

# Spring Boot



provided by Ze Avengers

# Contents

What is Spring Boot?

Components of Spring Boot?

Development setup

Hello World Application

# What is Spring Boot / Ze Avengers

**Spring Boot** makes it easy to create stand-alone, production-grade **Spring** based Applications that you can "just run".

We take an opinionated view of the **Spring** platform and third-party libraries so you can get started with minimum fuss. Most

**Spring Boot** applications need very little **Spring** configuration.

## Features

- Create stand-alone **Spring** applications:

*Spring is a popular Java application framework. It provides various libraries and tools for enterprise application programming. It is also a very good integration system that helps glue together various enterprise components.*

- Embed Tomcat\*, Jetty\* or Undertow\* directly (no need to deploy WAR files(**Web Application Resource** or **Web Application Archive**))

*\* Those are Java Servers*

# What is Spring Boot / Ze Avengers

- Provide opinionated 'starter' dependencies to simplify your build configuration

*Spring Boot starters can help to reduce the number of manually added dependencies just by adding one dependency. So instead of manually specifying the dependencies just add one starter as in the following example:*

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

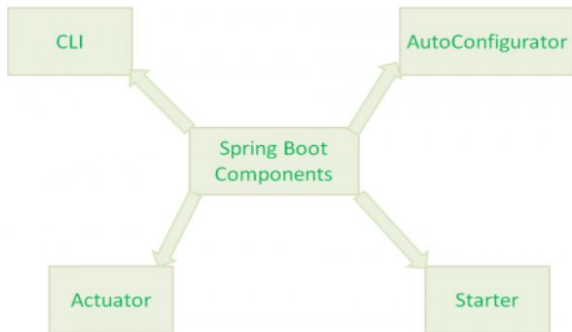
- Automatically configure **Spring** and 3rd party libraries whenever possible
- Provide production-ready features such as metrics, health checks and externalized configuration

Done by: **Spring Boot Actuator module**

- Absolutely no code generation and no requirement for XML configuration

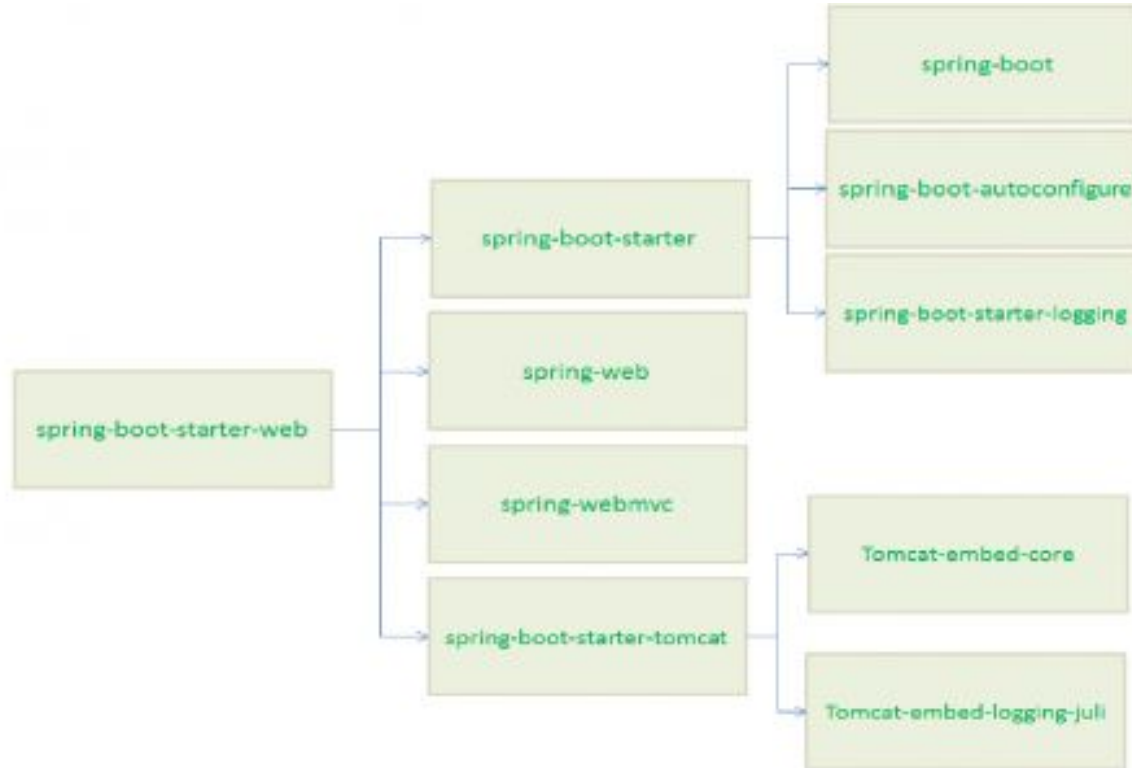
# Components of Spring Boot

Spring Boot Framework has mainly four major components.



