

Professional Skills
Agile Fundamentals
Jira
Git
Databases Introduction
Java Beginner
Maven
Testing (Foundation)
Java Intermediate
HTML
CSS
Javascript
Spring Boot
<div><div></div>Introduction to Spring Boot</div>
<div><div></div>Multi-Tier Architecture</div>
<div><div></div>Beans</div>
<div><div></div>Bean Scopes</div>
<div><div></div>Bean Validation</div>
<div><div></div>Dependency Injection</div>
<div><div></div>Components</div>
<div><div></div>Configuration</div>
<div><div></div>Connecting to a Database</div>
<div><div></div>Entities</div>
<div><div></div>Postman</div>
<div><div></div>Controllers</div>
<div><div></div>Services</div>
<div><div></div>Repositories</div>
<div><div></div>Custom Queries</div>
<div><div></div>Data Transfer Objects</div>
<div><div></div>Lombok</div>
<div><div></div>Custom Exceptions</div>
<div><div></div>Swagger</div>
<div><div></div>Profiles</div>
<div><div></div>Pre-Populating Databases for Testing</div>
<div><div></div>Unit testing with Mockito</div>

# Introduction to Spring Boot

## Contents

- [Overview](#)
- [Installation](#)
  - [Standalone](#)
  - [Eclipse](#)
- [Tutorial](#)
- [Exercises](#)

## Overview

Spring is an application framework and *inversion-of-control* container for Java.

The framework can be used by any Java project, but there are extensions for building web applications on top of the Java EE platform.

The Spring framework is divided into modules, known as *projects*, which allow developers to pick and choose which parts of the framework they wish to use.

There are 20+ specialised projects within the Spring framework, each used for different purposes:

- Accessing data in a consistent manner (**Spring Data**).
- Deploying microservice applications to a cloud (**Spring Cloud**).
- Securing applications against attacks (**Spring Security**).
- Rapid creation of low-configuration projects; with the intent of running them 'out-of-the-box' (**Spring Boot**).

## Installation

While a Spring project is, at the end of the day, just a Java project and can be built using a standard Java IDE (like Eclipse) there are specific IDEs available which can make the task a bit simpler for developers. Spring Tools Suite is one such IDE, built off of Eclipse by Pivotal (the company behind Spring).

Spring Tools Suite can be downloaded either as a standalone IDE, or can be installed as an extension to Eclipse. (You may also install it for IntelliJ, VSC, or Atom, but these are not recommended and are not covered in this tutorial.)

*note: no matter which method you choose, ensure that you have set up your workspace in an easy-to-access area of your system; e.g. 'C:/Users/admin/Desktop/spring-workspace' or similar.*

## Standalone

Navigate to [the Spring Tools page](#), then click the relevant download for your system:

This will download a **tar.gz** (Linux), **.dmg** (macOS), or self-extracting **.jar** (Windows).

Extract/Open this file; this should generate a folder containing the installation.

Inside this folder (e.g. **sts-4.8.1-RELEASE**), run the **SpringToolSuite4** application:

This should open the Spring Tool Suite IDE.

<div><div></div><div>Testing</div></div>
Selenium
Sonarqube
Advanced Testing (Theory)
Cucumber
MongoDB
Express
NodeJS
React
Express-Testing
Networking
Security
Cloud Fundamentals
AWS Foundations
AWS Intermediate
Linux
DevOps
Jenkins Introduction
Jenkins Pipeline
Markdown
IDE Cheatsheet

## Eclipse

Spring Tools can be installed as an Eclipse extension by navigating to [this marketplace link](#) and dragging the **Install** button inside of a running Eclipse:

This will import and automatically install Spring according to your Eclipse configuration.

## Tutorial

In this tutorial, we'll build our first Spring application.

Once your workspace has loaded, go to **File > New > Project...**, then choose **Spring Starter Project** from the list:

On the following screen, fill out the details of your project as follows:

- **Name** can be anything you wish (in PascalCase), e.g. **StarterProject**.
- Ensure that you're saving your project to your spring workspace.
- **Type** refers to the build tool, which should be **Maven**.
- **Packaging** refers to the archive created by the build tool, which should be **Jar**.
- Ensure that the **Java version** is at least **8** (preferably **11** or higher, as Java 8 is starting to become obsolete).
- Of course, keep the **Language** as **Java**.
- The **Group** should match your organisation, e.g. **com.qa**.
- The **Artifact** should match the **Name**.
- Keep the **Version** as-is for now.
- The **Package** should be your **Group** field plus your project **Name** (all in lowercase), e.g. **com.qa.starterproject**.

Ignore everything else, then click **Next**.

The following screen allows you to select any Spring framework *projects* you wish to use.

From the list, select **Spring Data JPA**, **Spring Web**, and **H2 Database**, then click **Finish**.

Spring will take time to set the project up - you can see its progress in the bottom-right of your screen.

Once done, you should be able to expand out your project in the **Project Explorer**:

## Exercises

There are no exercises for this module.