Spring MVC Interview Questions

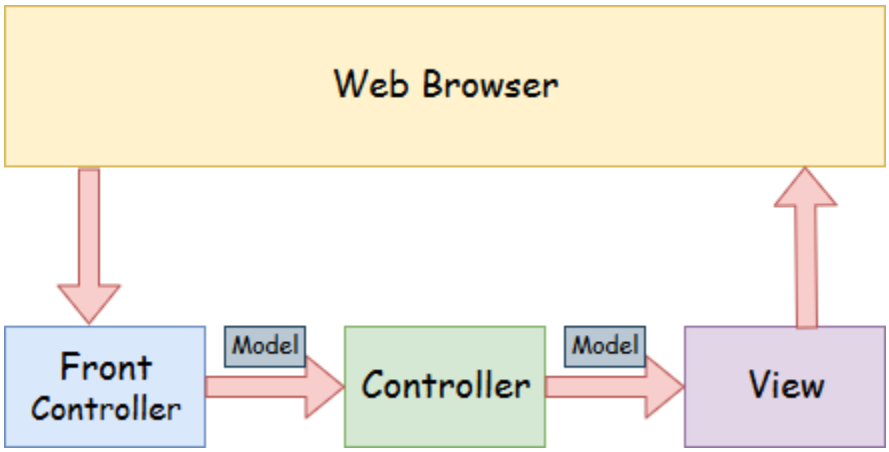
1) What is MVC?

The MVC (Model-View-Controller) is a software architectural design pattern. It separates the functionality of an application into three interconnected parts - Model, View, and Controller. This approach facilitates the reusability of the code and parallel development.

---

2) What is Spring MVC?

A Spring MVC is a Java Framework which is used to develop dynamic web applications. It implements all the basic features of a core spring framework like Inversion of Control and Dependency Injection. It follows the Model-View-Controller design pattern.



Here,

- **Model** - A model contains the data of the application. Data can be a single object or a collection of objects.
- **Controller** - A controller contains the business logic of an application. Here, the @Controller annotation is used to mark the class as the controller.
- **View** - A view represents the provided information in a particular format. So, we can create a view page by using view technologies like JSP+JSTL, Apache Velocity, Thymeleaf, and FreeMarker.

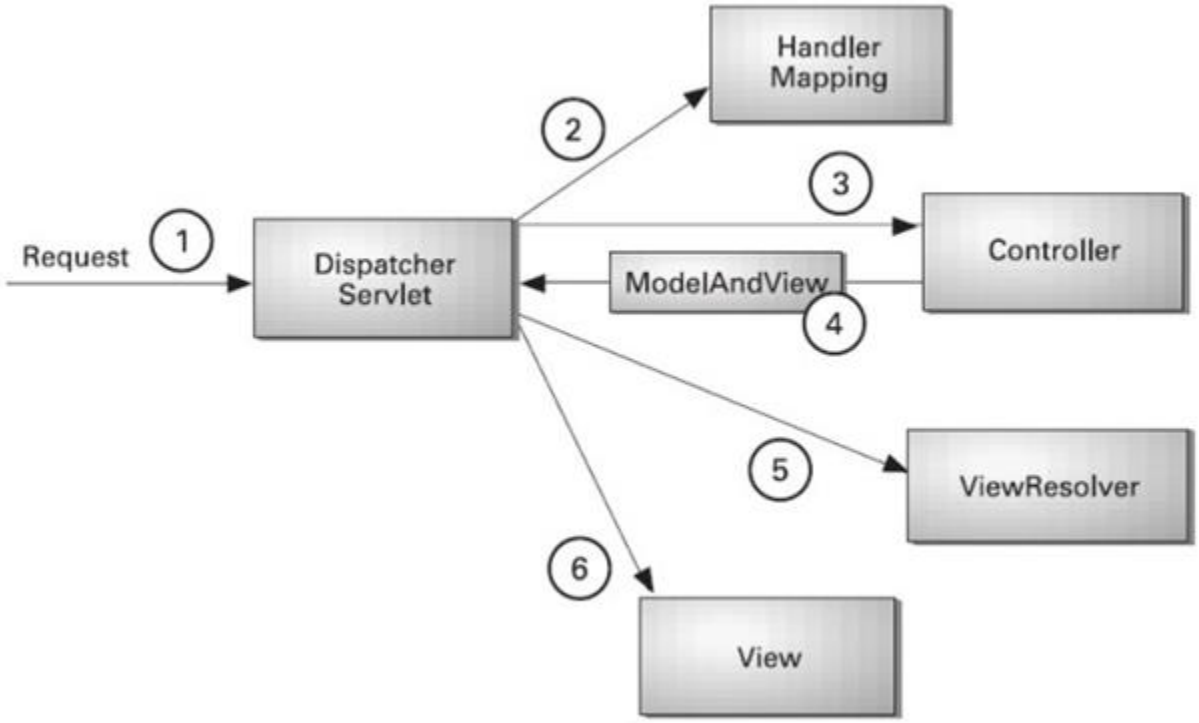
3) What is the front controller of Spring MVC?

