

# Spring Interview Questions

---

 [codespaghetti.com/spring-interview-questions/](https://codespaghetti.com/spring-interview-questions/)

## SPRING

---

Java Spring Interview Questions, Programs and Examples to help you ace your next Technical interview.



Table of Contents:

---

CHAPTER 1: How To Describe Spring In Interviews

---

CHAPTER 2: Advantages Of Spring Framework

---

CHAPTER 3: Core Spring Interview Questions

---

CHAPTER 4: Spring AOP Interview Questions

---

CHAPTER 5: Spring MVC Interview Questions

---

CHAPTER 6: Spring REST Interview Questions

---

CHAPTER 7: Spring Boot Interview Questions

---

CHAPTER 8: Spring Batch Interview Questions

---

CHAPTER 9: Spring JDBC Interview Questions

---

CHAPTER 10: Spring Security Interview Questions

---

CHAPTER 11: Spring Interview Resources

---

CHAPTER 12: Summary

---

CHAPTER 13: Spring Interview Questions PDF

---

## What Is Spring Framework

---

Spring is one of the most popular Java EE framework. Core concepts of Spring framework are “Dependency Injection” and “Aspect Oriented Programming”.

Spring framework can be used in plain java applications to achieve loose coupling among different application components. This loose coupling is achieved by implementing dependency injection.

Spring framework provides a lot of features and modules for specific tasks such as Spring MVC and Spring JDBC.

Since it's an open source framework with a lot of online resources and active community members, working with Spring framework is easy and fun at same time. Some of the features of spring framework are:

- Spring is Lightweight framework and easier to use in development.
- Spring provides dependency injection or Inversion of Control to write components that are independent.
- Spring IoC container manages Spring Bean life cycle and project specific configurations such as JNDI lookup.
- Spring MVC framework can be used to create web applications as well as restful web services capable of returning XML as well as JSON response.

---

## What Are Advantages Of Using Spring Framework?

---

Some advantages of using Spring Framework are:

---

- Reducing direct dependencies between different components of the application, usually Spring IoC container is responsible for initializing resources or beans and inject them as dependencies.
- Spring framework is divided into several modules, it helps us in keeping our application lightweight.
- Reduces the amount of boiler-plate code, such as initializing objects, open/close resources.
- Spring framework support most of the Java EE features and even much more.
- Writing unit test cases are easy in Spring framework because our business logic doesn't have direct dependencies with actual resource implementation classes.

## Core Spring Interview Questions

---



---

Question: What Are Important Spring Modules?

---

Some of the Spring Framework modules are

- **Spring Context:** This is for dependency injection.
- **Spring JDBC:** This is for JDBC and DataSource support. Spring ORM – for ORM tools support such as Hibernate

- **Spring AOP:** This is for aspect oriented programming.
  - **Spring DAO:** This is for database operations using DAO pattern
  - **Spring Web:** This is for creating web applications.
  - **Spring MVC:** This is for Model View Controller implementation for creating web applications, web services etc.
- 

## Question: What Is Spring Bean?

---

---

Spring bean is a normal java class that is initialized by Spring IoC container.

Spring ApplicationContext is used to get the Spring Bean instance. And spring IoC container manages the life cycle of Spring Bean.

---

## Question: What Is Spring Bean Configuration File?

---

---

Spring Bean configuration file is used to define all the beans that will be initialized by Spring context.

When an instance of Spring ApplicationContext is created. It reads the spring bean xml file. Once the context is initialized, it is used to get different bean instances.

Apart from Spring Bean configuration, this file also contains

-Spring MVC interceptors

-View resolvers

-And other elements to support annotations based configurations

---

## Question: How To Configure A Class As Spring Bean?

---

---

There are three different ways to configure Spring Bean. And here they are

### 1- XML Based Configuration

---

In this configuration we can use bean elements in a context file to configure a Spring Bean. For example:

```
<bean name="testBean" class="com.Codespaghetti.spring.beans.testBean"></bean>
```

### 2- Java Based Configuration

---

A Spring bean can also be configured by using @Bean annotation. This annotation is used with @Configuration class to configure a spring bean. for example:

```
@Configuration
@ComponentScan(value="com.Codespaghetti.spring.main")
public class TestConfiguration {

    @Bean
    public TestService getService(){
        return new TestService();
    }
}
```

In order to get this bean from spring context, we need to use following code

```
AnnotationConfigApplicationContext contex = new AnnotationConfigApplicationContext(
TestConfiguration.class);
TestService testService =contex.getBean(TestService.class);
```

### 3- Annotation Based Configuration

---

You can also use `@Component`, `@Service`, `@Repository` and `@Controller` annotations with classes to configure them as spring bean.

---

#### Question: What Is Spring Bean LifeCycle?

---

Spring Beans are initialized by Spring Container and all the dependencies are injected. When spring context is destroyed, it also destroys all the initialized beans.

This works well in most of the cases but sometimes we want to initialize other resources or do some validation before making our beans ready to use. Spring framework provides support for post-initialization and pre-destroy methods in spring beans.

We can do this by two ways

- 1- By implementing **InitializingBean** and **DisposableBean** interfaces or
  - 2- By using **init-method** and **destroy-method** attribute in spring bean configurations.
- 

#### Question: What Is Bean Wiring?

---

The process of injection spring bean dependencies while initializing it, is called spring bean wiring.

Spring framework also supports the possibility of doing autowiring. We can use **@Autowired** annotation with fields.

For auto wiring annotation to work, we need to enable annotation based configuration in spring bean. This can be done by **context:annotation-config** element.

---

---

## Question: What are different types of Spring Bean autowiring?

---

There are four types of autowiring available in Spring framework.

