



Spring Framework Fast Track 2019

A complete of essential concepts and examples

Author: Erfin Feluzy & Wahyu Sudrajat
March 2019

Profile

ERFIN FELUZY

(<https://github.com/erfinfeluzy>)

- 11+ years professional experience in Java Programming (10+ years spring framework)
- General Manager in Software Solution
- Sr Middleware Solution Architect

WAHYU SUDRAJAT

- 8+ years professional experience in Spring
- Linux master & Mathematician
- Solution Architect in Software Solution



Agenda Day 1 (March 9, 2019)

09:00 – 10:00 : Spring Framework intro

10:00 – 10:30 : Spring Boot intro (Hands-On)

10:30 – 11:00 : Basic build with Maven (Hands-On)

11:00 – 12:00 : Hands-On prerequisite setup

12:00 – 13:00 : Ishoma

13:00 – 14:00 : Spring web MVC (Hands-On)

14:00 – 15:00 : Spring Data Database (Hands-On)

15:00 – 16:00 : Spring Web Service REST producer (Hands-On)

16:00 – 17:00 : Spring Web Service SOAP producer (Hands-On)

Agenda Day 2 (March 16, 2019)

09:00 – 10:00 : Spring Web Service REST consumer (Hands-On)

10:00 – 11:00 : Spring Web Service SOAP consumer (Hands-On)

11:00 – 12:00 : Spring JMS Publisher & Subscriber (Hands-On)

12:00 – 13:00 : Ishoma

13:00 – 14:30 : Lab : Use Case 1

14:30 – 16:00 : Lab : Use Case 2

16:00 – 17:00 : Recap & Closing



What is Spring?

Back then, enterprise application commonly use a technology called **Enterprise Java Bean (EJB)** which relatively heavy and tightly depend on enterprise solution such as: Application Web Server(s).

What is Spring?

- An open-source framework.
- An alternative to heavier enterprise Java technologies.
- Addresses the complexity of enterprise application development, thus it “Simplifies Java development”

<https://www.youtube.com/watch?v=a14xfNV-UIs>



Spring is Non-Invasive

What does that mean?

- You are not forced to import or extend any Spring APIs
- An invasive framework takes over your code.
- Anti-pattern: (sample of invasive framework)
 - **EJB** forces you to use **Java Naming and Directory Interface (JNDI)**, commonly found at enterprise application framework
 - **Struts** forces you to extend **Action, ActionForm**

Invasive frameworks are inherently difficult to test. You have to stub the runtime that is supplied by the application server.

<https://www.slideshare.net/pmanvi/spring-framework-overview-ppt-3068245>

Benefit of Using Spring

Spring is a Framework which help to solve common software developing problems with Java programming language, such as:

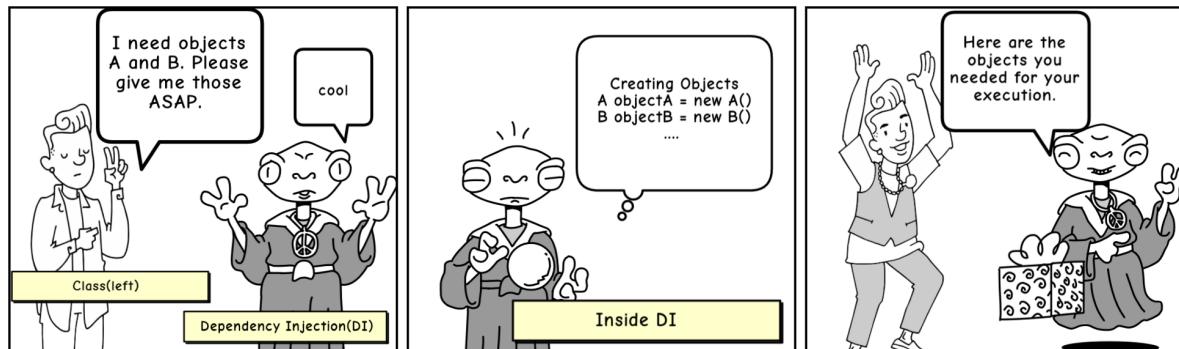
- Manage lifecycle of Java classes (called beans)
- Dependency Injection framework
- Provide Data Access which commonly works through Java Database Connectivity (JDBC), Object-Relational Mapping (ORM), etc.
- Provide Spring-MVC which standardize patterns and structure to develop Web Application
- more...

