Spring Boot

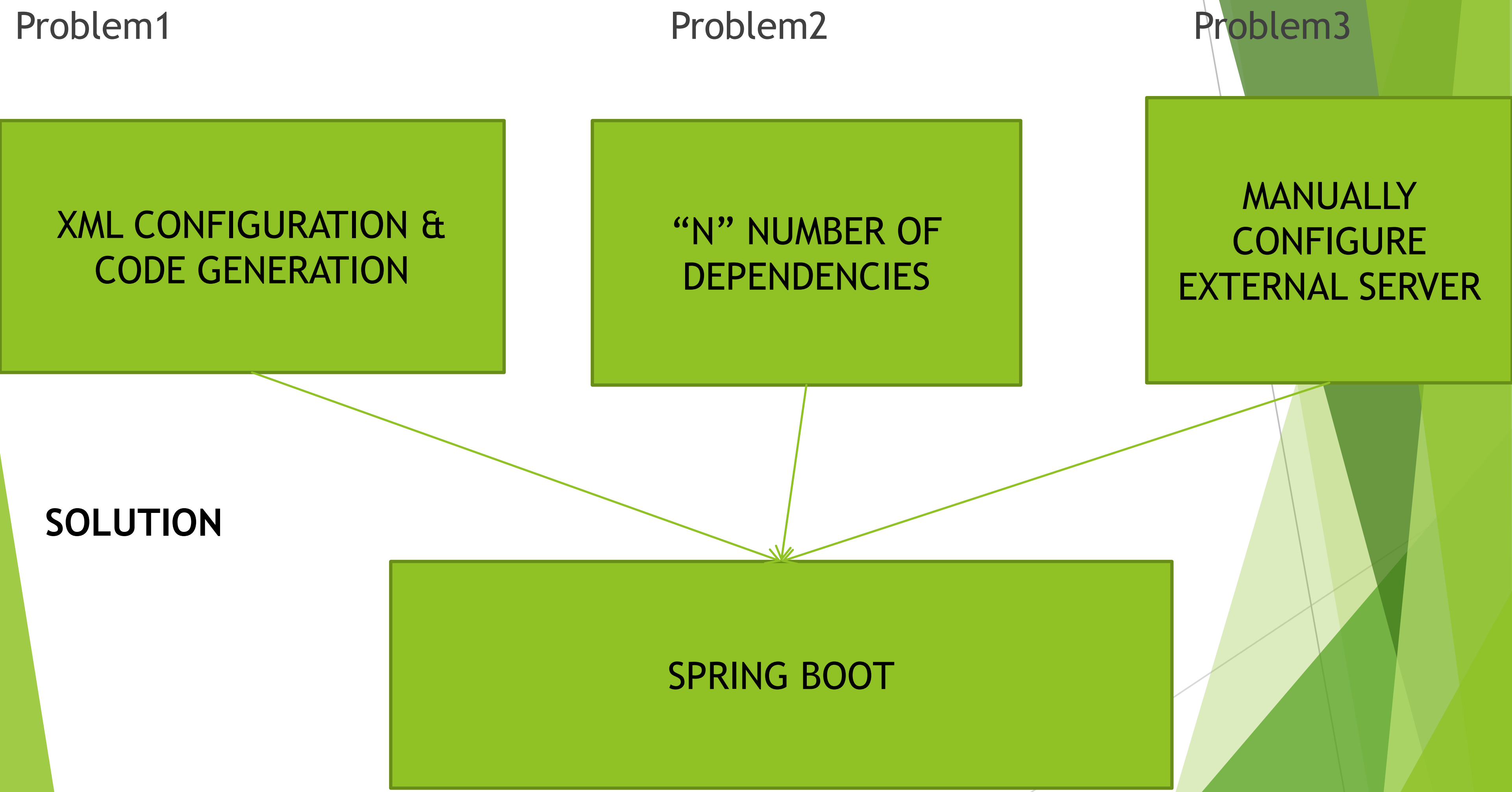
- What is Spring boot
- Key features & Components
- How spring boot made easy?
- IDE for spring boot
- Ways to start a spring boot project

Lets Go...!!