The front controller is a **DispatcherServlet** class present in **org.springframework.web.servlet** package. It dispatches the request to the

appropriate controller and manages the flow of the application. It is required to specify the **DispatcherServlet** class in the web.xml file.

---

4) Explain the flow of Spring MVC?



- **Once the request has been generated, it intercepted by the DispatcherServlet that works as the front controller.**
- **The DispatcherServlet gets an entry of handler mapping from the XML file and forwards the request to the controller.**
- **The controller returns an object of ModelAndView.**
- **The DispatcherServlet checks the entry of view resolver in the XML file and invokes the specified view component.**

---

5) What are the advantages of Spring MVC Framework?

The following are the advantages of Spring MVC Framework : -

- Separate roles **- The Spring MVC separates the application into three interconnected layers where each layer has its role.**
- Light-weight **- It uses light-weight servlet container to develop and deploy your application.**

- Powerful Configuration **- It provides a robust configuration for both framework and application classes that includes easy referencing across contexts, such as from web controllers to business objects and validators.**
- Rapid development **- The Spring MVC facilitates fast and parallel development.**
- Reusable business code **- Instead of creating new objects, it allows us to use the existing business objects.**
- Flexible Mapping **- It provides the specific annotations that easily redirect the page.**

---

6) What does an additional configuration file contain in Spring MVC application?

The Spring MVC application contains an additional configuration file that contains the properties information. This file can be created either in the form of **an xml** file or **properties** file. In this file, we generally define the base-package and view resolver where **DispatcherServlet** searches for the controller classes and view components path. However, it can also contain various other configuration properties.

---

7) What is an InternalResourceViewResolver in Spring MVC?

The **InternalResourceViewResolver** is a class which is used to resolve internal view in Spring MVC. Here, you can define the properties like prefix and suffix where prefix contains the location of view page and suffix contains the extension of view page. For example:-

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

---

8) How to declare a class as a controller class in Spring MVC?

The @Controller annotation is used to declare a class as a controller class. It is required to specify this annotation on the class name. For example:-

```
1. @Controller
2. class Demo
```

3. {
4.
5. }

---

9) How to map controller class and its methods with URL?

The **@RequestMapping** annotation is used to map the controller class and its methods. You can specify this annotation on the class name as well as method name with a particular URL that represents the path of the requested page. For example:-

1. @Controller
2. **@RequestMapping("/ form")**
3. class Demo
4. **{**
5. @RequestMapping("/show")
6. **public String display()**
7. {
8.
9. }
10.
11. }

---

10) Name the annotations used to handle different types of incoming HTTP request methods?

The following annotations are used to handle different types of incoming HTTP request methods: -

- **@GetMapping**
- **@PostMapping**
- **@PutMapping**
- **@PatchMapping**
- **@DeleteMapping**

---

11) What is the purpose of @PathVariable annotation in Spring MVC?

The @PathVariable annotation is used to extract the value of the URI template. It is passed within the parameters of the handler method.

For example :-

```
1. @RequestMapping("/show/{id}")
2. public String handler(@PathVariable("id") String s, Model map)
3.  {
4. }
```

---

12) What is the role of @ResponseBody annotation in Spring MVC?

The @ResponseBody annotation is used to serialize the returned object automatically in JSON and bind it with the Http response body. Here, it not required to invoke the model.

For example :-

```
1. @RequestMapping("/show")
2.     @ResponseBody
3.     public ResponseHandler display(
4.       @RequestBody ShowForm form) {
5.         return new ResponseHandler("display form");
6.      }
7. }
```

---

13) What is the role of the Model interface in Spring MVC?

The **Model** interface works as a container that contains the data of the application. Here, data can be in any form such as objects, strings, information from the database, etc.

click here for more details

---

14) What do you mean by ModelAndView in Spring MVC?