- **Autowire** byName
  - **Autowire** byType
  - **Autowire** by constructor
  - **Autowiring** by **@Autowired** and **@Qualifier** annotations
- 

---

## Question: Are Spring Beans Thread Safe?

---

The default scope of Spring bean is singleton, so there will be only one instance per context. So this means that Spring beans are not thread safe by default.

However we can change spring bean scope to request, prototype or session to achieve thread-safety at the cost of performance.

---

---

## Question: What Are Some Design Patterns Used In The Spring Framework?

---

Some of the design patterns used in spring framework are

- Singleton Pattern
  - Factory Pattern
  - Proxy Pattern
  - Template Method Pattern
  - Front Controller
  - Prototype Pattern
  - Adapter Pattern
  - Data Access Object
  - Model View Controller
- 

---

## Question: What Is Scope Prototype?

---

Scope prototype means that every time an instance of the Bean is called. Spring will create a new instance and return it.

This differs from the default *singleton* scope, where a single object instance is instantiated once per Spring IoC container.

---

---

## Question: How To Get ServletContext and ServletConfig Objects In Spring?

---

---

In spring bean, container specific objects can be retrieved in the following two ways.

**FIRST:** By Implementing a spring Aware interfaces

**SECOND:** Using @Autowired annotation with bean variable of type ServletContext and ServletConfig.

---

Question: **BeanFactory VS**

---

**ApplicationContext?**

---

Spring BeanFactory is an interface which represents a container that provides and manages bean instances.

ApplicationContext is an interface representing a container holding all information, and beans in the application.

It also extends the BeanFactory interface but the default implementation instantiates beans eagerly when the application starts. This behavior can be overridden for individual beans.

---

Question: What Is **Dependency Injection?**

---

Dependency Injection allows the developers to remove the hard coded dependencies among components and make the application loosely coupled, extendable and maintainable.

Some of the benefits of using Dependency Injection are:

- Loose coupling
  - Separation of Concerns
  - Boilerplate Code reduction
  - Unit testing with ease
- 

Question: **How To Implement Dependency Injection In Spring?**

---

In Spring framework dependency injection can be implemented by using Spring XML based configuration or by using annotation based configuration.

---

Question: **How To Inject Beans In Spring?**

---

Beans in Spring framework can be injected in few different ways

- By using setter Injection
- By using constructor Injection
- By using field Injection

The configuration can be done using XML files or annotations.

---

## Question: What Are Different Modes Of Autowiring Supported By Spring?

---

Spring supports following modes to instruct Spring container to use autowiring for dependency injection.

**No:**

Default, no auto wiring

**byName:**

Autowiring by property name. Spring container looks at the properties of the beans on which *autowire* attribute is set to *byName* in the XML configuration file and it tries to match it with name of bean in xml configuration file.

**byType:**

Autowiring by property datatype. Spring container looks at the properties of the beans on which *autowire* attribute is set to *byType* in the XML configuration file.

It then tries to match and wire a property if its **type** matches with exactly one of the beans name in configuration file.

**constructor:**

byType mode in constructor argument.

**autodetect:**

Spring first tries to wire using autowire by *constructor*, if it does not work, Then spring tries to autowire by *byType*.

## Spring Aspect Oriented Interview Questions

---





---

## Question: **What Is Aspect Oriented Programming?**

---

In Object Oriented Programming, modularity of application is achieved by classes, whereas in AOP application modularity is achieved by aspects and they are configured to cut across different class methods.

AOP takes out the direct dependency of cross-cutting tasks from classes that is not possible in normal object oriented programming.

AOP provides the way to dynamically add the cross-cutting concern before, after or around the actual logic by using simple pluggable configurations.

It makes the code easy to maintain. You can add/remove concerns without recompiling complete code.

Spring AOP can be used by following 2 ways.

- 1) By Aspect Annotation Style
- 2) By XML Configuration Style

---

## Question: **What Is The Difference Between Concern and Cross-cutting Concern In Spring AOP?**

---

**Concern:** can be defined as a functionality we want to implement to solve a specific business problem. E.g. in any eCommerce application different concerns (or modules) could be inventory management or user management etc.

**Cross cutting:** concern is a concern which is applicable throughout the application, e.g. logging , security and data transfer are the concerns which are needed in almost every module of an application. Therefore they are termed as cross-cutting concerns.

---

## Question: What is Aspect, Advice, JointPoint and Advice Arguments in AOP

---

**Aspect:** Aspect is a class that implements cross-cutting concerns, such as transaction management.

Aspects can be a normal class configured in spring bean configuration file or it can be spring aspect using `@Aspect` annotation.

**Advice:** Advice is the action taken for a particular join point. In terms of programming, they are methods that gets executed when a specific join point with matching pointcut is reached in the application.

**Join Point:** A join point is a specific point in the application such as method execution, exception handling. In Spring AOP a join points is always the execution of a method.

**Advice Arguments:** We can pass arguments in the advice methods. We can use `args()` expression in the pointcut to be applied to any method that matches the argument pattern.

If we use this, then we need to use the same name in the advice method from where argument type is determined.

---

## Question: What Are Different Modes Of Autowiring Supported By Spring?

---

Spring supports following modes to instruct Spring container to use autowiring for dependency injection.

**No:**

Default, no auto wiring

**byName:**

Autowiring by property name. Spring container looks at the properties of the beans on which *autowire* attribute is set to *byName* in the XML configuration file and it tries to match it with name of bean in xml configuration file.

**byType:**

Autowiring by property datatype. Spring container looks at the properties of the beans on which *autowire* attribute is set to *byType* in the XML configuration file. It then tries to match and wire a property if its **type** matches with exactly one of the beans name in configuration file.