Key features

- Create stand-alone Spring applications
- Embed Tomcat, Jetty or Undertow directly (no need to deploy WAR files)
- Provide opinionated 'starter' POMs to simplify your Maven configuration
- Automatically configure Spring whenever possible
- Provide production-ready features such as metrics, health checks and externalized configuration
- Absolutely **no code generation** and **no requirement for XML configuration**

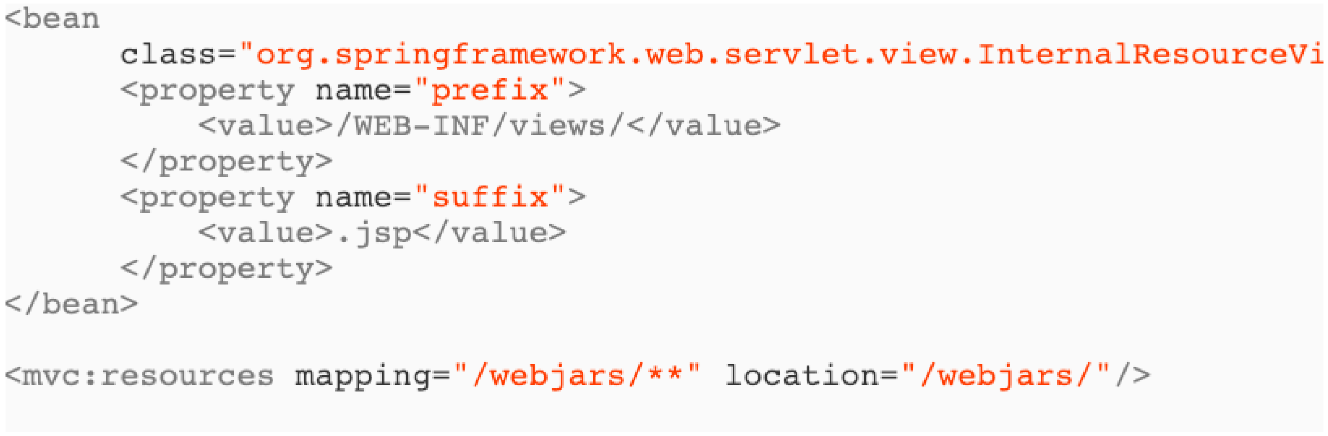
Why do we need Spring Boot?


Problem1
Problem2
Problem3
XML CONFIGURATION & CODE GENERATION
"N" NUMBER OF DEPENDENCIES
MANUALLY CONFIGURE EXTERNAL SERVER
SOLUTION
SPRING BOOT

Problem #1 : Spring Boot Auto Configuration :Say no to XML Configurations

- Spring based applications have a lot of configuration.
- When we use Spring MVC, we need to configure component scan, dispatcher servlet, a view resolver, web jars(for delivering static content) among other things.

Spring Configuration XML File

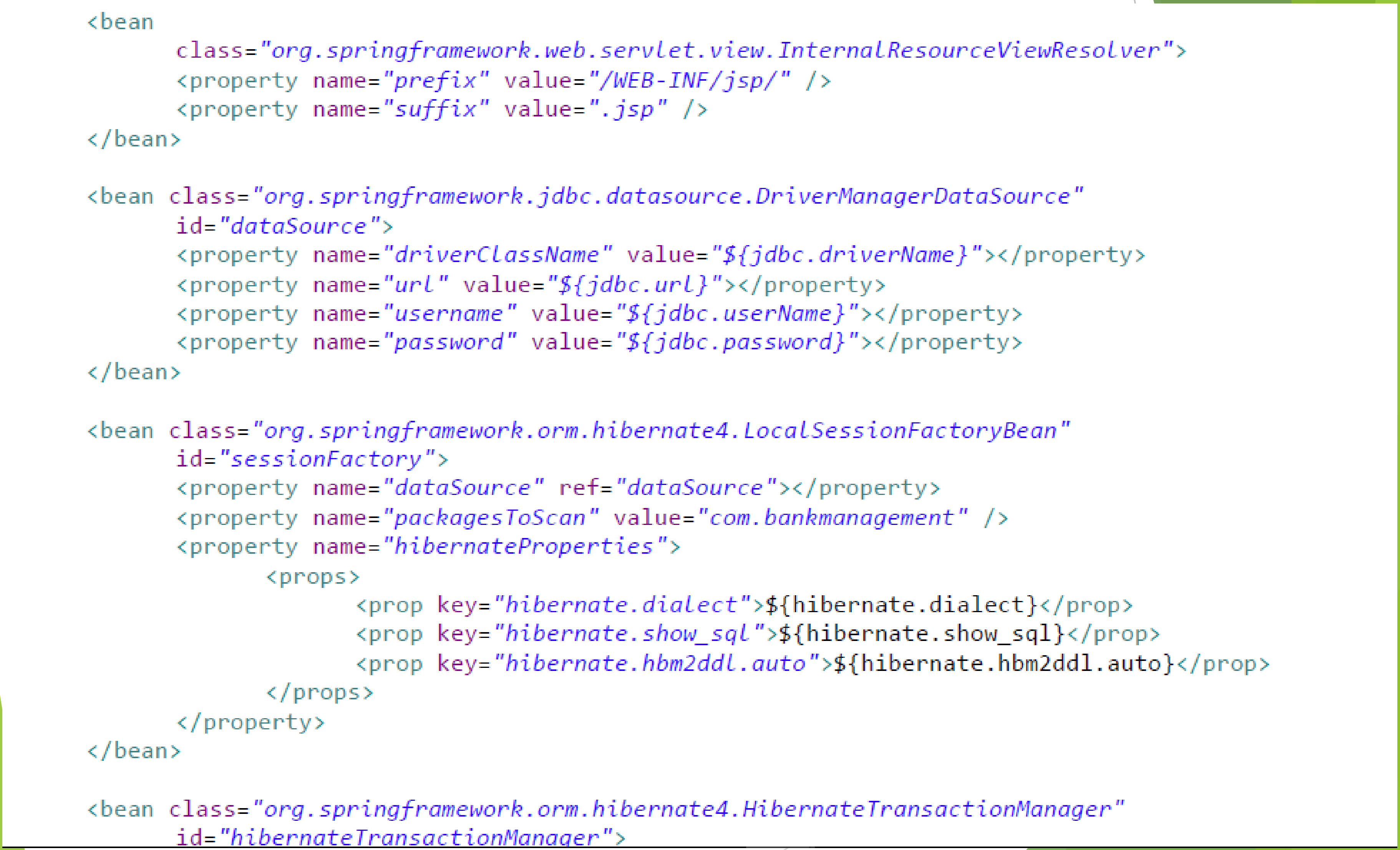


```
<bean
      class="org.springframework.web.servlet.view.InternalResourceVi
      <property name="prefix">
          <value>/WEB-INF/views/</value>
      </property>
      <property name="suffix">
          <value>.jsp</value>
      </property>
</bean>

<mvc:resources mapping="/webjars/**" location="/webjars/"/>
```

Typical configuration of a dispatcher servlet in a web application.

```
<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>
```

Data source, hibernate, transaction manager Configuration in Spring.xml file



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

<bean class="org.springframework.jdbc.datasource.DriverManagerDataSource"
      id="dataSource">
      <property name="driverClassName" value="${jdbc.driverName}"></property>
      <property name="url" value="${jdbc.url}"></property>
      <property name="username" value="${jdbc.userName}"></property>
      <property name="password" value="${jdbc.password}"></property>
</bean>

<bean class="org.springframework.orm.hibernate4.LocalSessionFactoryBean"
      id="sessionFactory">
      <property name="dataSource" ref="dataSource"></property>
      <property name="packagesToScan" value="com.bankmanagement" />
      <property name="hibernateProperties">
            <props>
                  <prop key="hibernate.dialect">${hibernate.dialect}</prop>
                  <prop key="hibernate.show_sql">${hibernate.show_sql}</prop>
                  <prop key="hibernate.hbm2ddl.auto">${hibernate.hbm2ddl.auto}</prop>
            </props>
      </property>
</bean>

<bean class="org.springframework.orm.hibernate4.HibernateTransactionManager"
      id="hibernateTransactionManager">
```

Starting of spring Boot Application

@SpringBootApplication =

@Configuration + @EnableAutoConfiguration + @ComponentScan

@Configuration- indicates that the class can be used by the Spring IoC container as a source of bean definitions .

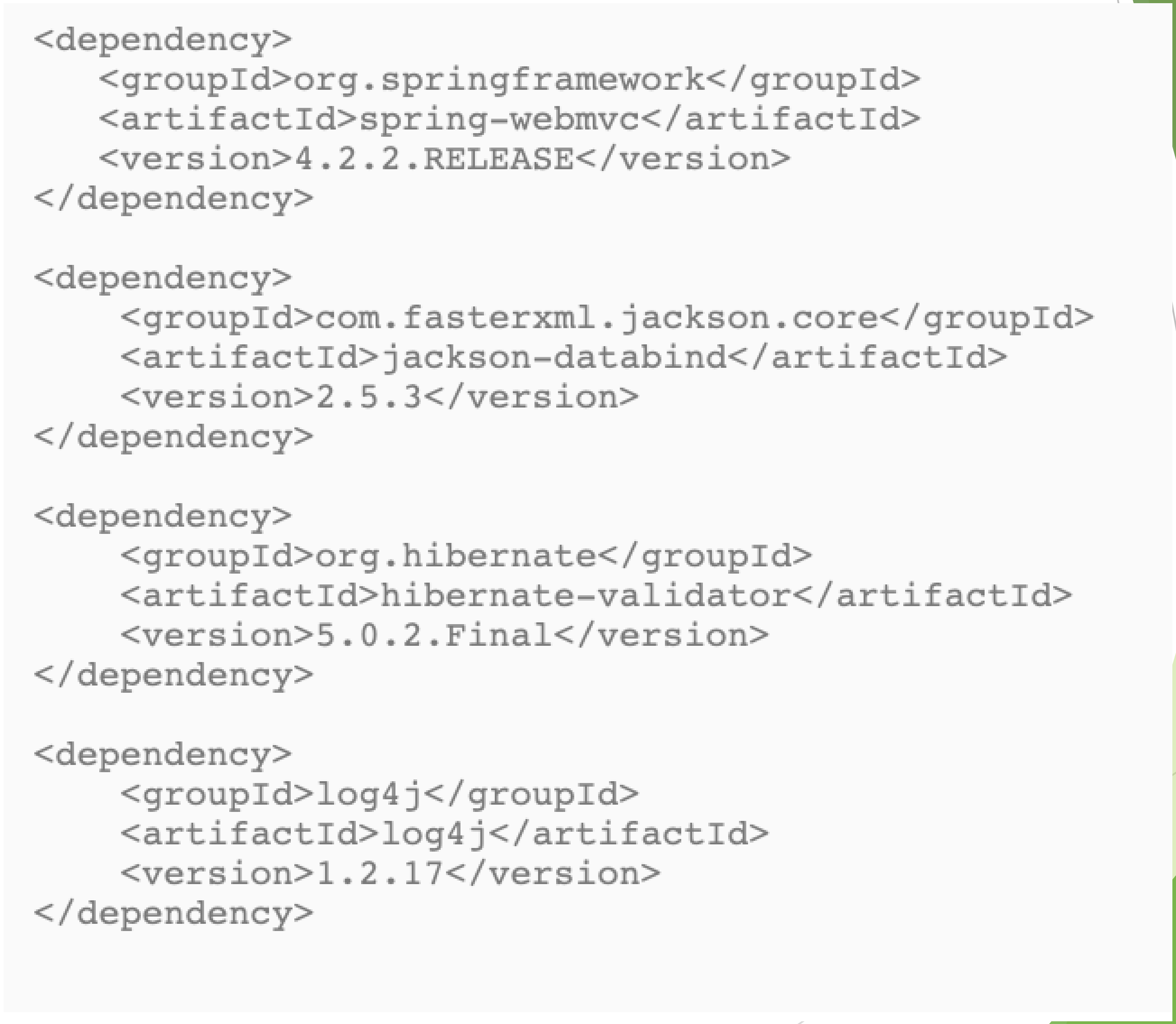
Its equal to <beans>………….</beans>

@EnableAutoConfiguration- will automatically do the spring configurations. it will create, register and load the Spring configuration beans required by the applications from the classes available in the class path.

@ComponentScan-<context:component-scan base-package="com.cognizant.com"

"tells Spring to look for other components, configurations, and services in the specified package"

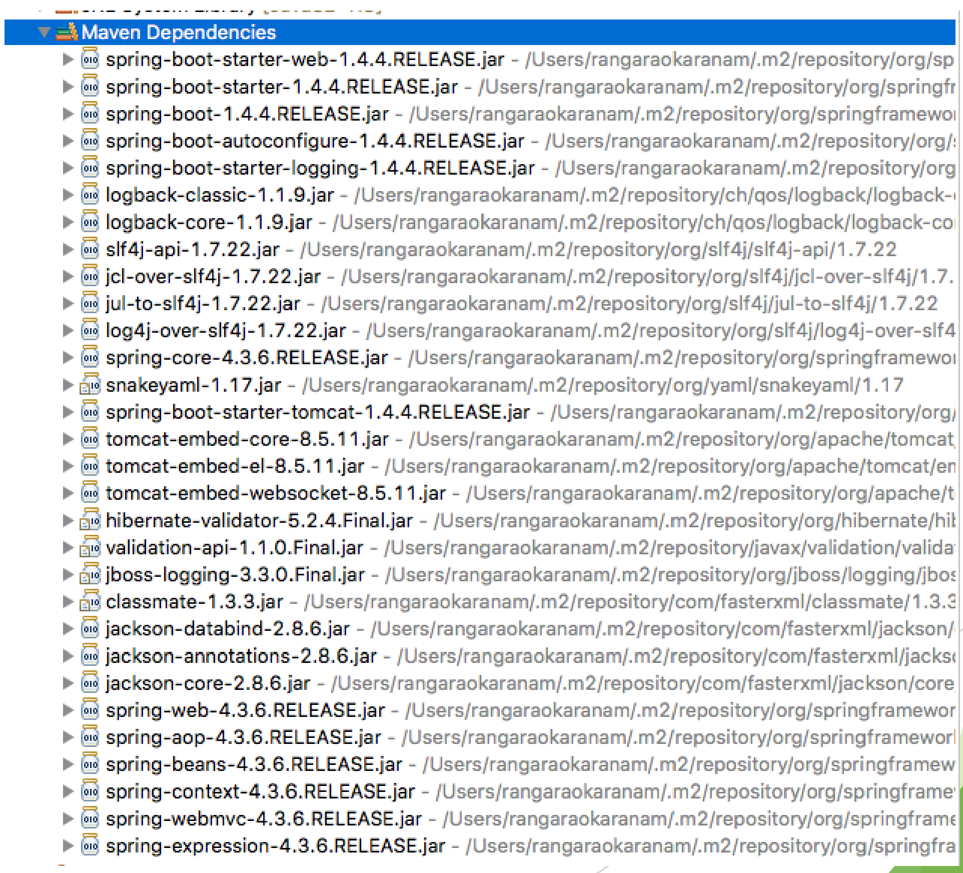
Problem #2 : Spring Boot Starter Projects



```
<dependency>
   <groupId>org.springframework</groupId>
   <artifactId>spring-webmvc</artifactId>
   <version>4.2.2.RELEASE</version>
</dependency>

<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>2.5.3</version>
</dependency>

<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-validator</artifactId>
    <version>5.0.2.Final</version>
</dependency>

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

Dependency for spring boot starter web

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



```
▼ Maven Dependencies
  ▶ spring-boot-starter-web-1.4.4.RELEASE.jar - /Users/rangaraokaranam/.m2/repository/org/sp
  ▶ spring-boot-starter-1.4.4.RELEASE.jar - /Users/rangaraokaranam/.m2/repository/org/springfr
  ▶ spring-boot-1.4.4.RELEASE.jar - /Users/rangaraokaranam/.m2/repository/org/springframewo
  ▶ spring-boot-autoconfigure-1.4.4.RELEASE.jar - /Users/rangaraokaranam/.m2/repository/org/
  ▶ spring-boot-starter-logging-1.4.4.RELEASE.jar - /Users/rangaraokaranam/.m2/repository/org
  ▶ logback-classic-1.1.9.jar - /Users/rangaraokaranam/.m2/repository/ch/qos/logback/logback-
  ▶ logback-core-1.1.9.jar - /Users/rangaraokaranam/.m2/repository/ch/qos/logback/logback-co
  ▶ slf4j-api-1.7.22.jar - /Users/rangaraokaranam/.m2/repository/org/slf4j/slf4j-api/1.7.22
  ▶ jcl-over-slf4j-1.7.22.jar - /Users/rangaraokaranam/.m2/repository/org/slf4j/jcl-over-slf4j/1.7.
  ▶ jul-to-slf4j-1.7.22.jar - /Users/rangaraokaranam/.m2/repository/org/slf4j/jul-to-slf4j/1.7.22
  ▶ log4j-over-slf4j-1.7.22.jar - /Users/rangaraokaranam/.m2/repository/org/slf4j/log4j-over-slf4
  ▶ spring-core-4.3.6.RELEASE.jar - /Users/rangaraokaranam/.m2/repository/org/springframewo
  ▶ snakeyaml-1.17.jar - /Users/rangaraokaranam/.m2/repository/org/yaml/snakeyaml/1.17
  ▶ spring-boot-starter-tomcat-1.4.4.RELEASE.jar - /Users/rangaraokaranam/.m2/repository/org/
  ▶ tomcat-embed-core-8.5.11.jar - /Users/rangaraokaranam/.m2/repository/org/apache/tomcat
  ▶ tomcat-embed-el-8.5.11.jar - /Users/rangaraokaranam/.m2/repository/org/apache/tomcat/en
  ▶ tomcat-embed-websocket-8.5.11.jar - /Users/rangaraokaranam/.m2/repository/org/apache/t
  ▶ hibernate-validator-5.2.4.Final.jar - /Users/rangaraokaranam/.m2/repository/org/hibernate/hi
  ▶ validation-api-1.1.0.Final.jar - /Users/rangaraokaranam/.m2/repository/javax/validation/valida
  ▶ jboss-logging-3.3.0.Final.jar - /Users/rangaraokaranam/.m2/repository/org/jboss/logging/jbo
  ▶ classmate-1.3.3.jar - /Users/rangaraokaranam/.m2/repository/com/fasterxml/classmate/1.3.3
  ▶ jackson-databind-2.8.6.jar - /Users/rangaraokaranam/.m2/repository/com/fasterxml/jackson/
  ▶ jackson-annotations-2.8.6.jar - /Users/rangaraokaranam/.m2/repository/com/fasterxml/jacks
  ▶ jackson-core-2.8.6.jar - /Users/rangaraokaranam/.m2/repository/com/fasterxml/jackson/core
  ▶ spring-web-4.3.6.RELEASE.jar - /Users/rangaraokaranam/.m2/repository/org/springframewor
  ▶ spring-aop-4.3.6.RELEASE.jar - /Users/rangaraokaranam/.m2/repository/org/springframewor
  ▶ spring-beans-4.3.6.RELEASE.jar - /Users/rangaraokaranam/.m2/repository/org/springframew
  ▶ spring-context-4.3.6.RELEASE.jar - /Users/rangaraokaranam/.m2/repository/org/springframe
  ▶ spring-webmvc-4.3.6.RELEASE.jar - /Users/rangaraokaranam/.m2/repository/org/springframe
  ▶ spring-expression-4.3.6.RELEASE.jar - /Users/rangaraokaranam/.m2/repository/org/springfra
```