**Spring Boot Starters:** The main responsibility of Spring Boot Starter is to combine a **group of common or related dependencies** into single dependencies. Spring Boot starters can help to reduce the number of manually added dependencies just by adding one dependency. So instead of manually specifying the dependencies just add one starter. Examples are **spring-boot-starter-web**, **spring-boot-starter-test**, **spring-boot-starter-data-jpa**, etc.

# Components of Spring Boot



# Components of Spring Boot


**Spring Boot AutoConfigurator:** One of the common complaint with Spring is, we need to make lot of XML based configurations. Spring Boot AutoConfigurator will simplify all these XML based configurations. It also reduces the number of annotations.



```
@Target(value=TYPE)
@Retention(value=RUNTIME)
@Documented
@Inherited
@Configuration
@EnableAutoConfiguration
@ComponentScan
public @interface SpringBootApplication
```

# Components of Spring Boot

**Spring Boot CLI:** Spring Boot CLI(Command Line Interface) is a Spring Boot software to run and test Spring Boot applications from command prompt. When we run Spring Boot applications using CLI, then it internally uses Spring Boot Starter and Spring Boot AutoConfigure components to resolve all dependencies and execute the application.



```
spring run HelloWorld.groovy
```

**Spring Boot Actuator:** Spring Boot Actuator is a sub-project of Spring Boot. It adds several production grade services to your application with little effort on your part. Actuators enable production-ready features to a Spring Boot application, without having to actually implement these things yourself. The Spring Boot Actuator is mainly used to get the internals of running application like health, metrics, info, dump, environment, etc. which is similar to your production environment monitoring setup.



# Development Environment Setup

## Step 1: Install Java JDK

- [Download JDK from Oracle](#)
- Double click to install and follow directions
- Add the JDK bin directory to your path
- Set the environment variable JAVA\_HOME to point to your JDK installation

## Step 2: Install Gradle

- [Download the binary-only Gradle package](#)
- Unzip the file
- Move the gradle folder anywhere to your C: drive (**C:\Program Files** for example)
- Add the Gradle bin directory to your path

## Step 3: Use your favorite Editor or IDE

- **VSCode**, Eclipse, IntelliJ

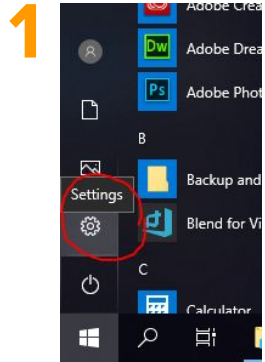
# Development Environment Setup

[Download Oracle JDK](#)

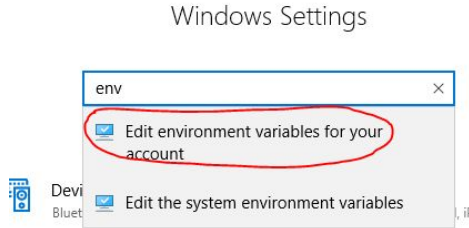
Java Platform, Standard Edition	
<b>Java SE 12</b> Java SE 12 is the latest release for the Java SE Platform <a href="#">Learn more</a> ▶	
<ul style="list-style-type: none"><li>▪ <a href="#">Installation Instructions</a></li><li>▪ <a href="#">Release Notes</a></li><li>▪ <a href="#">Oracle JDK License</a></li><li>▪ <a href="#">Java SE Licensing Information User Manual</a><ul style="list-style-type: none"><li>▪ Includes Third Party Licenses</li></ul></li><li>▪ <a href="#">Certified System Configurations</a></li><li>▪ <a href="#">Readme</a></li></ul>	<div><b>Oracle JDK</b> <b>DOWNLOAD</b> ⬇</div>

# Development Environment Setup

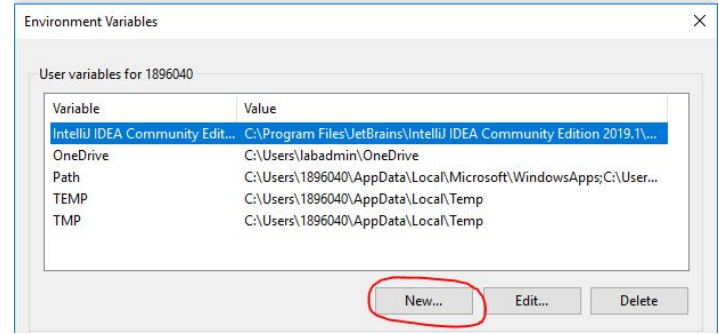
Add the JDK bin directory to your path



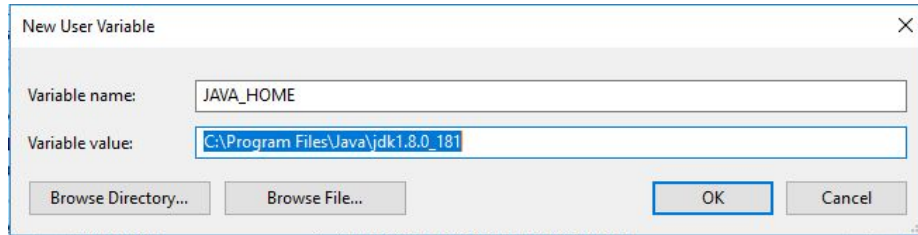
2



3

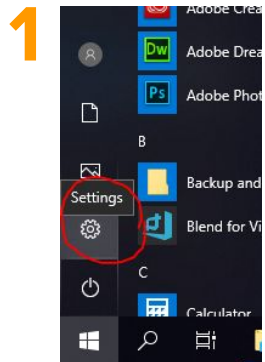


4

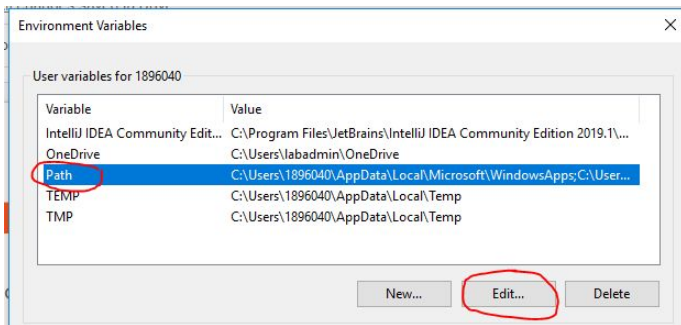


# Development Environment Setup

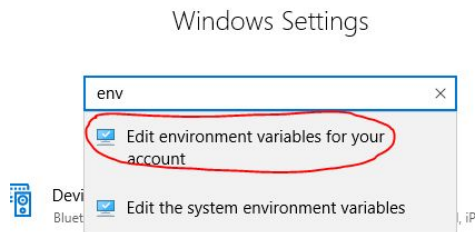
Set JAVA\_HOME to point to your JDK installation



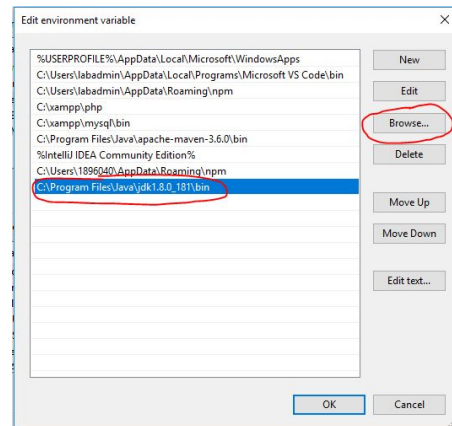
2



3



4



# What is Gradle

**Gradle** is an open-source [build automation](#) tool that is designed to be flexible enough to build almost any type of software.

## What is Gradle?



- Gradle is a general purpose **build system**
- It comes with a rich build description language (DSL) based on **Groovy**
- It supports "**build-by-convention**" principle
- But it is very **flexible** and **extensible**
- It has **built-in plug-ins** for Java, Groovy, Scala, Web, OSGi
- It derives all the best and integrates well with **Ivy, Ant and Maven**

For more detailed explanation, please read the documentation at:

[https://docs.gradle.org/current/userguide/what\\_is\\_gradle.html](https://docs.gradle.org/current/userguide/what_is_gradle.html)

# What is Gradle

## Overview

Gradle is an open-source [build automation](#) tool that is designed to be flexible enough to build almost any type of software. The following is a high-level overview of some of its most important features:

### **High performance**

Gradle avoids unnecessary work by only running the tasks that need to run because their inputs or outputs have changed. You can also use a build cache to enable the reuse of task outputs from previous runs or even from a different machine (with a shared build cache).

