15 Must-Know Spring MVC Interviev Questions

Spring has become one of the most used Java frameworks for the development of w applications. All the new Java applications are by default using Spring core and Spr frameworks. Thanks to its growing popularity, recruiters all over the globe are lool candidates hands-on with the Spring framework. If you're appearing for an intervi **Java developer role**, Spring MVC is one of the first things that you should brush up irrespective of whether you're a fresher or someone with experience.

In this article, we'll be talking about 15 such Spring MVC must questions which you can expect to encounter in any interview for.

1. What is the Spring framework?

Spring is an open-source framework that was built to simplify application develop a layered structure which allows the developer to be selective about the component It has three main components – Spring Core, Spring AOP, and Spring MVC.

Further, you can talk about your experience with Spring, if any. That'll add a lot of weigh answer.

2. What are the main features of Spring framework?

Spring framework offers a lot of features to make the developer's life easy. Some o

- **Lightweight:** Spring is extremely lightweight, the basic version is around 1MB, v negligible processing overheads.
- **Inversion of Control (IoC):** Dependency Injection or Inversion of Control is one important features of Spring. Using IoC, the developers don't need to create the environment for the object and its dependencies; they can simply create and tes they are handling at the given point of time. Object dependencies will be include upon when the need arises.
- **Aspect Oriented Programming:** Spring supports Aspect Oriented Programming. secondary functions from the programmer's business logic. This not only provi

modularity but also makes the code maintainable.

- **MVC architecture:** Spring comes with an MVC framework for web-applications. framework is highly configurable using various technologies like JSP, Tiles, iTex
- **JDBC exception handling:** Spring comes with a predefined JDBC abstraction laye simplifies the overall exception handling process.

3. Explain a bit more about Dependency Injection.

Inversion of Control or Dependency Injection aims to simplify the process of objec following a simple concept – don't create objects, just describe how they should be Using IoC, the objects are given their dependencies at build-time by an external en responsible for coordinating each object in the system. In essence, we're injecting dependencies into objects using IOC or Dependency Injection.

4. Explain the different types of Dependency Injections in Spring? When t which?

Spring provides the developers with the following two types of dependency injectio

- **Constructor-based DI:** Constructor-based DI is accomplished by passing a num arguments (each of which represents a dependency on other class) to a class's c Simply, dependencies are given in the form of constructor parameters.
- **Setter-based DI:** When you are working with a no-argument constructor, you w by passing arguments through setter function to instantiate the bean under con this is called setter-based dependency injection.

When will you use which one of these, boils down to your requirements. However, recommended to use Setter-based DI for optional dependencies and Constructor-b mandatory dependencies.

5. What is the Spring MVC framework?

Spring MVC is one of the core components of the Spring framework. It comes with components and elements that help developers build flexible and robust web appli the name suggests, the MVC architecture separates the different aspects of the app input logic, business logic, and UI logic. It also provides a loose coupling between t C of the application.

6. What are some benefits of Spring MVC framework over other MVC fra

The Spring MVC framework has some clear benefits over other frameworks. Some
benefits are:

- **Clear separation of roles** – There is a specialised object for every role, thus prov
  separation of roles.
- **Reusable business code** – With Spring MVC, you don't need to duplicate your co
  use your existing objects as commands instead of mirroring them in order to ext
  particular framework base class.
- Customizable binding and validation
- Customizable locale and theme resolution
- Customizable handler mapping and view resolution
- From Spring 2.0 onwards, the framework comes with a JSP form tag library whic
  writing forms in JSP pages much easier.

7. What is DispatcherServlet?

Spring MVC framework is request-driven and is designed around a central Servlet
all the HTTP requests and responses. The DispatcherServlet, however, does a lot m
that. It seamlessly integrates with the IoC container and allows you to use each fea
Spring.

On receiving an HTTP request, the DispatcherServlet consults HandlerMapping (th
configuration files) to call the appropriate Controller. Then, the controller calls ap
service methods to set the Model data. It also returns the view name to Dispatcher
DispatcherServlet, with the help of ViewResolver, picks up the defined view for the
Once the view is finalised, the DispatcherServlet passes the Model data to View – w
finally rendered on the browser.

8. What is the front controller class of the Spring MVC?

A front controller is a controller which handles all requests for a Web application. W
comes to Spring MVC, DispatcherServlet is that front controller. When a web reque
a Spring MVC application, the DIspatcherServlet takes care of everything. First, it t

request. Then, it organises the different components like request handlers, control resolvers, and such – all needed to handle the request. And finally, it renders the co the browser.

9. What is a Viewresolver pattern and how does it work in MVC?

View Resolver is a J2EE pattern which allows the applications to dynamically choos technology for rendering the data on the browser (View). Any technology like HTM Tapestry, XSLT, JSF, or any other such technology can be used for View. The View R pattern holds the mapping of different views. The Controller returns the name of th which is then passed to View Resolver for selecting the appropriate technology.

10. How does Spring MVC provide validation support?

Spring primarily supports two types of validations:

- Using JSR-303 Annotations and any reference implementation, for example, Hib Validator, or
- Implementing org.springframework.validation.Validator interface.

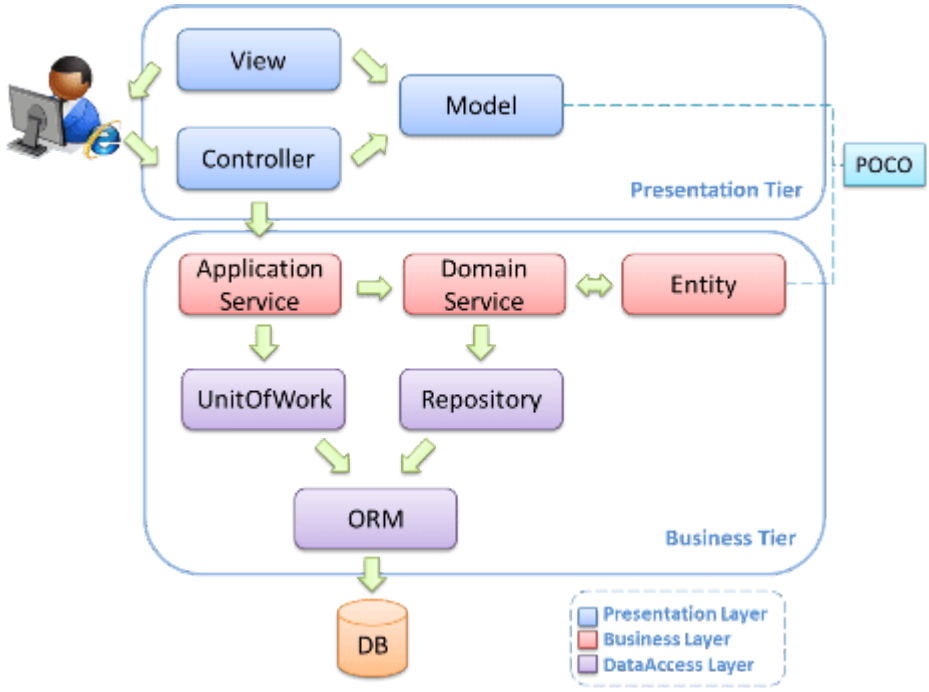
11. A user gets a validation error in other fields on checking a checkbox, af he unchecks it. What would be the current selection status in command o Spring MVC? How will you fix this issue?

This is one of the trickier questions to answer if you aren't aware of the HTTP Post in Spring MVC.

During HTTP Post, if you uncheck the checkbox, then HTTP does not include a requ parameter for the checkbox – which means the updated selection won't be picked that, you can use hidden form field which starts with '_'.

12. How will you compare the MVC framework to the three-tier architect

A Three-tier architecture is an architecture style whereas MVC is a design pattern.



Having said that, in larger applications, MVC forms the presentation tier of a three architecture. The Model, View, and Controller are concerned only with the presenta use the middle tier to populate their models.

13. How should we use JDBC in Spring to optimise the performance?

Spring provides a template class called as JDBCTemplate. Using JDBC with this tem manifolds better performance.

14. What do you mean by a "Bean" in the context of Spring framework?

Any class that is initialised by the IoC container is known as a bean in Spring. The l Spring Bean is managed by Spring IoC Container.

15. What is a "Scope" in reference to Spring Beans?

Spring Beans comes with following five scopes:

- **Prototype**: Whenever there's a request for a bean, a separate prototype is create
- **Request**: It is like the previous scope, but only for web-based applications. For e request, Spring creates a new bean instance.
- **Singleton**: There's only one bean created for every container, and it acts as the d of that bean. In all these instances, the beans cannot use a shared instance varia lead to data-inconsistency.