**constructor:**

byType mode in constructor argument.

**autodetect:**

Spring first tries to wire using auto wire by *constructor*, if it does not work, Then spring tries to auto wire by *byType*.

---

## Question: **What Are Some Available AOP implementations**

---

Main java based AOP implementations are listed below :

1. AspectJ
  2. Spring AOP
  3. JBoss AOP
- 

## Question: **How Many Advice Types In Spring?**

---

In spring an advice is implementation of cross-cutting concern. You can apply these on other modules of application.

There are 5 types of Advices

1. **Before advice:** Advice that executes before a join point, you need to use `@Before` annotation to enable it.
  2. **After returning advice:** Advice to be executed after a join point completes normally. To use this advice, use `@AfterReturning` annotation.
  3. **After throwing advice:** Advice to be executed if a method exits by throwing an exception. To use this advice, use `@AfterThrowing` annotation.
  4. **After advice:** Advice to be executed regardless of the means by which a join point exits. To use this advice, use `@After` annotation.
  5. **Around advice:** Advice that surrounds a join point such as a method invocation. This is the most powerful kind of advice. To use this advice, use `@Around` annotation.
- 

## Question: **What Is Spring AOP Proxy?**

---

A proxy is an object that looks like another object, but adds special functionality behind the scene.

AOP proxy is an object created by the AOP framework in order to implement the aspect contracts in runtime.

Spring AOP defaults to using standard JDK dynamic proxies for AOP proxies. This enables any interface to be proxied.

---

Spring AOP can also use CGLIB proxies. This is necessary to proxy classes, rather than interfaces. CGLIB is used by default if a business object does not implement an interface.

---

---

Question: **What Is Weaving?**

---

Weaving is the process of linking aspects with other out side application types or objects to create an advised object.

---

This can be done at compile time, load time, or at runtime. Spring AOP, like other pure Java AOP frameworks, performs weaving at runtime only. In contrast, AspectJ framework supports both compile-time and load-time weaving.

---

Spring MVC Interview Questions

---



Question: What Is Spring MVC Framework?

---

---

The Spring MVC framework provides Model-View-Controller architecture and various components that can be used to develop flexible and loosely coupled web applications.

The MVC pattern results in separating the different aspects of the application, while providing a loose coupling between these elements.

- The **Model** encapsulates the application data and in general they will consist of POJO.
- The **View** is responsible for rendering the model data and in general it generates HTML output that the client's browser can interpret.
- The **Controller** is responsible for processing user requests and building an appropriate model and passes it to the view for rendering.

---

## Question: How To Handle Exceptions In Spring MVC Environment?

---

There are 3 ways to handle exceptions in Spring MVC

### 1- @ExceptionHandler

By Using *@ExceptionHandler* at controller level. This approach has a major feature, the *@ExceptionHandler* annotated method is only active for that particular controller, not globally for the entire application.

### 2- @HandlerExceptionResolver

By using *HandlerExceptionResolver*, this will resolve any exception thrown by the application.

### 3- @ControllerAdvice

By using *@ControllerAdvice*, Spring 3.2 brings support for a global *@ExceptionHandler* with the *@ControllerAdvice* annotation, which enables a mechanism that breaks away from the older MVC model and makes use of *ResponseEntity* along with the type safety and flexibility of *@ExceptionHandler*

---

## Question: What Is Spring MVC Interceptor And How To Use It?

---

Spring MVC Interceptors allow us to intercept a client request and process it at three places.

1. Before handling
2. After handling
3. After completion of a request. The interceptor can be used for cross-cutting concerns and to avoid repetitive handler code like logging, changing globally used parameters in Spring model, etc.

## Question: What is a Controller in Spring MVC?

---

**Controller** - Controller acts on both model and view. It controls the data flow into model object and updates the view whenever data changes. It keeps view and model separate.

Simply put, all the requests processed by the *DispatcherServlet* are directed to classes annotated with `@Controller`. Each controller class maps one or more requests to methods that process and execute the requests with provided inputs.

---

## Question: What's The Difference Between @Component, @Controller, @Repository & @Service annotations in Spring?

---

**@Component:** is used to indicate that a class is a component. These classes are used for auto detection and configured as bean.

**@Controller:** is a specific type of component. It is used in MVC applications and mostly used with RequestMapping annotation.

**@Repository:** annotation is used to indicate that a component is used as repository and a mechanism to store-retrieve-search data.

**@Service:** is used to indicate that a class is a Service.

---

## Question: What is ViewResolver in Spring?

---

ViewResolver implementations are used to resolve the view pages by name. Usually they are configured in the spring bean configuration file.

```
<!-- Resolves views selected for rendering by @Controllers to .jsp resources in the
/WEB-INF/views directory -->
<beans:bean
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
  <beans:property name="prefix" value="/WEB-INF/views/" />
  <beans:property name="suffix" value=".jsp" />
</beans:bean>
```

---

## Question: What Is A MultiPartResolver And When It Is Used?

---

Spring comes with MultipartResolver to handle file upload in web application. There are two concrete implementations included in Spring

- **CommonsMultipartResolver** for Jakarta Commons FileUpload
  - **StandardServletMultipartResolver** for Servlet 3.0 Part API
-

## Question: What is ContextLoaderListener?

---

ContextLoaderListener is the listener class used to load root context and define spring bean configurations that will be visible to all other contexts. It's configured in web.xml file as

```
<context-param>
```

```
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/spring/root-context.xml</param-value>
</context-param>
```

```
<listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-
class>
</listener>
```

---

## Question: How We Can Create Spring MVC application?

---

For creating a Spring MVC application, following tasks are needed.

- 1- Add spring-context and spring-webmvc dependencies in the project.
- 2- Configure DispatcherServlet in the web.xml file to handle requests through spring container.
- 3- Spring bean configuration file to define beans, if using annotations then it has to be configured here. Also we need to configure view resolver for view pages.
- 4- Controller class with request mappings defined to handle the client requests.

These 4 steps are enough to create a simple Spring MVC application.

---

## Question: How To Achieve Localization In Spring MVC?

---

Spring provides excellent support for localization or i18n through resource bundles. Basis steps needed to make our application localized are

1. Creating message resource bundles for different locales, such as messages\_en.properties etc.
2. Defining messageSource bean in the spring bean configuration file of type ResourceBundleMessageSource or ReloadableResourceBundleMessageSource.
3. For change of locale support, define localeResolver bean of type CookieLocaleResolver and configure LocaleChangeInterceptor interceptor. Example configuration can be like below:
4. Use spring:message element in the view pages with key names, DispatcherServlet picks the corresponding value and renders the page in corresponding locale and return as response.

---

## Question: Can We Send An Object As Response of Controller handler method?

---

Yes, this is possible by using **@ResponseBody** annotation. This is how we send JSON or XML based response in restful web services.

---

## Question: How To Upload File In Spring MVC Application?

---

Spring provides built-in support for uploading files through **MultipartResolver** interface implementations.

---

## Question: How to validate form data in Spring Web MVC Framework?

---

Spring provides built-in support for uploading files through **MultipartResolver** interface implementations.

---

## Question: What Is Interceptor Spring MVC Framework?

---

Spring MVC Interceptors allow us to intercept client request and process it. We can intercept client request at three places

- **preHandle**
  - **postHandle**
  - **afterCompletion**
- 

## Question: What Is Role of @Qualifier Annotation?

---

It is used simultaneously with the *@Autowired* annotation to avoid confusion when multiple instances of a bean type are present.

---

## Question: What Is Role of @Autowire Annotation?

---

*@Autowired* annotation can be used with fields or methods for injecting a bean by type.

This annotation allows Spring to resolve and inject collaborating beans into your bean

---

## Question: What Is Role of @Required Annotation?

---



---

The `@Required` annotation is used on setter methods, and it indicates that the bean property that has this annotation must be populated at configuration time. Otherwise, the Spring container will throw a `BeanInitializationException` exception.

---

## Spring REST Interview Questions

---



---

Question: **What Does REST Stand For?**

---

REST stands for **Representational State Transfer**. Which uses HTTP protocol to send data from client to server. Like a book in the server can be delivered to the client using JSON or XML.

---

Question: **What Are Safe REST Operations?**

---

REST API uses HTTP methods to perform operations. Some of the HTTP operations which doesn't modify the resource at the server is known as safe operations e.g. GET and HEAD. Operations like PUT, POST, and DELETE are unsafe because they modify the resource on the server.

---

Question: **Is REST scalable?**

---

Yes, REST is Scalable. It doesn't depends upon a specific technology both at client and server side. You can use any technology like Java, C++, Python or JavaScript to create

RESTful web services and consume them at the client side.

---

## Question: **What Are The Advantages Of RestTemplate?**

---

RestTemplate class is an implementation of Template method pattern in Spring framework. It simplifies the interaction of RESTful Web Services on the client side.

---

## Question: **Which HTTP Methods REST use?**

---

REST can use any HTTP methods but the most popular ones are

- GET for retrieving a resource
  - POST for creating a resource
  - PUT for updating resource
  - DELETE for removing a resource from the server
- 

## Question: **Is REST Stateless?**

---

Yes, REST API is stateless because it is based on HTTP protocol, which is also stateless. A Request in REST API should contain all the details required it to process.

And it should not rely on some data maintained at the server end. REST specification put a constraint to make it stateless and developers should keep that in mind while designing REST API.

---

## Question: **What does @RequestMapping annotation do?**

---

The @RequestMapping annotation is used to map web requests to Spring Controller methods. You can map request based upon HTTP methods e.g. GET and POST and various other parameters.

Here are some examples

```
1 @RequestMapping (method = RequestMethod.POST, consumes="application/json")
2 public Book save(@RequestBody Book testBook) {
3     return bookRepository.save(testBook);
4 }
```

You can similarly create other handler methods to produce JSON or XML.

---

### Question: **What is the HTTP status return code for a successful DELETE statement?**

---

There is no strict rule with respect to what status code your REST API should return after a successful DELETE operation.

In general, if the DELETE operation is successful and the response body is empty return 204.

If the DELETE request is successful and the response body is NOT empty, return 200

---

### Question: **What Is @ResponseStatus annotation in Spring MVC?**

---

The @ResponseStatus annotation is required during error handling in Spring MVC and REST. Normally when an error or exception is thrown at server side, web server returns "HTTP status code 500 – Internal server error". But for REST clients. User needs to send a proper status code e.g. 404 if the resource is not found. That's where you can use @ResponseStatus annotation, which allows you to send custom HTTP status code along with proper error message in case of Exception.

---

### Question: **Is REST Secure?**

---

Normally REST is not secure but you can secure it by using Spring security. Developers can also enable HTTP basic authentication by using HTTP Spring security configuration file. Similarly, we can expose REST API using HTTPS to add an extra layer of security.

---

### Question: **How To Create Restful Web Service In Spring?**

---

We can use Spring Framework to create Restful web services that returns JSON data. Spring provides integration with Jackson JSON API that can be used to send JSON response in restful web service.

Following steps are needed to configure Spring MVC application to send JSON response:

-Adding Jackson JSON dependencies

-Configure RequestMappingHandlerAdapter bean in the spring bean configuration file and set the messageConverters property to MappingJackson2HttpMessageConverter bean.

-In the controller handler methods, return the Object as response using @ResponseBody annotation.

-You can invoke the rest service through any API, but if you want to use Spring then we can easily do it using RestTemplate class.

## Spring Boot Interview Questions

---



---

### Question: What Is Spring Boot Framework?

---

Spring Boot aims to make it easy to create Spring-powered, production-grade applications and services with minimum fuss.

It takes an opinionated view of the Spring platform so that new and existing users can quickly get to the bits they need.

You can use it to create stand-alone Java applications that can be started using `'java -jar'` or more traditional WAR deployments. We also provide a command line tool that runs 'spring scripts'.

---

### Question: Spring Boot vs Spring MVC vs Spring?

---

#### Spring Framework

Spring Framework is based on Dependency Injection. At the core of all Spring Modules is Dependency Injection or IOC Inversion of Control.

When DI or IOC is used properly, we can develop loosely coupled applications. And loosely coupled applications can be easily unit tested.

## Spring MVC

Spring MVC Framework provides decoupled way of developing web applications. With simple concepts like Dispatcher Servlet, ModelAndView and View Resolver, it makes it easy to develop web applications.

## Spring Boot

The problem with Spring and Spring MVC is the amount of configuration that is needed.

Spring Boot solves this problem through a combination of Auto Configuration and Starter Projects.

Spring Boot also provide a few non functional features to make building production ready applications easier and much faster.

---

### Question: **What are Spring Boot Starter Projects?**

---

Starter are a set of convenient dependency descriptors that you can include in your application.

You get a single entry point for all the spring and related technology that you need, without having to hunt through sample code and copy paste loads of dependency descriptors.

For example, if you want to get started using Spring and JPA for database access, just include the spring-boot-starter-data-jpa dependency in your project, and you are good to go.

---

### Question: **What Starter Project Options Do Spring Boot provides?**

---

Spring Boot provides following starter projects

- spring-boot-starter-web-services
- spring-boot-starter-web
- spring-boot-starter-test
- spring-boot-starter-jdbc
- spring-boot-starter-hateoas
- spring-boot-starter-security
- spring-boot-starter-data-jpa
- spring-boot-starter-data-rest

---

### Question: **What Is The Java Version Required for Spring Boot 2 and Spring 5?**

---

Spring Boot 2.0 requires Java 8 or later. Java 6 and 7 are no longer supported.

## Question: **Why spring-boot-maven-plugin Is Needed?**

---

Spring-boot-maven-plugin provides a few commands to package the code as a jar file or run the application

- spring-boot:run runs your Spring Boot application.
  - spring-boot:repackage repackages your jar/war to be executable.
  - spring-boot:start and spring-boot:stop to manage the lifecycle of your Spring Boot application (i.e. for integration tests).
  - spring-boot:build-info generates build information that can be used by the Actuator.
- 

## Question: **What is Spring Data?**

---

Spring Data's mission is to provide a familiar and consistent, Spring-based programming model for data access while still retaining the special traits of the underlying data store.

It makes it easy to use data access technologies, relational and non-relational databases, map-reduce frameworks, and cloud-based data services.

To make it simpler, Spring Data provides Abstractions (interfaces) you can use irrespective of underlying data source.

---

## Question: **What is Spring Data REST?**

---

Spring Data REST can be used to expose HATEOAS RESTful resources around Spring Data repositories.

---

## Question: **What Happens When Spring Boot Application Run as Java Application?**

---

If you are using Eclipse IDE, Eclipse maven plugin ensures that as soon as you add a dependency or make a change to the class file, it is compiled and ready in the target folder! And after that its just like any other Java application. When you launch the java application, then the spring boot auto configuration magic kicks in. It launches up tomcat when it sees that you are developing a web application.

---

## Question: **RequestMapping VS GetMapping?**

---

**RequestMapping:** It can be used with GET, POST, PUT or any other request methods using the method attribute on the annotation.

**GetMapping:** It is specific to GET method. In reality GetMapping is an improvement of RequestMapping for better clarity.

---

## Question: Why Profiles Are Necessary?

---

Profiles provide ability for application developers to specify different application configuration for different environments.

Application development lifecycle goes through various stages and gets deployed on different environments. And often times it is necessary to have different configurations in each environment.

Spring and Spring Boot provide features where developers can specify

- What is the configuration for different environments in different profiles?
- Set the active profile for a specific environment.

Spring Boot will pick up the application configuration based on the active profile that is set in a specific environment.

## Spring Batch Interview Questions

---



## Question: What Is Spring Batch Framework?

---

Spring Batch is a lightweight framework designed to enable the development of robust batch applications.

Spring Batch framework is based on productivity and POJO based application development approach.

It allows the developers to use more advanced enterprise services when necessary.

---

## Question: Explain Spring Batch Architecture?

---

The Spring batch framework architecture contains three main components

**-Application:** It contains all the jobs and code developers write using the Spring Batch framework.

**-Batch Core:** It contains all the API classes that are needed to control and launch a Batch Job

**-Batch Infrastructure:** It contains the readers, writers, and services used by both application and Batch components.

---

## Question: What Are Job And Step?

---

In a Spring Batch application, a job is the batch process that is to be executed. It runs from start to finish without interruption. This job is further divided into steps.

We will configure a job in Spring Batch using an XML file or a Java class. Following is the XML configuration of a Job in Spring Batch.

### Step

---

A **step** is an independent part of a job which contains the necessary information to define and execute the job (its part).

As specified in the diagram, each step is composed of an ItemReader, ItemProcessor (optional) and an ItemWriter. **A job may contain one or more steps.**

---

## Question: What Are Readers, Writers And Processors?

---

An **item reader** reads data into a Spring Batch application from a particular source, whereas an **item writer** writes data from the Spring Batch application to a particular destination.

An **Item processor** is a class which contains the processing code which processes the data read into the spring batch. If the application reads "**n**" records, then the code in the processor will be executed on each record.

When no reader and writer are given, a **tasklet** acts as a processor for SpringBatch. It processes only a single task. For example, if we are writing a job with a simple step in it



where we read data from MySQL database and process it and write it to a file (flat), then our step uses –

- A **reader** which reads from MySQL database.
- A **writer** which writes to a flat file.
- A **custom processor** which processes the data as per our wish.

```
<job id = "helloWorldJob">
  <step id = "step1">
    <tasklet>
      <chunk reader = "mysqlReader" writer = "fileWriter"
        processor = "CustomitemProcessor" ></chunk>
    </tasklet>
  </step>
</ job>
```

Spring Batch provides a long list of **readers** and **writers**. Using these predefined classes, we can define beans for them. We will discuss **readers** and **writers** in greater detail in the coming chapters.

---

## Question: What Is JobRepository?

---

A Job repository in Spring Batch provides Create, Retrieve, Update, and Delete (CRUD) operations for the JobLauncher, Job, and Step implementations. We will define a job repository in an XML file as shown below.

```
<job-repository id = "jobRepository"/>
```

In addition to **id**, there are some more options (optional) available. Following is the configuration of job repository with all the options and their default values.

```
<job-repository id = "jobRepository"
  data-source = "dataSource"
  transaction-manager = "transactionManager"
  isolation-level-for-create = "SERIALIZABLE"
  table-prefix = "BATCH_"
  max-varchar-length = "1000"/>
```

---

## Question: What Is JobLauncher?

---

JobLauncher is an interface which launches the Spring Batch job with the **given set of parameters**. **SampleJoblauncher** is the class which implements the **JobLauncher** interface. Following is the configuration of the JobLauncher.

```
<bean id = "jobLauncher"
  class = "org.springframework.batch.core.launch.support.SimpleJobLauncher">
  <property name = "jobRepository" ref = "jobRepository" />
</bean>
```

---

## Question: What Is **Spring Batch Tasklet**?

---

The Tasklet which is a simple interface with just one method execute. Using this we can perform single tasks like executing queries, deleting files.

A: Spring Batch provides a Tasklet interface, which will be called to perform a single task only, like clean or delete or set up resources before or after any step execution.

---

---

## Question: How To **schedule A Spring Batch Job**?

---

Spring Batch can be scheduled using Cron Job.

---

---

## Question: **What Are Important Features Of Spring Batch Framework**?

---

Here is a list of important Spring batch framework

**Restorability:** Restart a batch program from where it failed.

**Different Readers and Writers:** Provides great support to read from text files, CSV, JMS, JDBC, Hibernate, iBatis etc. It can write to JMS, JDBC, Hibernate, files and many more.

**Chunk Processing:** If we have 1 Million records to process, these can be processed in configurable chunks (1000 at a time or 10000 at a time).

Easy to implement proper transaction management even when using chunk processing.

Easy to implement parallel processing. With simple configuration, different steps can be run in parallel.

---

---

## Question: **What Is Parallel Processing In Batch Framework**?

---

Parallel processing enables multiple batch runs jobs to run in parallel to reduce the total elapsed batch processing time. Parallel processing is simpler as long as the same file or database table is not shared among the processes otherwise the processes should process partitioned data.

Another approach would be using a control table for maintaining interdependencies and to track each shared resource in use by any process or not.

Other key issues in parallel processing include load balancing and the availability of general system resources such as files, database buffer pools etc. Also note that the control table itself can easily become a critical resource.

---

---

## Question: **What Are Spring Batch Metadata Schema**?

---

The Spring Batch Meta-Data tables are used to persist batch domain objects such as JobInstance, JobExecution, JobParameters, and StepExecution for internally managing the Batch Jobs.

The JobRepository is responsible for saving and storing each Java object into its correct table

---

Question: **What Are Spring Batch Metadata Schema?**

---

JobLauncher represents a simple interface for launching a Job with a given set of JobParameters.

## Spring JDBC Interview Questions

---



---

Question: What Is **Spring JdbcTemplate class**?

---

Spring Framework provides excellent integration with JDBC API and provides JdbcTemplate utility class. Developers can use this class to avoid boiler plate code from database logic.

---

Question: **How To Achieve Transaction Management in Spring?**

---

Spring framework provides transaction management support through Declarative Transaction Management as well as programmatic transaction management.

Declarative transaction management is most widely used because it's easy to use and works in most of the cases.

We use annotate a method with `@Transactional` annotation for Declarative transaction management.

We need to configure transaction manager for the DataSource in the spring bean configuration file.

---

### Question: **What is Spring DAO?**

---

Spring DAO support is provided to work with data access technologies like JDBC, Hibernate in a consistent and easy way.

---

### Question: **How to integrate Spring and Hibernate Frameworks?**

---

Spring ORM module is used to integrate Spring and Hibernate frameworks. Spring ORM provides support for using Spring declarative transaction management

---

### Question: **What Are Advantages of Spring JdbcTemplate?**

---

Spring simplifies database access handling with the Spring JDBC Template. For example a Job repository in Spring Batch provides Create, Retrieve, Update, and Delete (CRUD) operations for the JobLauncher.

Spring JdbcTemplate eliminates all the above problems of JDBC API. It provides methods to write the queries directly.

The Spring JDBC Template has many advantages compared to the standard JDBC.

- The Spring JDBC template automatically cleans up the resources
  - The Spring JDBC template converts the standard JDBC SQLExceptions into RuntimeExceptions. This ensures faster response time to identify and eliminate errors.
- 

### Question: **How to use Tomcat JNDI DataSource in Spring Web Application?**

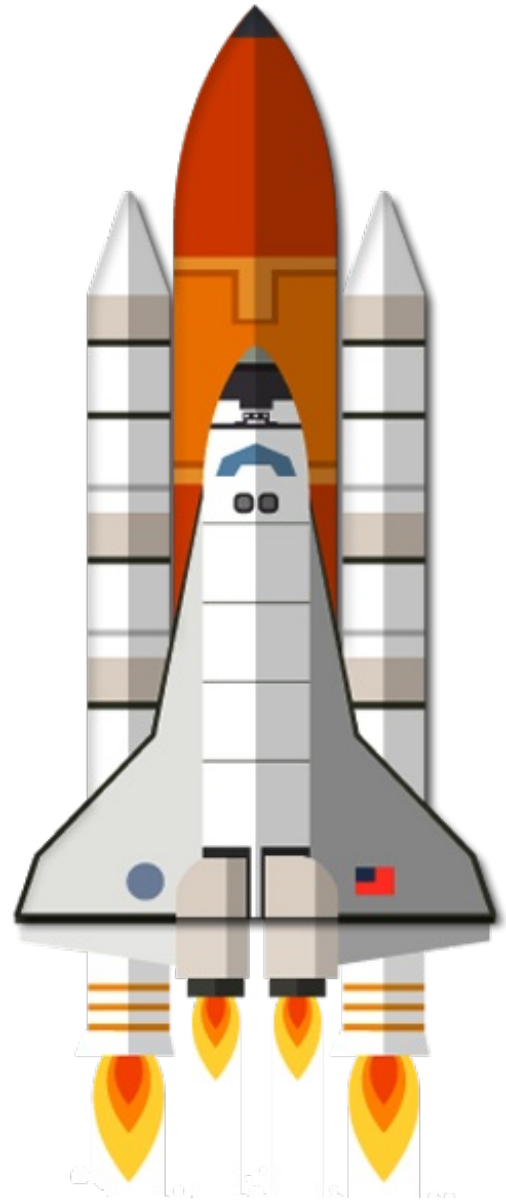
---

For using servlet container configured JNDI DataSource, we need to configure it in the spring bean configuration file and then inject it to spring beans as dependencies. Then we can use it with JdbcTemplate to perform database operations.

---

## Spring Security Interview Questions

---



---

## Question: What Is **Spring Security**?

---

Spring Security is a powerful and highly customizable framework that focuses on providing both authentication and authorization to Java applications.

Like all Spring projects, the real power of Spring Security is found in how easily it can be extended to meet custom requirements

It is the de facto standard for securing Spring-based applications.

Some of the features of Spring Security are

- Comprehensive and extensible support for both Authentication and Authorization
  - Servlet API integration Optional integration with Spring Web MVC
  - Protection against attacks like session fixation, click jacking, cross site request forgery, SQL injections etc
-

---

## Question: **How To Achieve Transaction Management in Spring?**

---

Spring framework provides transaction management support through Declarative Transaction Management as well as programmatic transaction management features.

Declarative transaction management is most widely used because it's easy to configure and works smoothly.

A method needs to be annotated with `@Transactional` annotation for Declarative transaction management. We also need to configure transaction manager for the `DataSource` in the spring bean configuration file.

---

---

## Question: **Which Filter Class Is Needed For Spring Security?**

---

for **XML** file we need to configure `org.springframework.web.filter.DelegatingFilterProxy` in `web.xml` for **Servlet 3**, we can use `AbstractSecurityWebApplicationInitializer`. It configures `DelegatingFilterProxy` and `ContextLoaderListener`.

---

---

## Question: **What Is Difference Between ROLE\_USER and ROLE\_ANONYMOUS?**

---

**ROLE\_ANONYMOUS**: is the default role assigned to an unauthenticated user when a configuration uses Spring Security's "anonymous authentication" filter. This is enabled by default.

**ROLE\_USER**: has no meaning unless this role is assigned to users when they are authenticated.

---

---

## Question: **How to configure DelegatingFilterProxy?**

---

In the `web.xml` we add the `DelegatingFilterProxy`. This is a delegating proxy to automatically intercept a URL with a particular pattern to apply spring security.

---

---

## Question: **What Is Difference Between @Secured and @PreAuthorize In Spring Security?**

---

`PreAuthorize` is used to allow method access to user if a role is assigned to him. For example `PreAuthorize("hasRole('ROLE_role1')`

and Using `@Secured({"role1", "role2"})` is treated as an OR

---

---

## Question: **What is Authentication and Authorization in Spring Security?**

---

---

**Authentication:** An application needs to know who is accessing it. So authentication is related to word *who*.

Application will check it by a login form. User will enter user name and password and these inputs will be validated by the application.

Once the validation is successful, user is declared as authenticated.

**Authorization:** Authorization is a process to check whether user can access the application or not.

---

---

## Question: **How To Encode Password By Using Spring security?**

---

To encode password, spring security provides `<password-encoder/>` tag.