The **ModelAndView** is a class that holds both Model and View where the model represents the data, and view represents the representation of that data. This class returns the model and view in the single return value.

---

15) What is ModelMap in Spring MVC?

The **ModelMap** is a class that provides the implementation of Map. It extends the LinkedHashMap class. It facilitates to pass a collection of values as if they were within a Map.

---

16) What are the ways of reading data from the form in Spring MVC?

The following ways to read the data from the form are: -

- HttpServletRequest interface **- The** HttpServletRequest **is a java interface present in javax.servlet.http package. Like Servlets, you can use HttpServletRequest in Spring to read the HTML form data provided by the user.**
- @RequestParam annotation **- The** @RequestParam **annotation reads the form data and binds it automatically to the parameter present in the provided method.**
- @ModelAttribute annotation **- The** @ModelAttribute **annotation binds a method parameter or its return value to a named model attribute.**

---

17) What is Spring MVC form tag library?

The Spring MVC form tags can be seen as data binding-aware tags that can automatically set data to Java object/bean and also retrieve from it. These tags are the configurable and reusable building blocks for a web page. It provides view technologies, an easy way to develop, read, and maintain the data.

click here for more details

---

18) What do you understand by validations in Spring MVC?

The validation is one of the most important features of Spring MVC, that is used to restrict the input provided by the user. To validate the user's input, it is required to use the Spring 4 or higher version and Bean Validation API. Spring validations can validate both server-side as well as client-side applications.

---

19) What is Bean Validation API?

The **Bean Validation API** is a Java specification which is used to apply constraints on object model via annotations. Here, we can validate a length, number, regular expression, etc. Apart from that, we can also provide custom validations.

As Bean Validation API is just a specification, it requires an implementation. So, for that, it uses Hibernate Validator. The Hibernate Validator is a fully compliant JSR-303/309 implementation that allows to express and validate application constraints.

---

20) What is the use of @Valid annotation in Spring MVC?

The **@Valid** annotation is used to apply validation rules on the provided object.

---

21) What is the purpose of BindingResult in Spring MVC validations?

The **BindingResult** is an interface that contains the information of validations. For example :-

```
1. @RequestMapping("/helloagain")
2.     public String submitForm( @Valid @ModelAttribute("emp") Employee e,
   BindingResult br)
3.     {
4.         if(br.hasErrors())
5.         {
6.             return "viewpage";
7.         }
8.         else
9.         {
10.        return "final";
```

```
11.     }
12.  }
```

22) How to validate user's input within a number range in Spring MVC?

In Spring MVC Validation, we can validate the user's input within a number range by using the following annotations: -

- @Min annotation **- It is required to pass an integer value with @Min annotation. The user input must be equal to or greater than this value.**
- @Max annotation **- It is required to pass an integer value with @Max annotation. The user input must be equal to or smaller than this value.**

click here for more details

23) How to validate the user input in a particular sequence in Spring MVC?

The Spring MVC Validation allows us to validate the user input in a particular sequence by using @Pattern annotation. Here, we can provide the required regular expression to **regexp** attribute and pass it with the annotation.

click here for more details

24) What is the purpose of custom validations in Spring MVC?

The Spring MVC framework allows us to perform custom validations. In such a case, we declare our own annotations. We can perform validation based on own business logic.

click here for more details

25) What do you understand by Spring MVC Tiles?

The Spring provides integration support with apache tiles framework. So we can manage the layout of the Spring MVC application with the help of spring tiles support. The following are the advantages of Tiles support in Spring MVC: -

- Reusability: **We can reuse a single component in multiple pages like header and footer components.**
- Centralized control: **We can control the layout of the page by a single template page only.**
- Easy to change the layout: **By the help of a single template page, we can change the layout of the page anytime. So your website can easily adopt new technologies such as bootstrap and jQuery.**

26) What is the front controller class of Spring MVC?

The **DispatcherServlet** class works as the front controller in Spring MVC.

More details...

---

27) What does @Controller annotation?

The **@Controller** annotation marks the class as controller class. It is applied on the class.

---

28) What does @RequestMapping annotation?

The **@RequestMapping** annotation maps the request with the method. It is applied on the method.

---

29) What does the ViewResolver class?

The **View Resolver** class resolves the view component to be invoked for the request. It defines prefix and suffix properties to resolve the view component.

---

30) Which ViewResolver class is widely used?

The **org.springframework.web.servlet.view.InternalResourceViewResolver** class is widely used.

---

31) Does spring MVC provide validation support?

Yes.