### Introduction to Spring Boot

**Introduction to Spring Boot**

Java is one of the most popular and widely used programming languages, powering everything from mobile applications to large-scale enterprise systems. Since Java can power almost any kind of development, numerous frameworks and tools have evolved to simplify and streamline Java-based software developments.

When it comes to building Java-based web applications, Spring is one of the most popular frameworks. In this article, we will look at the Spring framework and how to build Java applications using it.

**What is the Spring Framework?**

The Spring Framework is an open-source framework for building enterprise Java applications. Spring aims to simplify the complex and cumbersome enterprise Java application development process by offering a framework that includes technologies such as:

Aspect-oriented programming (AOP)

Dependency injection (DI)

Plain Old Java Object (POJO)

Even with all these technologies, Spring is a lightweight framework that can be used to create scalable, secure, and robust enterprise web applications.

At a macro-level, we can consider the Spring framework a collection of sub frameworks such as Spring Web Flow, Spring MVC, and Spring ORM. In addition to Java, Spring also supports Kotlin and Groovy.

The Spring framework is also the base that powers all the other Spring-based projects, such as:

Spring Boot

Spring Cloud

Spring GraphQL

**Features of Spring - Inversion Of Control, Dependency Injection**

Core Features of Spring Framework

In software engineering, inversion of control (IoC) is a programming technique in which object coupling is bound at run time by an assembler object and is typically not known at compile time using static analysis. In this spring tutorial, learn the difference between ioc and dependency injection in spring with example.

1. What is Inversion of Control (IoC)

In traditional programming, the flow of the business logic is determined by objects that are statically assigned to one another. With inversion of control, the flow depends on the object graph that is instantiated by the assembler and is made possible by object interactions being defined through abstractions. The binding process is achieved through dependency injection, although some argue that the use of a service locator also provides inversion of control.

Inversion of control as a design guideline serves the following purposes:

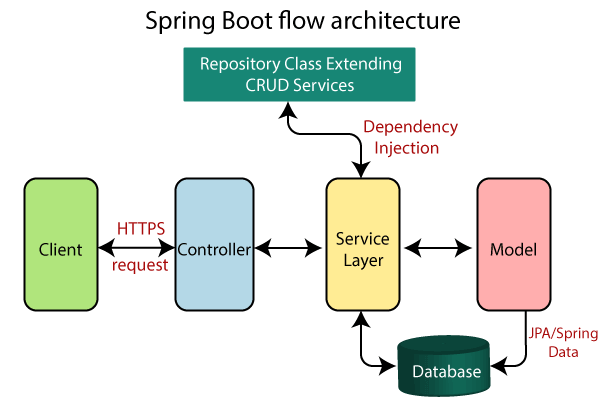
1. There is a decoupling of the execution of a certain task from implementation.
2. Every module can focus on what it is designed for.
3. Modules make no assumptions about what other systems do but rely on their contracts.
4. Replacing modules has no side effect on other modules.

2. What is Dependency Injection (DI)

IoC is a design paradigm with the goal of giving more control to the targeted components of your application, the ones getting the work done. While Dependency injection is a pattern used to create instances of objects that other objects rely on without knowing at compile time which class will be used to provide that functionality. IoC relies on dependency injection because a mechanism is needed in order to activate the components providing the specific functionality.

The two concepts work together in this way to allow for much more flexible, reusable, and encapsulated code to be written. As such, they are important concepts in designing object-oriented solutions.

**Spring Boot Architecture**



1. The Client makes an HTTP request(GET, PUT, POST, etc.)
2. The HTTP request is forwarded to the Controller. The controller maps the request. It processes the handles and calls the server logic.
3. The business logic is performed in the Service layer. The spring boot performs all the logic over the data of the database which is mapped to the spring boot model class through Java Persistence Library(JPA).
4. The JSP page is returned as Response from the controller.

**What is Maven?**

Maven is a popular open-source build tool developed by the Apache Group to build, publish, and deploy several projects at once for better project management. The tool allows developers to build and document the lifecycle framework.

Maven is written in Java and is used to build projects written in C#, Scala, Ruby, etc. Based on the Project Object Model (POM), this tool has made the lives of Java developers easier while developing reports, checks build and testing automation setups.

Maven focuses on the simplification and standardization of the building process,

Maven is chiefly used for Java-based projects, helping to download dependencies, which refers to the libraries or JAR files. The tool helps get the right JAR files for each project as there may be different versions of separate packages.

After Maven, downloading dependencies doesn’t require visiting the official websites of different software. You can visit mvnrepository to find libraries in different languages. The tool also helps to create the right project structure in struts, servlets, etc., which is essential for execution.

Maven consists of Project Object Model (POM), which is an XML file that has all the information regarding project and configuration details. The POM has the description of the project, details regarding the versioning, and configuration management of the project.

The XML file is located in the project home directory. When you execute a task, Maven searches for the POM in the current directory.

**Create a simple project with Maven**

1. Create the folder and open in VS code

2. In this folder add the hierarchy of folder

src/main/java/hello

3. Create a new file HelloWorld.java

package hello;

public class HelloWorld{

public static void main(String[] args) {

System.out.println("hello world");

}

}

4. Create pom.xml file in the project folder, outside src folder

(get the content for this file from https://spring.io/guides/gs/maven/ link, Define a simple Maven build)

5. Let make some changes in the pom.xml file

<groupId>com.practice</groupId>

<artifactId>demomaven</artifactId>

<packaging>jar</packaging>

<version>0.1.0</version>

<properties>

<maven.compiler.source>11</maven.compiler.source>

<maven.compiler.target>11</maven.compiler.target>

</properties>

6. Close the folder and reopen it so the visual studio code can configure the maven project

7. Download and place the maven installation files in a folder

8. In terminal, type the command

..\apache-maven-3.8.5\bin\mvn -v

to know the version of the maven installed

..\apache-maven-3.8.5\bin\mvn compile

to compile to project, in the target folder class files will be generated

..\apache-maven-3.8.5\bin\mvn package

to create the jar files

java -jar .\target\demomaven-0.1.0.jar

to execute the jar file created in the previous command

9. To add external jars in the project, find the corresponding xml from maven repository:

https://mvnrepository.com/

Search for the library to be added eg. joda time

Copy the xml code and paste in pom.xml file

<dependencies>

<!-- https://mvnrepository.com/artifact/joda-time/joda-time -->

<dependency>

<groupId>joda-time</groupId>

<artifactId>joda-time</artifactId>

<version>2.10.14</version>

</dependency>

</dependencies>

import org.joda.time.LocalTime;

public class HelloWorld{

public static void main(String[] args) {

LocalTime currentTime = new LocalTime();

System.out.println("hello world " + currentTime);

}

}

10. Recompile and run the code

**Install STS 4 (Spring Tool Suite)**

Spring Tool Suite (STS) is a java IDE tailored for developing Spring-based enterprise applications. It is easier, faster, and more convenient. And most importantly it is based on Eclipse IDE. Make sure you have installed Java Development Kit (JDK) version 8 or newer.

These are steps to be followed sequentially for installing STS$:

Download SpringToolSuite as per the operating system to the local machine.

Move the downloaded JAR file to the corresponding folder.

Unzip this JAR file and open the corresponding folder.

Click on the SpringToolSuite4 Application file

Select the directory representing workspace and press the ‘LAUNCH’ button.

**Create a project with spring boot starter**

Create a Spring Boot Project in Spring Initializr

Go to https://start.spring.io/ and create a Spring Boot project. Please fill in all the details accordingly.

Service URL: Default

Name: Your Project Name

Type: Maven Project

Java Version: 11 or greater than 11

Packaging: As your need

Language: As your need

Group: A unique base name of the company or group that created the project

### Artifact: A unique name of the project

### Version: Default

### Description: As your need

### Package: Your package name

### Lastly click on the GENERATE button below. This will download your Spring Boot project in zip format. Now extract the folder into your local machine.