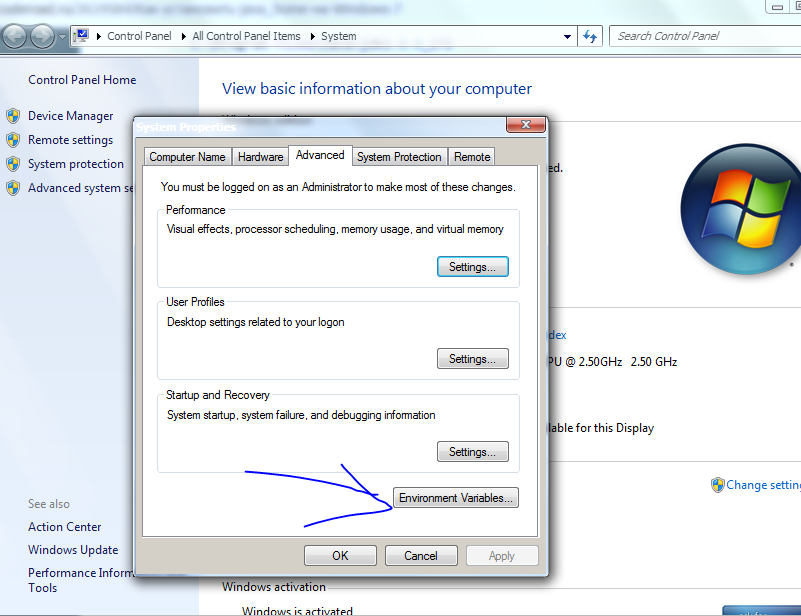
[Lesson 1. First application 2](#_Toc72839024)

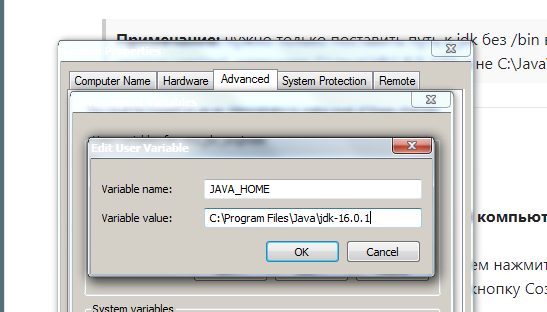
## Lesson1. First application

To create our application we don’t use spring initializer, because it does a lot of things automatically. Instead of this we create empty Gradle project. And here we have the first problem that we didn’t have with maven:



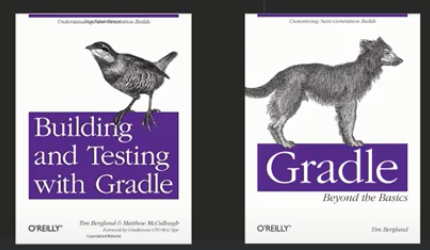
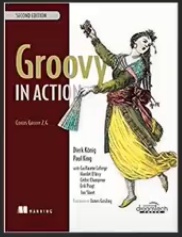
Solution:





To solve compilationError I changed JAVA\_HOME to 15th SDK and created new project with it. 16th SDK throws errors with both Maven and Gradle. Still don’t know why.

Recommended books for Gradle:

Convenient view:

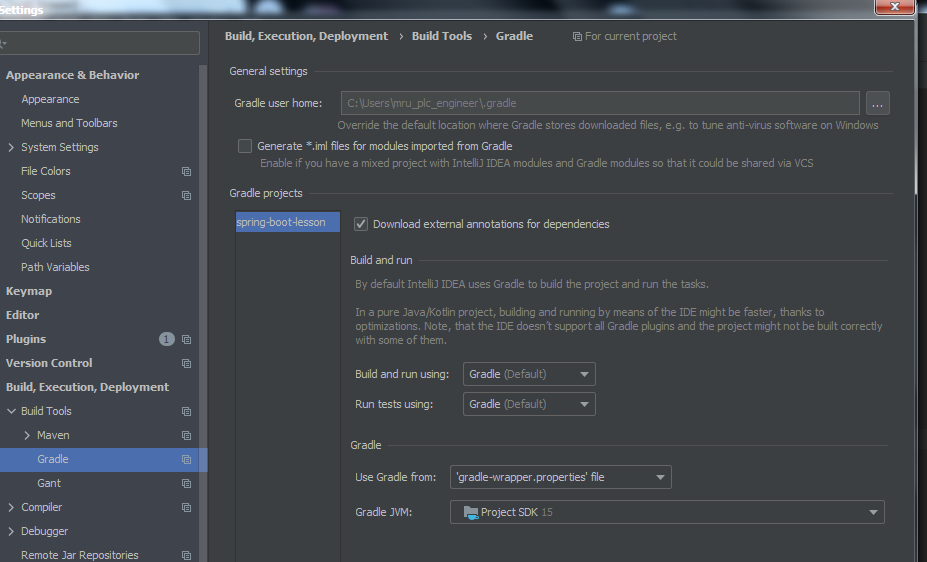
Shift-shift->enter Presentation Mode . And then mouse up View->…->Project. We have nothing but code))

To exit : View->Exit Presentation mode.

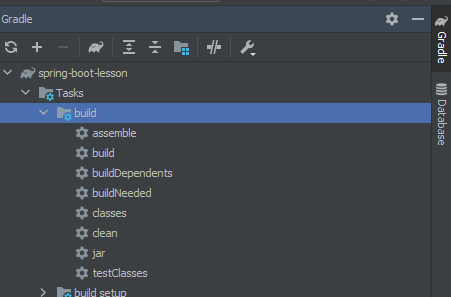
Gradle or Maven wrappers allow us to move our application from one pc to another without bounding to build tool version. We can change wrapper version in property file and it will be pulled up.



We check Gradle JVM version and change if needed.

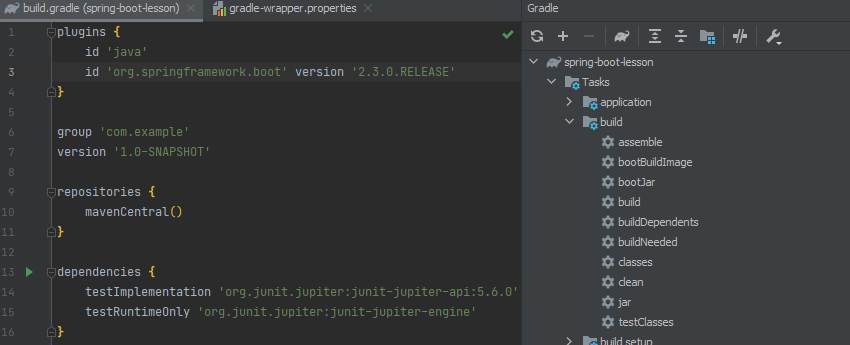


So far we have following application lifecycle:



But it’s an ordinary lifecycle (without using spring boot). Spring boot has different lifecycle. So we add two plugins.

Lifecycle change after adding the first plugin:

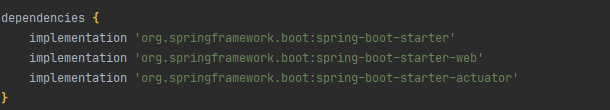


Second plugins helps to avoid collisions of dependency versions:

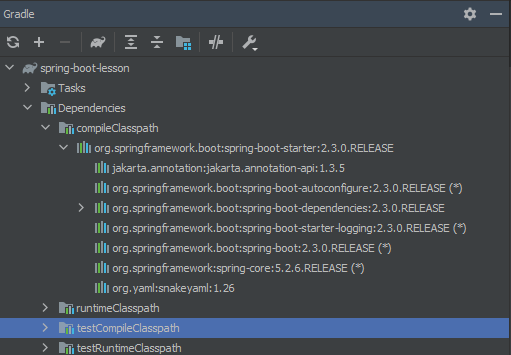


With that plugin we don’t even need to specify versions for most of our libraries.

For beginning we add following dependencies:



IDEA with Gradle shows all used dependencies grouped by scope:



We add default springboot class with main method and run the app:

