

[Home](#) > [member-paid](#) > [Spring IO tutorial in 6 steps](#)

# Spring IO tutorial in 6 steps

Posted on [November 22, 2015](#) by [Arulkumaran Kumaraswamipillai](#)

**Spring IO** Platform is all about “list of dependencies and their **versions** that work well together”. It is implemented as a Maven POM file via Maven Bill-of-Materials dependency that you can import into your projects to set the versions for dependencies. **Spring Boot** is built on top of the “**Spring IO**” platform. Spring Boot makes it easy to create **stand-alone, production-grade** Spring applications that you can just run as covered in “[Simple Spring Boot Tutorial in 8 steps](#)” and [Simple Spring Boot Restful Web Service Tutorial](#).

**Step 1:** Create a simple maven project using the maven “archetype” plugin.

```
1 mvn archetype:generate -DgroupId=com.mytutorial -
```

**Step 2:** import it into eclipse by choosing **File** → **Import** → **Maven** → Existing Maven projects and select the folder “simple-spring-io” you had just created with Maven.

**Step 3:** Modify the **pom.xml** file to as shown below. As you can see after adding “platform-pom”, you don’t need to

## 600+ Full Stack Java/JEE Interview Q&As ♥Free ♦FAQs

[open all](#) | [close all](#)

- ✚ Ice Breaker Interview
- ✚ Core Java Interview C
- ✚ JEE Interview Q&A (3
- ✚ Pressed for time? Jav
- ✚ SQL, XML, UML, JSC
- ✚ Hadoop & BigData Int
- ✚ Java Architecture Inte
- ✚ Scala Interview Q&As
- ✚ Spring, Hibernate, & I
- ✚ Spring (18)
- ✚ Spring boot (4)
- ✚ Spring IO (1)
- ✚ Spring IO tuto
- ✚ Spring JavaConl
- ✚ 01: ♥♦ 13 Spring
- ✚ 01b: ♦ 13 Spring
- ✚ 02: ► Spring DI
- ✚ 03: ♥♦ Spring DI
- ✚ 04 ♦ 17 Spring b
- ✚ 05: ♦ 9 Spring B
- ✚ 06: ♥ Debugging
- ✚ 07: Debugging S

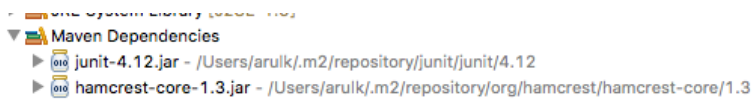
specify the version XML tag for "junit".

```

1
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3   xsi:schemaLocation="http://maven.apache.org/
4     <modelVersion>4.0.0</modelVersion>
5     <groupId>com.mytutorial</groupId>
6     <artifactId>simple-io</artifactId>
7     <packaging>jar</packaging>
8     <version>1.0-SNAPSHOT</version>
9     <name>simple-io</name>
10    <url>http://maven.apache.org</url>
11
12    <dependencyManagement>
13      <dependencies>
14        <dependency>
15          <groupId>io.spring.platform</gro
16          <artifactId>platform-bom</artifa
17          <version>2.0.0.RELEASE</version>
18          <type>pom</type>
19          <scope>import</scope>
20        </dependency>
21      </dependencies>
22    </dependencyManagement>
23
24    <dependencies>
25      <dependency>
26        <groupId>junit</groupId>
27        <artifactId>junit</artifactId>
28        <scope>test</scope>
29      </dependency>
30    </dependencies>
31  </project>
32
33

```

The following dependencies will be added when you execute:  
"mvn clean package"



```

Maven Dependencies
├── junit-4.12.jar - /Users/arulk/.m2/repository/junit/junit/4.12
└── hamcrest-core-1.3.jar - /Users/arulk/.m2/repository/org/hamcrest/hamcrest-core/1.3

```

Spring IO BOM adds following dependency jars.

**Step 4:** Add a **Spring dependency** to the **pom.xml** file as shown below

```

1
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3   xsi:schemaLocation="http://maven.apache.org/
4     <modelVersion>4.0.0</modelVersion>
5     <groupId>com.mytutorial</groupId>
6     <artifactId>simple-io</artifactId>

```

Spring loading p

- [+ Hibernate \(13\)](#)
- [+ AngularJS \(2\)](#)
- [+ Git & SVN \(6\)](#)
- [+ JMeter \(2\)](#)
- [+ JSF \(2\)](#)
- [+ Maven \(3\)](#)
- [+ Testing & Profiling/Sa](#)
- [+ Other Interview Q&A 1](#)
- [+ Free Java Interview](#)

## 16 Technical Key Areas

[open all](#) | [close all](#)

- [+ Best Practice \(6\)](#)
- [+ Coding \(26\)](#)
- [+ Concurrency \(6\)](#)
- [+ Design Concepts \(7\)](#)
- [+ Design Patterns \(11\)](#)
- [+ Exception Handling \(3\)](#)
- [+ Java Debugging \(21\)](#)
- [+ Judging Experience I](#)
- [+ Low Latency \(7\)](#)
- [+ Memory Management](#)
- [+ Performance \(13\)](#)
- [+ QoS \(8\)](#)
- [+ Scalability \(4\)](#)
- [+ SDLC \(6\)](#)
- [+ Security \(13\)](#)
- [+ Transaction Managen](#)

## 80+ step by step Java Tutorials

[open all](#) | [close all](#)

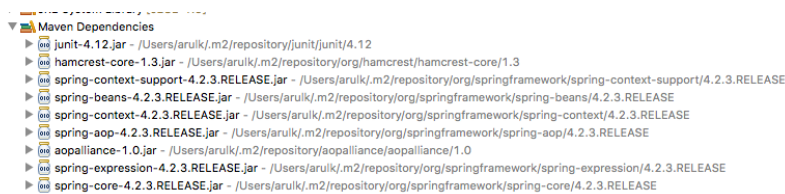
- [+ Setting up Tutorial \(6\)](#)
- [+ Tutorial - Diagnosis \(2](#)

```

7      <packaging>jar</packaging>
8      <version>1.0-SNAPSHOT</version>
9      <name>simple-io</name>
10     <url>http://maven.apache.org</url>
11
12     <dependencyManagement>
13         <dependencies>
14             <dependency>
15                 <groupId>io.spring.platform</gro
16                 <artifactId>platform-bom</artifa
17                 <version>2.0.0.RELEASE</version>
18                 <type>pom</type>
19                 <scope>import</scope>
20             </dependency>
21         </dependencies>
22     </dependencyManagement>
23
24
25     <dependencies>
26         <dependency>
27             <groupId>junit</groupId>
28             <artifactId>junit</artifactId>
29             <scope>test</scope>
30         </dependency>
31
32         <dependency>
33             <groupId>org.springframework</groupI
34             <artifactId>spring-context-support</
35         </dependency>
36     </dependencies>
37 </project>
38

```

The following dependencies will be added when you execute:  
“mvn clean package”



```

Maven Dependencies
├─ junit-4.12.jar - /Users/arulkj.m2/repository/junit/junit/4.12
├─ hamcrest-core-1.3.jar - /Users/arulkj.m2/repository/org/hamcrest/hamcrest-core/1.3
├─ spring-context-support-4.2.3.RELEASE.jar - /Users/arulkj.m2/repository/org/springframework/spring-context-support/4.2.3.RELEASE
├─ spring-beans-4.2.3.RELEASE.jar - /Users/arulkj.m2/repository/org/springframework/spring-beans/4.2.3.RELEASE
├─ spring-context-4.2.3.RELEASE.jar - /Users/arulkj.m2/repository/org/springframework/spring-context/4.2.3.RELEASE
├─ spring-aop-4.2.3.RELEASE.jar - /Users/arulkj.m2/repository/org/springframework/spring-aop/4.2.3.RELEASE
├─ aopalliance-1.0.jar - /Users/arulkj.m2/repository/aopalliance/aopalliance/1.0
├─ spring-expression-4.2.3.RELEASE.jar - /Users/arulkj.m2/repository/org/springframework/spring-expression/4.2.3.RELEASE
└─ spring-core-4.2.3.RELEASE.jar - /Users/arulkj.m2/repository/org/springframework/spring-core/4.2.3.RELEASE