There are many other optimizations that Gradle implements and the development team continually work to improve Gradle's performance.

### **JVM foundation**

Gradle runs on the JVM and you must have a Java Development Kit (JDK) installed to use it. This is a bonus for users familiar with the Java platform as you can use the standard Java APIs in your build logic, such as custom task types and plugins. It also makes it easy to run Gradle on different platforms.

Note that Gradle isn't limited to building just JVM projects, and it even comes packaged with support for building native projects.

# What is Gradle

## Conventions

Gradle takes a leaf out of Maven's book and makes common types of projects — such as Java projects — easy to build by implementing conventions. Apply the appropriate plugins and you can easily end up with slim build scripts for many projects. But these conventions don't limit you: Gradle allows you to override them, add your own tasks, and make many other customizations to your convention-based builds.

## Extensibility

You can readily extend Gradle to provide your own task types or even build model. See the Android build support for an example of this: it adds many new build concepts such as flavors and build types.

## IDE support

Several major IDEs allow you to import Gradle builds and interact with them: Android Studio, IntelliJ IDEA, Eclipse, and NetBeans. Gradle also has support for generating the solution files required to load a project into Visual Studio.

## Insight

[Build scans](#) provide extensive information about a build run that you can use to identify build issues. They are particularly good at helping you to identify problems with a build's performance. You can also share build scans with others, which is particularly useful if you need ask for advice in fixing an issue with the build.

# Development Environment Setup

## Install Gradle

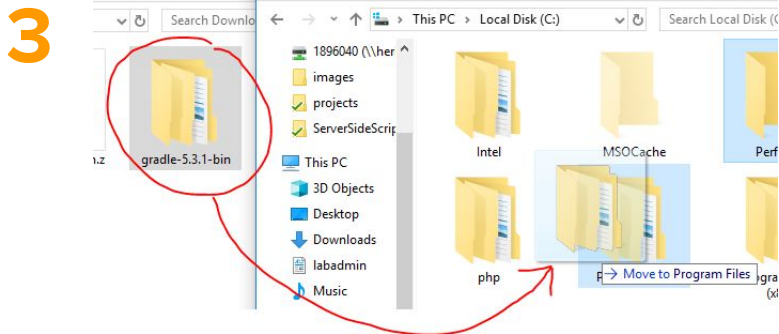
### 1 Installing manually

Step 1. Download the latest Gradle distribution

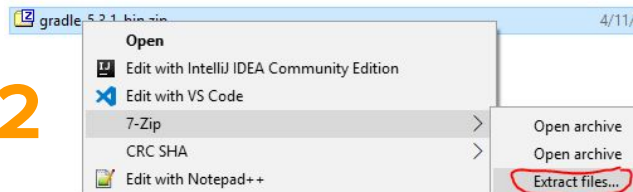
The current Gradle release is version 5.3.1, released on 2019-04-11, and comes in two flavors:

- Binary-only
- Complete, with docs and sources

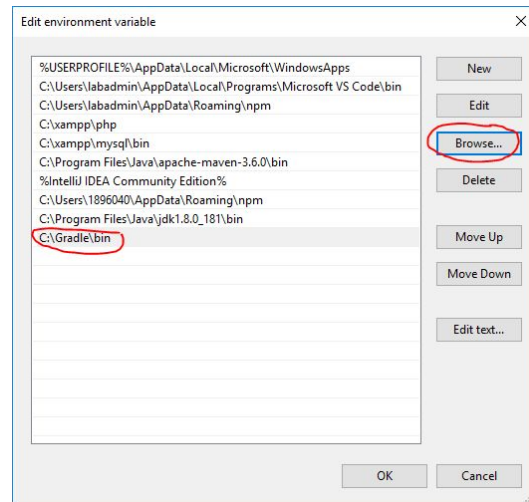
If in doubt, choose the binary-only version and browse



2



4





# Hello world app / Ze Avengers

Creating the Hello World app is fairly easy following all the steps and available documentation at:

<https://spring.io/guides/gs/spring-boot/>



**Ze Avengers** wish you happy programming with **Spring Boot** and **Java!!!**

**Did you like the framework?**