---

---

## Question: **What Is Channel Security In Spring?**

---

Spring security provides the feature to secure the specific URL patterns. For any URL pattern if we want to allow only HTTPS access, we have to do a small configuration in the spring security configuration.

```
<intercept-url pattern="/login" access="ROLE_USER" requires-channel="https" />
```

---

---

## Question: **How Sessions Are Managed In Spring Security?**

---

To manage session we need to use `<session-management>` tag within `<http>` tag in our spring context XML.

```
<session-management
```

```
invalid-session-url="/login" session-fixation-protection="newSession" > <concurrency-control max-sessions="1" error-if-maximum-exceeded="true" /> </session-management>
```

---

---

## Question: **How Can We Enable Web Security Using Java Configuration In Spring?**

---

Spring security provides `@EnableWebSecurity` annotation which is used with `@Configuration`.

`@EnableWebSecurity` annotation provides spring security configuration defined in `WebSecurityConfigurer` or `WebSecurityConfigurerAdapter`.

To override security configuration we can extend `WebSecurityConfigurerAdapter` class in our java configuration class.

```
@Configuration @EnableWebSecurity public class Security extends  
WebSecurityConfigurerAdapter {} /code>
```

---

## Question: **What Is Method Security?**

---

Spring Security uses AOP for security at the method level

---

## Question: **What Do @Secured and @RolesAllowed Do?**

---

@Secured and @RolesAllowed both annotation provide method level security in to Spring Beans.

**@Secured** is Spring Security annotation from version 2.0 onwards Spring Security.

But **@RolesAllowed** is JSR 250 annotation.

Spring Security provides the support for JSR 250 annotation as well for method level security.

---

## Question: **What Is Security Context?**

---

Security context in Spring Security includes details of the principal currently using the application.

Security context is always available to methods in the same thread of execution, even if the security context is not explicitly passed around as an argument to those methods.

---

## Question: **Does Spring Security Support Password Hashing?**

---

Spring Security provides support for password hashing. And a salt is used to prevent dictionary attacks against the key in the event your encrypted data is compromised.

---

## Question: **What Is OAuth2? How To Implement It Using Spring Boot Security?**

---

OAuth (Open Authorization) is a simple way to publish and interact with protected data.

It is an open standard for token-based authentication and authorization on the Internet.



It allows an end user's account information to be used by third-party services, such as Facebook, but without exposing the user's password.

The OAuth specification describes five grants for acquiring an access token:

- Authorization code grant
- Implicit grant
- Resource owner credentials grant
- Client credentials grant
- Refresh token grant

## Spring Resources

---

In this section you will find a list of excellent Spring resources.

### Java Spring Interview Questions:

---

- <https://www.journaldev.com/2696/spring-interview-questions-and-answers>
- <http://www.baeldung.com/spring-interview-questions>
- <https://www.edureka.co/blog/interview-questions/spring-interview-questions/>

### Online Java Spring Courses [Free+ Paid]:

---

- <https://www.udemy.com/spring-boot-getting-started/>
- <https://www.udemy.com/spring-framework-video-tutorial/>
- <https://www.pluralsight.com/courses/spring-mvc4-introduction>
- <https://www.udemy.com/spring-hibernate-tutorial/>

### Best Java Spring Blogs:

---

- <http://www.baeldung.com/>
- <https://blog.frankel.ch/tag/spring/>
- <https://www.udemy.com/spring-hibernate-tutorial/>
- <https://spring.io/blog>
- <http://blog.codeleak.pl/>

### Best Books on Spring Framework:

---

**1- Spring in Action:** Spring in Action is a best guide on Spring Framework. This book covers the following features

- Spring 4
- Spring MVC,
- REST with Spring
- Web Flow
- Spring Security

**Link:** <https://www.manning.com/books/spring-in-action-fourth-edition>

**2- Spring Microservices in Action:** In this book, John Carnell will teach you how to build micro service based applications using Java and the Spring platform. This book will also teach you about all the technology you need to develop and deploy your micro services in the real world scenarios. **Link:** <https://www.manning.com/books/spring-microservices-in-action>

**3-Spring 5 Recipes: A Problem-Solution Approach:** If you like to learn new concepts by following how to's tutorial and recipe books, then this is the book for you.

This book also covers It also covers Spring Batch, NoSQL, and Big Data for building and integrating various cloud computing services and resources

**Link:** <https://www.apress.com/in/book/9781484227893>

## Summary

---

Spring is a very popular and widely used java EE framework. It is adopted by all the major corporations and thus have a very good job market.

But it is not very easy to appear in a Java Spring related interviews. You have already seen in this guide that spring framework spans across a wide range of topics. As a developer you need to be aware of these concepts.

The aim of this guide was to give you access to all the interview questions in each area of Spring. These interview questions range from beginners to expert level developers.

In order to beat the competition and increase your chances of success in interviews, make sure that you go through all of these questions.

### Other Related Interview Questions:

---

- [AngularJa interview Questions](#)
- [Java MultiThreading Interview Questions](#)
- [Interview Questions](#)
- [Phone Interview Questions](#)

## AngularJS Interview Questions PDF

---

[AngularJS-Interview-Questions-PDF](#)

## About The Author

---

## References

---

- <https://spring.io/>
- [https://en.wikipedia.org/wiki/Spring\\_Framework](https://en.wikipedia.org/wiki/Spring_Framework)
- <https://www.journaldev.com/2696/spring-interview-questions-and-answers>
- <http://www.baeldung.com/spring-interview-questions>
- <https://github.com/in28minutes/spring-interview-guide>
- <https://stackoverflow.com/questions/1061717/what-exactly-is-spring-framework-for>
- <http://www.javainuse.com/spring/spring-security-interview-questions>
- <https://www.concretepage.com/interview/spring-interview/interview-questions-spring-security>
- <https://www.dineshonjava.com/spring-security-interview-questions-and-answers/>
- <https://www.wisdomjobs.com/e-university/spring-security-interview-questions.html>