```

Spring IO BOM adding Spring dependencies

**Step 5:** Now to bring in the “**Spring Data JPA**” dependencies, you can add the following without any versions.

```

1
2 <project xmlns="http://maven.apache.org/POM/4.0.
3     xsi:schemaLocation="http://maven.apache.org/
4     <modelVersion>4.0.0</modelVersion>
5     <groupId>com.mytutorial</groupId>
6     <artifactId>simple-io</artifactId>
7     <packaging>jar</packaging>

```

- [Akka Tutorial \(9\)](#)
- [Core Java Tutorials \(2\)](#)
- [Hadoop & Spark Tuto](#)
- [JEE Tutorials \(19\)](#)
- [Scala Tutorials \(1\)](#)
- [Spring & Hibernate Ti](#)
- [Hibernate Tutorial \(](#)
- [Spring Basics Tuto](#)
- [Spring batch tutoria](#)
- [JPA-Hibernate-Spri](#)
- [Spring Boot Tutoria](#)
- [Spring IO Tutorial \(](#)
- [Spring IO tutorial](#)
- [Spring ORM Tutoria](#)
- [Spring other tutoria](#)
- [Spring OXM Tutoria](#)
- [Tools Tutorials \(19\)](#)
- [Other Tutorials \(45\)](#)

## 100+ Java pre-interview coding tests

[open all](#) | [close all](#)

- [Can you write code? \(](#)
- [♦ Complete the given](#)
- [Converting from A to I](#)
- [Designing your classe](#)
- [Java Data Structures](#)
- [Passing the unit tests](#)
- [What is wrong with th](#)
- [Writing Code Home A](#)
- [Written Test Core Jav](#)
- [Written Test JEE \(1\)](#)

## How good are your .....?

[open all](#) | [close all](#)

```

8      <version>1.0-SNAPSHOT</version>
9      <name>simple-io</name>
10     <url>http://maven.apache.org</url>
11
12     <dependencyManagement>
13         <dependencies>
14             <dependency>
15                 <groupId>io.spring.platform</gro
16                 <artifactId>platform-bom</artifa
17                 <version>2.0.0.RELEASE</version>
18                 <type>pom</type>
19                 <scope>import</scope>
20             </dependency>
21         </dependencies>
22     </dependencyManagement>
23
24     <dependencies>
25         <dependency>
26             <groupId>junit</groupId>
27             <artifactId>junit</artifactId>
28             <scope>test</scope>
29         </dependency>
30
31         <dependency>
32             <groupId>org.springframework</groupI
33             <artifactId>spring-context-support</
34         </dependency>
35
36         <!-- Database (H2) -->
37         <dependency>
38             <groupId>com.h2database</groupId>
39             <artifactId>h2</artifactId>
40         </dependency>
41
42         <!-- DataSource (HikariCP) -->
43         <dependency>
44             <groupId>com.zaxxer</groupId>
45             <artifactId>HikariCP</artifactId>
46         </dependency>
47
48         <!-- JPA Provider (Hibernate) -->
49         <dependency>
50             <groupId>org.hibernate</groupId>
51             <artifactId>hibernate-entitymanager<
52         </dependency>
53
54         <!-- Spring Data JPA -->
55         <dependency>
56             <groupId>org.springframework.data</g
57             <artifactId>spring-data-jpa</artifac
58         </dependency>
59     </dependencies>
60 </project>
61
62