Minimum dependencies

- Dependencies can be classified into:
- Spring - core, beans, context, aop
- Web MVC - (Spring MVC)
- Jackson - for JSON Binding
- Validation - Hibernate Validator, Validation API
- Embedded Servlet Container - Tomcat
- Logging - logback, slf4j
- Any typical web application would use all these dependencies. Spring Boot Starter Web comes pre packaged with these. As a developer, I would not need to worry about either these dependencies or their compatible versions.

Spring boot Starter project options

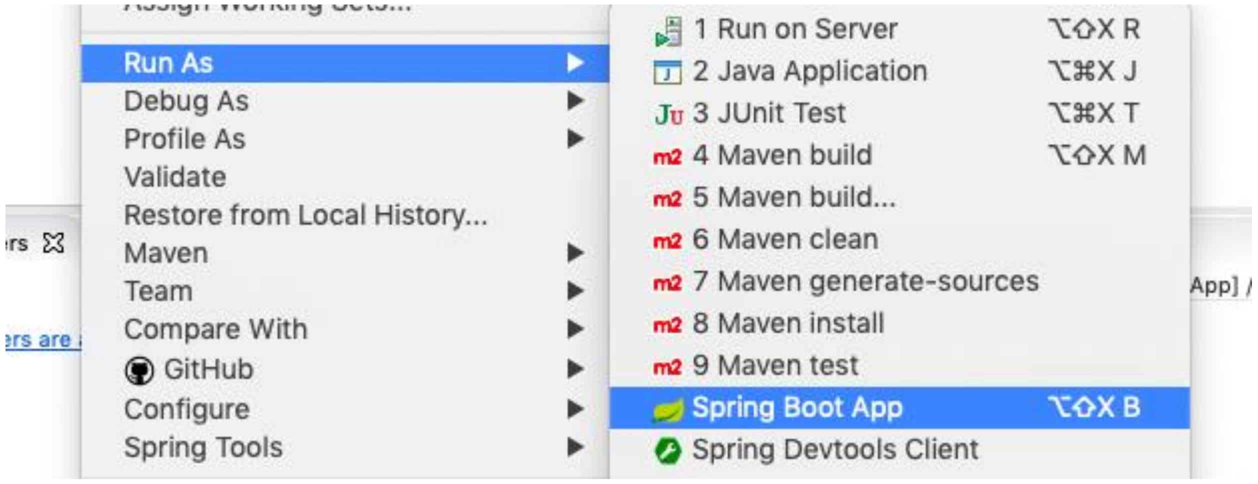
S.NO	STARTERS	DESCRIPTION
1	spring-boot-starter-data-jpa	Starter for using Spring Data JPA with Hibernate
2	spring-boot-starter-activemq	Starter for JMS messaging using Apache ActiveMQ
3	spring-boot-starter	Core starter, including auto-configuration support, logging and YAML
4	spring-boot-starter-integration	Starter for using Spring Integration
5	spring-boot-starter-actuator	provides production ready features to help you monitor and manage your application
6	spring-boot-starter-security	Starter for using Spring Security
7	spring-boot-starter-test	Starter for testing Spring Boot applications with libraries including JUnit, Hamcrest and Mockito