<https://www.youtube.com/watch?v=gq4S-ovWVIM>

Dependency Injection (DI)

Wikipedia definition of DI:

*“In software engineering, **dependency injection** is a technique whereby one object (or static method) supplies the dependencies of another object. A dependency is an object that can be used (a service)”*



This comic was created at www.MakeBeliefsComix.com. Go there and make one now!

<https://medium.freecodecamp.org/a-quick-intro-to-dependency-injection-what-it-is-and-when-to-use-it-7578c84fa88f>

What is Spring Boot?

Although Spring does simplify Java development process, it can give programmers an adversity while developing an application.

- Boiler plate configuration
 - Programmer writes a lot of codes (configurations) to do minimal task
- Takes time to have Spring application up and running

Spring Boot offers the solution for those problems. Spring Boot is a suite of pre-configured framework and technologies and it is a shortest way to have Spring application up and running.

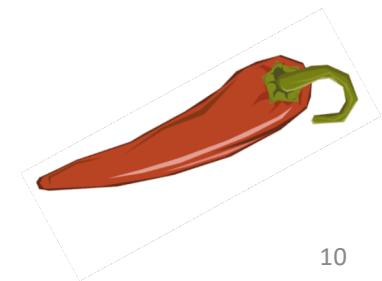
<https://www.youtube.com/watch?v=LLfZ2g2Jalc>

Hand-On Prerequisites Setup

- Oracle SDK 8+, you might also use OpenJDK, make sure to have JDK installed instead of JRE
- Java IDE, Eclipse 2018-12 is preferred
 - Eclipse by default comes with Apache Maven embedded
- Properly setup Java Runtime Environment (JRE) on Environment Variable (Windows), or PATH variable (Unix based)
- Lombok Project for Simplify POJO