```

The following dependencies will be added when you execute:  
“mvn clean package”

**Maven Dependencies**

- ▶ junit-4.12.jar - /Users/arulkj.m2/repository/junit/junit/4.12
- ▶ hamcrest-core-1.3.jar - /Users/arulkj.m2/repository/org/hamcrest/hamcrest-core/1.3
- ▶ spring-context-support-4.2.3.RELEASE.jar - /Users/arulkj.m2/repository/org/springframework/spring-context-support/4.2.3.RELEASE
- ▶ spring-beans-4.2.3.RELEASE.jar - /Users/arulkj.m2/repository/org/springframework/spring-beans/4.2.3.RELEASE
- ▶ spring-context-4.2.3.RELEASE.jar - /Users/arulkj.m2/repository/org/springframework/spring-context/4.2.3.RELEASE
- ▶ spring-expression-4.2.3.RELEASE.jar - /Users/arulkj.m2/repository/org/springframework/spring-expression/4.2.3.RELEASE
- ▶ spring-core-4.2.3.RELEASE.jar - /Users/arulkj.m2/repository/org/springframework/spring-core/4.2.3.RELEASE
- ▶ h2-1.4.190.jar - /Users/arulkj.m2/repository/com/h2database/h2/1.4.190
- ▶ HikariCP-2.4.2.jar - /Users/arulkj.m2/repository/com/zaxxer/HikariCP/2.4.2
- ▶ slf4j-api-1.7.13.jar - /Users/arulkj.m2/repository/org/slf4j/slf4j-api/1.7.13
- ▶ hibernate-entitymanager-4.3.11.Final.jar - /Users/arulkj.m2/repository/org/hibernate/hibernate-entitymanager/4.3.11.Final
- ▶ jboss-logging-3.3.0.Final.jar - /Users/arulkj.m2/repository/org/jboss/logging/jboss-logging/3.3.0.Final
- ▶ jboss-logging-annotations-1.2.0.Beta1.jar - /Users/arulkj.m2/repository/org/jboss/logging/jboss-logging-annotations/1.2.0.Beta1
- ▶ hibernate-core-4.3.11.Final.jar - /Users/arulkj.m2/repository/org/hibernate/hibernate-core/4.3.11.Final
- ▶ antir-2.7.7.jar - /Users/arulkj.m2/repository/antir/antir/2.7.7
- ▶ jandex-1.1.0.Final.jar - /Users/arulkj.m2/repository/org/jboss/jandex/1.1.0.Final
- ▶ dom4j-1.6.1.jar - /Users/arulkj.m2/repository/dom4j/dom4j/1.6.1
- ▶ xml-apis-1.0.b2.jar - /Users/arulkj.m2/repository/xml-apis/xml-apis/1.0.b2
- ▶ hibernate-commons-annotations-4.0.5.Final.jar - /Users/arulkj.m2/repository/org/hibernate/common/hibernate-commons-annotations/4.0.5.Final
- ▶ hibernate-jpa-2.1-api-1.0.0.Final.jar - /Users/arulkj.m2/repository/org/hibernate/javax/persistence/hibernate-jpa-2.1-api/1.0.0.Final
- ▶ jboss-transaction-api-1.2\_spec-1.0.0.Final.jar - /Users/arulkj.m2/repository/org/jboss/spec/javax/transaction/jboss-transaction-api-1.2\_spec/1.0.0.Final
- ▶ javassist-3.18.1-GA.jar - /Users/arulkj.m2/repository/org/javassist/javassist/3.18.1-GA
- ▶ spring-data-jpa-1.9.1.RELEASE.jar - /Users/arulkj.m2/repository/org/springframework/data/spring-data-jpa/1.9.1.RELEASE
- ▶ spring-data-commons-1.11.1.RELEASE.jar - /Users/arulkj.m2/repository/org/springframework/data/spring-data-commons/1.11.1.RELEASE
- ▶ spring-orm-4.2.3.RELEASE.jar - /Users/arulkj.m2/repository/org/springframework/spring-orm/4.2.3.RELEASE
- ▶ spring-jdbc-4.2.3.RELEASE.jar - /Users/arulkj.m2/repository/org/springframework/spring-jdbc/4.2.3.RELEASE
- ▶ spring-aop-4.2.3.RELEASE.jar - /Users/arulkj.m2/repository/org/springframework/spring-aop/4.2.3.RELEASE
- ▶ aopalliance-1.0.jar - /Users/arulkj.m2/repository/aopalliance/aopalliance/1.0
- ▶ spring-tx-4.2.3.RELEASE.jar - /Users/arulkj.m2/repository/org/springframework/spring-tx/4.2.3.RELEASE
- ▶ aspectjrt-1.8.7.jar - /Users/arulkj.m2/repository/org/aspectj/aspectjrt/1.8.7
- ▶ jci-over-slf4j-1.7.13.jar - /Users/arulkj.m2/repository/org/slf4j/jci-over-slf4j/1.7.13