Reference URL: https://docs.spring.io/spring-boot/docs/current/reference/html/using-boot-build-systems.html

Problem3-Need of external Server

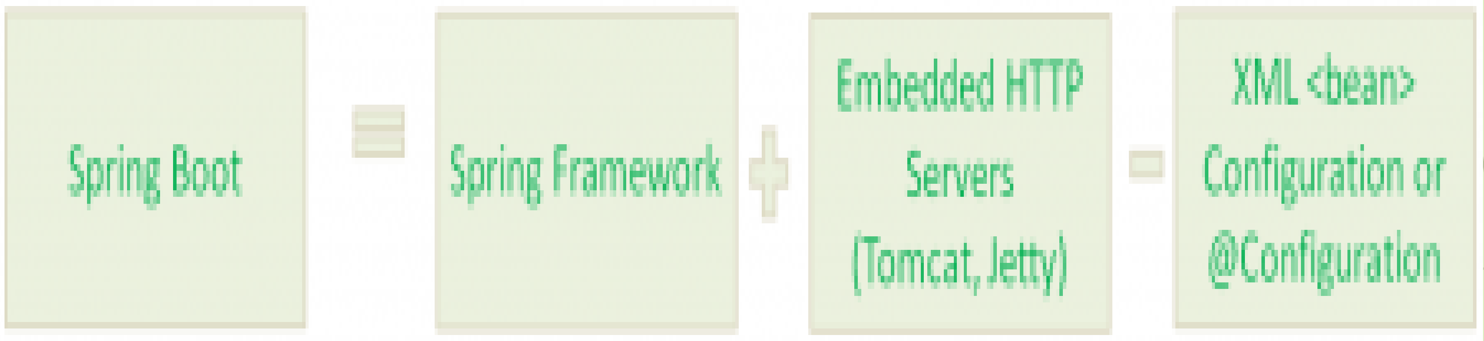
- **What is an Embedded Server?**
- Think about what you would need to be able to deploy your application (typically) on a virtual machine.
- Step 1 : Install Java
- Step 2 : Install the Web/Application Server (Tomcat/Websphere/Weblogic etc)
- Step 3 : Deploy the application war

Embedded Tomcat,Jetty (no need to deploy war files)



To be simple

- Spring Boot is a spring framework module which provides Rapid Application Development feature to the Spring framework.


Spring Boot
=
Spring Framework
+
Embedded HTTP Servers (Tomcat, Jetty)
-
XML <bean> Configuration or @Configuration

Why spring boot?

- To ease the Java-based applications Development, Unit Test and Integration Test Process.
- To reduce Development, Unit Test and Integration Test time by providing some defaults.
- To increase Productivity.