OpenJDK

eclipse

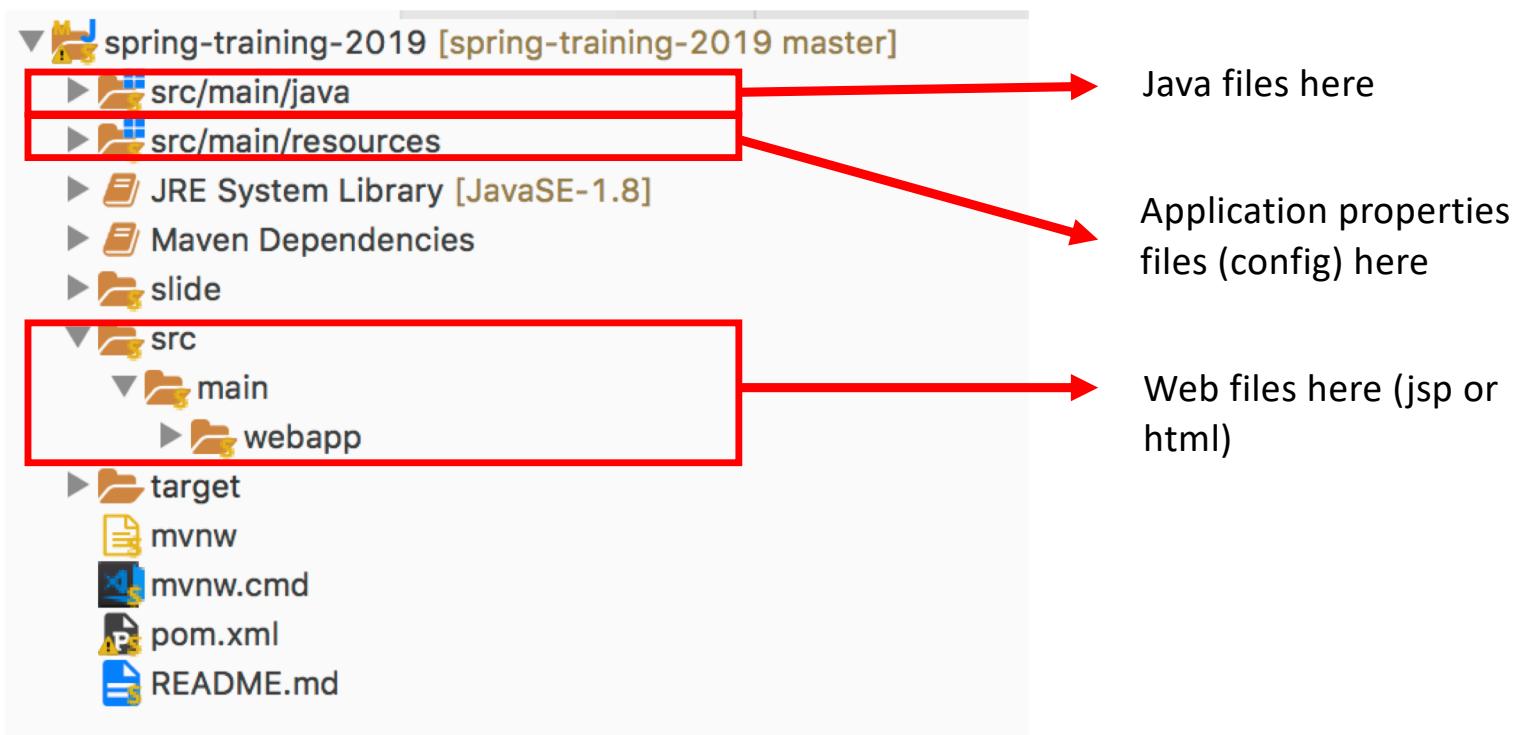




Spring Boot Hands-On

Spring Boot Intro

Maven Common Project Structure



Spring Boot's pom.xml

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>1.4.3.RELEASE</version>
</parent>
```

Spring boot parent artifact

```
<dependencies>
  <!-- Spring Boot -->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
```

Based on web MVC

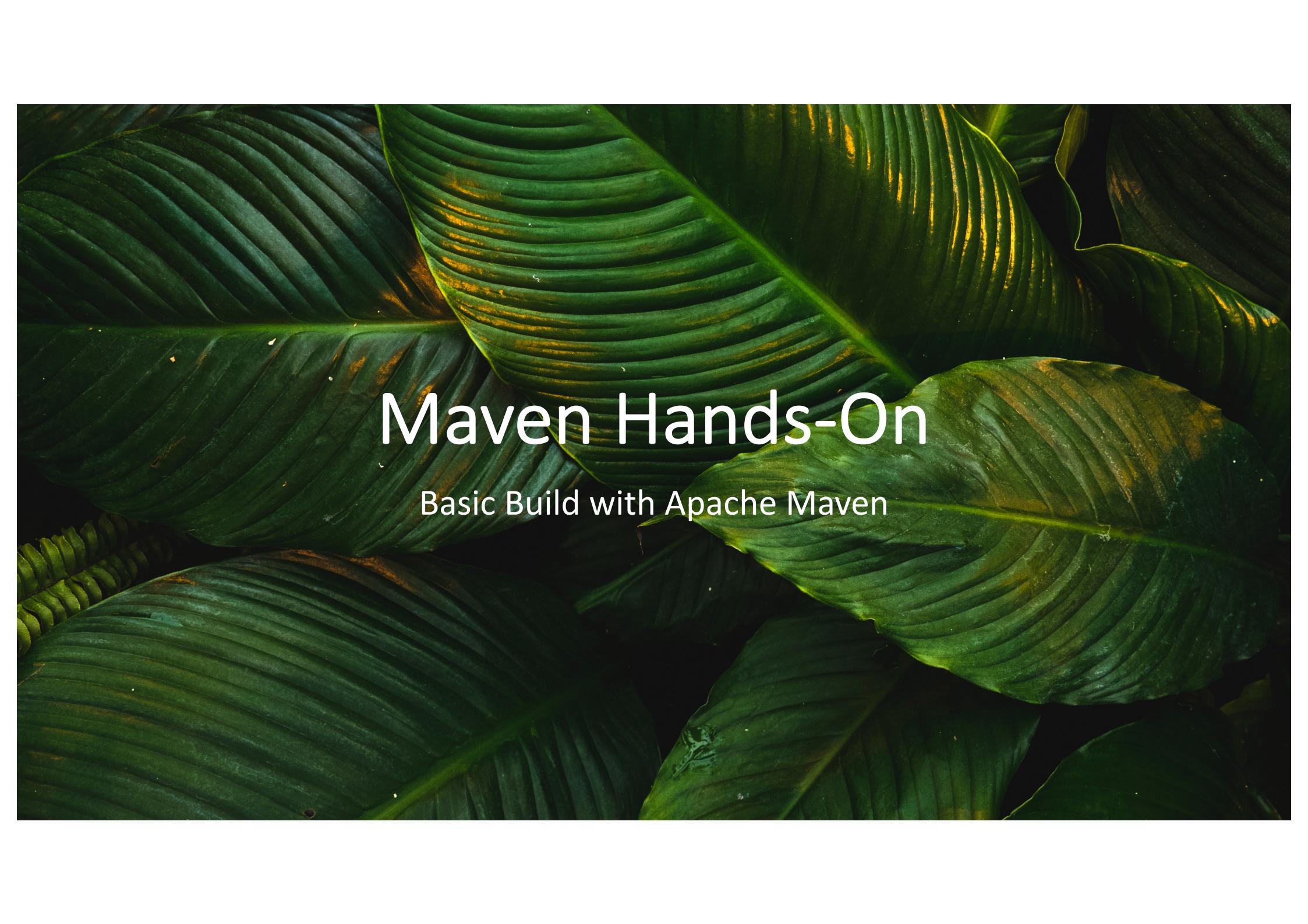
Spring Boot Starter