Spring Data JPA dependencies and versions via spring io BOM

## Step 6: To bring in the web dependencies

```

1
2      <!-- Spring Web -->
3      <dependency>
4          <groupId>org.springframework</groupId>
5          <artifactId>spring-webmvc</artifactId>
6      </dependency>
7

```

The following web dependencies will be included when you execute: “mvn clean package”

```

▶ 010 spring-webmvc-4.2.3.RELEASE.jar - /Use
▶ 010 spring-expression-4.2.3.RELEASE.jar - /l
▶ 010 spring-web-4.2.3.RELEASE.jar - /Users/a

```

Spring web dependencies added with Spring IO BOM

## Popular Member Posts

◆ 11 Spring boot interview questions & answers

904 views

◆ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

816 views

## 001A: ♦ 7+ Java integration styles & patterns

### interview questions & answers

427 views

## 18 Java scenarios based interview Questions and Answers

407 views

## ♦ 7 Java debugging interview questions & answers

322 views

## 01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

310 views

## 01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

302 views

## ♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

301 views

## ♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

251 views

## 001B: ♦ Java architecture & design concepts interview questions & answers

209 views

Bio

Latest Posts



### Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via [Amazon.com](https://www.amazon.com) in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription



based site.**945+** paid members. [join my LinkedIn Group](#). [Reviews](#)



#### About [Arulkumaran Kumaraswamipillai](#)

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via [Amazon.com](#) in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. [join my LinkedIn Group](#). [Reviews](#)

◀ 02: Simple Spring Boot Restful Web Service Tutorial

Apache POI reading a xlsx file tutorial ▶

**Posted in** member-paid, Spring IO, Spring IO Tutorial

## Empowers you to open more doors, and fast-track

### Technical Know Hows

☀ [Java generics in no time](#) ☀ [Top 6 tips to transforming your thinking from OOP to FP](#) ☀ [How does a HashMap internally work? What is a hashing function?](#)  
 ☀ [10+ Java String class interview Q&As](#) ☀ [Java auto un/boxing benefits & caveats](#) ☀ [Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect](#)

### Non-Technical Know Hows

☀ [6 Aspects that can motivate you to fast-track your career & go places](#) ☀ [Are you reinventing yourself as a Java developer?](#) ☀ [8 tips to safeguard your Java career against offshoring](#) ☀ [My top 5 career mistakes](#)

## Prepare to succeed

☀ [Turn readers of your Java CV go from “Blah blah” to “Wow”?](#) ☀ [How to prepare for Java job interviews?](#) ☀ [16 Technical Key Areas](#) ☀ [How to choose from multiple Java job offers?](#)

## © Disclaimer

The contents in this Java-Success are copy righted. The author has the right to correct or enhance the current content without any prior notice.

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.