

Industrial strength Java/JEE Career Companion to open more doors

[Home](#)
[Java FAQs](#)
[600+ Java Q&As](#)
[Career](#)
[Tutorials](#)
[Member](#)
[Why?](#)
[Can u Debug?](#)
[Java 8 ready?](#)
[Top X](#)
[Productivity Tools](#)
[Judging Experience?](#)

[Home](#) › [member-paid](#) › ♦ 11 Spring boot interview questions & answers

♦ 11 Spring boot interview questions & answers

Posted on [November 26, 2015](#) by [Arulkumaran Kumaraswamipillai](#)



Q1. What is the key benefit of using Spring boot?

A1. The key benefit is that you can “build a production ready application from scratch in a matter of minutes”.

Q2. How does spring boot enable you to “build a production ready application from scratch in a matter of minutes”?

A2. It takes the approach of “**convention over configuration**”.

1) The Spring jars dependency management and versioning are simplified as demonstrated in the spring boot example – [Simple Spring Boot Tutorial in 8 steps](#)

Spring Boot’s main benefit is its ability to configure resources based on what it finds in your classpath. If your Maven POM

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](#)

✚ [Job Interview Ice B](#)

✚ [FAQ Core Java Jot](#)

✚ [FAQ JEE Job Inter](#)

✚ [FAQ Java Web Ser](#)

✚ [Java Application Ar](#)

✚ [Hibernate Job Inter](#)

✚ [Spring Job Intervie](#)

✚ [♦ 11 Spring boot](#)

✚ [01: ♥♦ 13 Spring](#)

✚ [01b: ♦ 13 Spring](#)

✚ [04 ♦ 17 Spring b](#)

✚ [05: ♦ 9 Spring B](#)

✚ [Java Key Area Ess](#)

✚ [OOP & FP Essenti](#)

✚ [Code Quality Job I](#)

✚ [SQL, XML, UML, JSC](#)

✚ [Hadoop & BigData Int](#)

includes JPA dependencies and a PostgreSQL driver, then Spring Boot will setup a persistence unit based on PostgreSQL. If you've added a web dependency, then you get Spring MVC configured with sensible defaults.

2) Spring boot is based on an HTTP server. Spring Boot has an embedded version of Tomcat by default, but gives you a way to opt for Jetty server if you wish. This is demonstrated via [Simple Spring Boot Restful Web Service Tutorial](#)

Q3. How do you specify dependencies in Spring boot?

A3. Via **spring-boot-starter-xxxxx**.

```

1
2 <dependencies>
3   <dependency>
4     <groupId>org.springframework.boot</groupId>
5     <artifactId>spring-boot-starter-web</artifactId>
6   </dependency>
7   <dependency>
8     <groupId>org.springframework.boot</groupId>
9     <artifactId>spring-boot-starter-data-jpa</artifactId>
10  </dependency>
11  <dependency>
12    <groupId>com.h2database</groupId>
13    <artifactId>h2</artifactId>
14    <version>1.3.174</version>
15  </dependency>
16  <dependency>
17    <groupId>org.springframework.boot</groupId>
18    <artifactId>spring-boot-starter-test</artifactId>
19    <scope>test</scope>
20  </dependency>
21 </dependencies>
22

```

Since “h2” database dependency is used, Spring boot will configure JPA persistence unit for H2 rather than the HSQLDB, which is the default. This approach is known as the **“Opinionated Defaults Configuration”**

Q4. How will you get Spring boot to use Jetty server instead of Tomcat, which is the default?

A4. By adding the jetty server dependency **“spring-boot-starter-jetty”** in the pom.xml file.

Q5. What documentation would you be using to get started with your enterprise Spring boot application?

- ✚ [Java Architecture Inte](#)
- ✚ [Scala Interview Q&As](#)
- ✚ [Spring, Hibernate, & I](#)
- ✚ [Spring \(18\)](#)
- ✚ [Hibernate \(13\)](#)
- ✚ [01: ♥♦ 15+ Hiber](#)
- ✚ [01b: ♦ 15+ Hiber](#)
- ✚ [02: Understandir](#)
- ✚ [03: Identifying ar](#)
- ✚ [04: Identifying ar](#)
- ✚ [05: Debugging H](#)
- ✚ [06: Hibernate Fil](#)
- ✚ [07: Hibernate mi](#)
- ✚ [08: Hibernate au](#)
- ✚ [09: Hibernate en](#)
- ✚ [10: Spring, Java](#)
- ✚ [11: Hibernate de](#)
- ✚ [12: Hibernate cu](#)
- ✚ [AngularJS \(2\)](#)
- ✚ [Git & SVN \(6\)](#)
- ✚ [JMeter \(2\)](#)
- ✚ [JSF \(2\)](#)
- ✚ [Maven \(3\)](#)
- ✚ [Testing & Profiling/Sa](#)
- ✚ [Other Interview Q&A I](#)
- ✚ [▶ 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 \(7\)](#)

A5. “<https://spring.io/docs>” has lots of getting started guides [<http://spring.io/guides>] & tutorials.

Q6. What is Spring Boot CLI?

A6. CLI stands for **Command Line Interface**, which is a Spring Boot software to run and test Spring Boot applications from command prompt. When you run Spring Boot applications using CLI, then it internally uses **Spring Boot Starter** and **Spring Boot AutoConfigure** components to resolve all dependencies and execute the application. It internally contains Groovy and Grape (JAR Dependency Manager) to add Spring Boot Defaults and resolve all dependencies automatically.

Q7. What are the different ways to generate a Spring boot project?

A7.

1) Using **Maven** as demonstrated in [Simple Spring Boot Tutorial in 8 steps](#)

2) Via online **Spring Initializr** – “<http://start.spring.io/>”. More info at “<http://spring.io/>”.

3) Via **Spring Boot CLI**.

4) Via **Spring Boot IDE**.

Q8 What is the difference between Spring IO & Spring Boot?

A8. 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 IO tutorial in 6 steps](#)

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](#).

- ▣ [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 \(2\)](#)
- ▣ [Akka Tutorial \(9\)](#)
- ▣ [Core Java Tutorials \(2\)](#)
- ▣ [Hadoop & Spark Tuto](#)
- ▣ [JEE Tutorials \(19\)](#)
- ▣ [Scala Tutorials \(1\)](#)
- ▣ [Spring & Hibernate Ti](#)
- ▣ [Tools Tutorials \(19\)](#)
- ▣ [Other Tutorials \(45\)](#)

100+ Java pre-interview coding tests

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

- ▣ [Can you write code? \(1\)](#)
- ▣ [♦ 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 \(1\)](#)

Q9 What are the key components of Spring Boot framework?

A9. Spring Boot has 4 key components.

1) Spring Boot Starter: is responsible for combining a group of common or related dependencies. E.g. spring-boot-starter-actuator, spring-boot-starter-web, spring-boot-starter-data-rest, spring-boot-starter-hateoas, spring-boot-starter-jdbc, spring-boot-starter-tomcat, etc.

2) Spring Boot AutoConfigurator: is responsible for simplifying the wiring up of Spring components. One of the common criticisms of Spring IO framework is that it requires lot of XML or Java based configurations. The Spring Boot AutoConfigurator component will take the burden of wiring up the Spring components. It also reduces the number of annotations. For example, `@SpringBootApplication` = `@Configuration` + `@ComponentScan` + `@EnableAutoConfiguration`.

3) Spring Boot CLI: is responsible for running & testing a Spring Boot application from a command prompt. It internally uses the components “Spring Boot Starters” and “Spring Boot AutoConfigurator”. You can also run Spring Web Applications from a command prompt.

4) Spring Boot Actuator: is responsible for providing production-ready features to a Spring Boot application without having to actually implement these things yourself. it exposes different types of information about the running application – health, metrics, info, env etc. This is not a replacement for a production-grade monitoring solution, but is a good starting point from a development & testing perspective.

Q10 How does Spring Boot work under the hood to simplify the build dependency & configuration?

A10. Spring Boot internally uses Groovy to tap into its features such as JAR dependency resolver engine (i.e. GRAPE) and default import statements.

Q11 What are the benefits of using Spring Boot in your next micro-service application?

**How good
are your?**

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

 [Career Making Know-](#)

 [Job Hunting & Resum](#)

A11. You can quickly build a stand-alone production ready application. It reduces lots of development time and increases the overall productivity due to

- 1) Lesser dependency management effort.
- 2) Lesser boiler plate code to wire up Spring components.
- 3) Easier to integrate within Spring ecosystems like spring-jdbc, spring-web, spring-orm, spring-data, spring-security, etc.
- 4) It follows “Opinionated Configuration Defaults” approach.
- 5) It provides embedded HTTP servers like Tomcat and Jetty to test your applications.
- 6) It provides lots of plugins to develop & test Spring applications with Maven & Gradle. For example, “spring-boot-maven-plugin” to create uber jars that can be deployed to a web server.
- 7) Spring Boot CLI tool helps to develop & test from a command line.

Popular Posts

♦ 11 Spring boot interview questions & answers

825 views

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

767 views

18 Java scenarios based interview Questions and Answers

400 views

001A: ♦ 7+ Java integration styles & patterns interview questions & answers

389 views

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

295 views

♦ 7 Java debugging interview questions & answers

293 views

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

285 views

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

279 views

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

240 views

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

202 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](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](#)

◀ Converting POJOs to Pentaho meta objects and then passing to a Kettle transformation

01: ♦ 19 Java 8 Functional Programming (i.e. FP) interview questions and answers ▶

Posted in member-paid, Spring boot, Spring Job Interview Essentials

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?](#)

Select Category ▼

© 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.