```
J Application.java ✘
1 package com.erfinfeluzy.training;
2
3 import org.springframework.boot.SpringApplication;
4
5
6 @SpringBootApplication(scanBasePackages = { "com.erfinfeluzy" })
7
8 @PropertySource({
9     "classpath:apps.properties"
10 })
11
12 public class Application {
13
14     public static void main(String[] args) {
15
16         SpringApplication.run(Application.class, args);
17     }
18
19 }
20
21
```

Scan all spring bean inside base package

Spring boot custom configuration.
Eg: server port, database, etc

Spring boot custom configuration.
Eg: server port, database, etc



Maven Hands-On

Basic Build with Apache Maven

Apache Maven

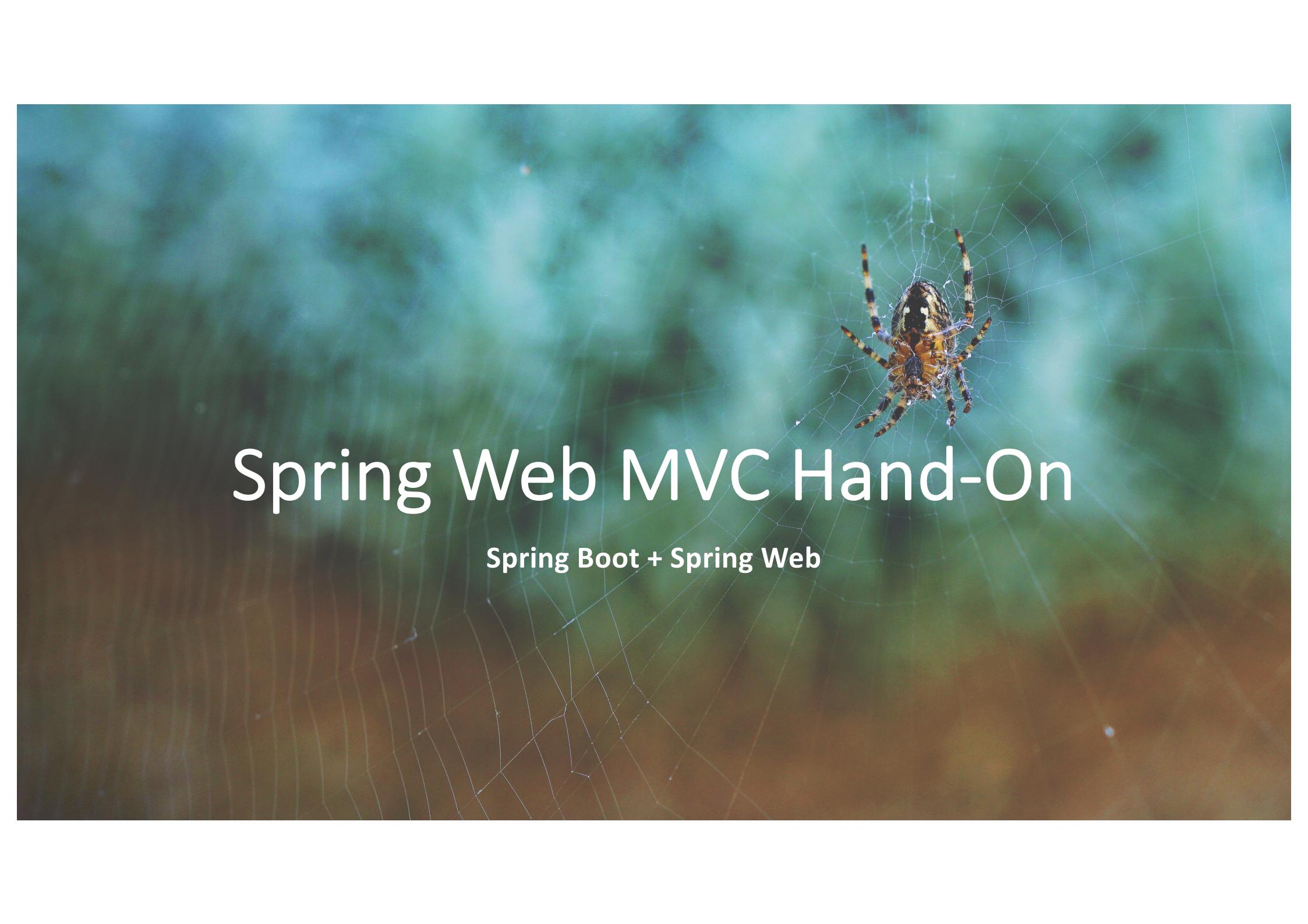
Maven is a software management and comprehension tool based on the concept of Project Object Model(POM). It can manage project build, reporting and documentation, from a central piece of information. Basically, this tool can be used as both, a build tool (just like ANT tool), as well as a project management tool. However, in this article we will use it just for building our project.

Basic usage of maven are:

```
$> mvn clean  
$> mvn compile  
$> mvn install
```

<https://medium.com/@himanshuagarwal1395/getting-started-with-apache-maven-hello-world-eccb278a262a>



A photograph of a spider with a patterned body and long legs, resting on a delicate, intricate web. The background is a soft-focus green, suggesting a natural outdoor setting.

Spring Web MVC Hand-On

Spring Boot + Spring Web

Spring MVC (Model-View-Controller)

```
J Customer.java ✘
1 package com.erfinfeluzy.training.spring.model;
2
3 import java.util.Date;
4
5 @Data
6 @Entity
7 @Table(name = "tbl_customer")
8 public class Customer {
9
10     @Id
11     @GeneratedValue
12     private Long id;
13
14     @Column(name="username")
15     private String username;
16
17     @Column(name="first_name")
18     private String firstName;
19
20     @Column(name="last_name")
21     private String lastName;
22
23     @Column(name="birth_date")
24     private Date birthdate;
25 }
```

model

```
@Controller
public class HomeController { controller
    @RequestMapping(value = "/", method = RequestMethod.GET)
    public String home() {
        System.out.println("masuk : / ");
        return "home";
    }
}
```

```
home.jsp ✘
1 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
2 <html>
3
4     <head>
5         <title>Training</title>
6     </head>
7
8     <body>
9
10        <h2>Halo Trainee</h2>
11
12        <p>greetings: ${greetings}</p>
13
14    </body>
15
16 </html>
```

view

A close-up photograph of a yellow Ethernet cable against a solid blue background. The cable is coiled and curves across the frame, with its white RJ45 connector visible at the top center.

Spring Data Hand-On

Spring Boot + Spring Data

Spring Web Service REST Producer Hand-On

Spring Boot + Spring Web



Spring Web Service SOAP Producer Hand-On

Spring Boot + Spring Web



Spring Web Service REST Consumer Hand-On

Spring Boot + Spring Web

Spring Web Service SOAP Consumer Hand-On

Spring Boot + Spring Web

Spring JMS Publisher & Subscriber Hand-On

Spring Boot + Spring JMS

Lab – Use Case 1

Create Multiple Backend Web Services

Lab – Use Case 2

Create Microservices apps with Spring Boot

Recap & Closing