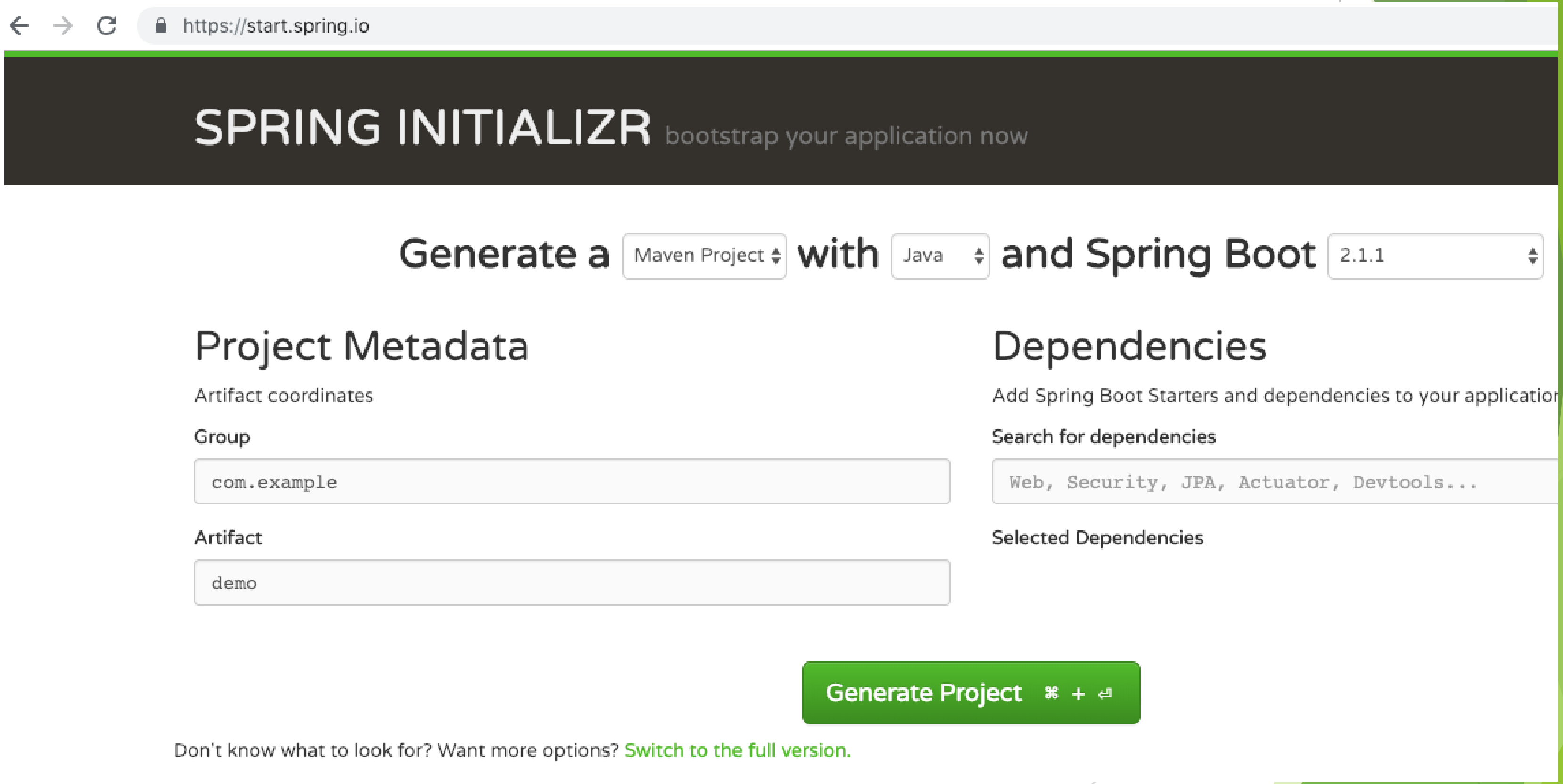
IDE for Spring Boot

- Spring Tool Suite(STS)

url- https://spring.io/tools3/sts/all

- Intelli J
- NetBeans

Ways to create a new spring boot application


https://start.spring.io

SPRING INITIALIZR bootstrap your application now

Generate a Maven Project with Java and Spring Boot 2.1.1

Project Metadata

Artifact coordinates

Group: com.example

Artifact: demo

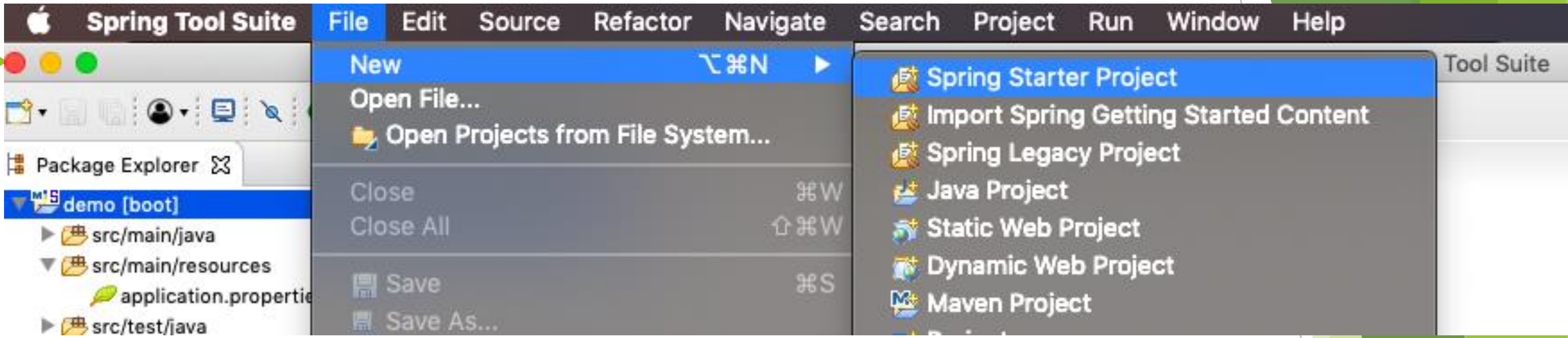
Dependencies

Add Spring Boot Starters and dependencies to your applicatio

Search for dependencies: Web, Security, JPA, Actuator, Devtools...

Selected Dependencies

Don't know what to look for? Want more options? Switch to the full version.



- Create Maven Project→Add Spring Based Dependency in pom.xml