

UNDERSTANDING THE WORLD BEFORE SPRING BOOT

- Setting up Spring Projects before Spring Boot was NOT easy!
- We needed to configure a lot of things before we have a production-ready application.

1. Dependency Management

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-webmvc</artifactId>
  <version>6.2.2.RELEASE</version>
</dependency>
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-databind</artifactId>
  <version>2.13.3</version>
</dependency>
<dependency>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>1.2.17</version>
</dependency>
```

- Manage frameworks and versions.
 - REST API - Spring framework, Spring MVC framework, JSON binding framework, ...
 - Unit Tests - Spring Test, Mockito, JUnit, ...

2. Web.xml

```
<servlet>
  <servlet-name>dispatcher</servlet-name>
  <servlet-class>
    org.springframework.web.servlet.DispatcherServlet
  </servlet-class>
  <init-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/todo-servlet.xml</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
  <servlet-name>dispatcher</servlet-name>
  <url-pattern>/*</url-pattern>
</servlet-mapping>
```

- Example: Configure DispatcherServlet for Spring MVC

3. Spring Configuration

```
<context:component-scan base-package="com.in28minutes" />

<bean
    class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix">
        <value>/WEB-INF/views/</value>
    </property>
    <property name="suffix">
        <value>.jsp</value>
    </property>
</bean>
```

- Define your Spring Configuration
 - Component Scan
 - View Resolver

4. Non-Functional Requirements

```
<plugin>
    <groupId>org.apache.tomcat.maven</groupId>
    <artifactId>tomcat7-maven-plugin</artifactId>
    <version>2.2</version>
    <configuration>
        <path>/</path>
        <contextReloadable>true</contextReloadable>
    </configuration>
</plugin>

<dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
    <version>1.2.17</version>
</dependency>
```

- Logging
- Error Handling
- Monitoring

- Setting up Spring Projects before Spring Boot was NOT easy!
 - 1: Dependency Management (pom.xml)
 - 2: Define Web App Configuration (web.xml)
 - 3: Manage Spring Beans (context.xml)
 - 4: Implement Non-Functional Requirements (NFRs)
- AND repeat this for every new project!
- Typically takes a few days to setup for each project (and countless hours to maintain)