

Assignment Questions 10

Q1.What is the Spring MVC framework?

ans::- Spring MVC Framework follows the Model-View-Controller design pattern. It is used to develop web applications. It works around DispatcherServlet.

DispatcherServlet handles all the HTTP requests and responses. It dispatches the requests to handlers. It uses @Controller and @RequestMapping as default request handlers. The @Controller annotation defines that a particular class is a controller.

@RequestMapping annotation maps web requests to Spring Controller methods. The terms model, view, and controller are as follows:

Model: The Model encapsulates the application data.

View: View renders the model data and also generates HTML output that the client's browser can interpret.

Controller: The Controller processes the user requests and passes them to the view for rendering.

Q2.What are the benefits of Spring MVC framework over other MVC frameworks?

ans::- Spring MVC Framework follows the Model-View-Controller design pattern. It is used to develop web applications. It works around DispatcherServlet. DispatcherServlet handles all the HTTP requests and responses. It dispatches the requests to handlers.

It uses @Controller and @RequestMapping as default request handlers. The @Controller annotation defines that a particular class is a controller. @RequestMapping annotation maps web requests to Spring Controller methods. The terms model, view, and controller are as follows:

Model: The Model encapsulates the application data.

View: View renders the model data and also generates HTML output that the client's browser can interpret.

Controller: The Controller processes the user requests and passes them to the view for rendering.

Spring MVC Framework works as follows:

All the incoming requests are intercepted by the DispatcherServlet that works as the front controller.

The DispatcherServlet then gets an entry of handler mapping from the XML file and forwards the request to the controller.

The object of ModelAndView is returned by the controller.

The DispatcherServlet checks the entry of the view resolver in the XML file and invokes the appropriate view component.

Advantages of Spring MVC Framework

The container is used for the development and deployment of applications and uses a lightweight servlet.

It enables rapid and parallel development.

Development of the application becomes fast.

Easy for multiple developers to work together.

Easier to Update the application.

It is Easier to Debug because we have multiple levels in the application.

Q3.What is DispatcherServlet in Spring MVC? In other words, can you explain the Spring MVC architecture?

ans:- Spring is one of the most popular Java EE frameworks. It is an open-source lightweight framework that allows Java EE 7 developers to build simple,

reliable, and scalable enterprise applications. This framework mainly focuses on providing various ways to help you manage your business objects.

It made the development of Web applications much easier than compared to classic Java frameworks and application programming interfaces (APIs),

such as Java database connectivity (JDBC), JavaServer Pages(JSP), and Java Servlet. This framework uses various new techniques such as Aspect-Oriented Programming (AOP),

Plain Old Java Object (POJO), and dependency injection (DI), to develop enterprise applications. Now we are going to discuss one of the most important concepts in the Spring framework

e.g Dispatcher Servlet.

DispatcherServlet acts as the Front Controller for Spring-based web applications. So now what is Front Controller? So it is pretty simple.

Any request is going to come into our website the front controller is going to stand in front and is going to accept all the requests and once the front controller accepts that request then this is the job of the front controller that it will make a decision that who is the right controller to handle that request.

For example, refer to the below image. Suppose we have a website called student.com and the client is make a request to save student data by hitting the following URL student.com/save and its first come to the front controller and once the front controller accepts that request it is going to assign to the Controller_1 as this controller handle the request for /save operation.

Then it is going to return back the response to the Client.

Q4.What is a View Resolver pattern and explain its significance in Spring MVC?

ans:- View Resolver pattern is a J2EE pattern that enables a web application to dynamically select its view technology. For example, HTML, JSP, Tapestry, JSF, XSLT etc. In this pattern, View resolver holds mapping of different views, controller return name of the view, which is then delegated to the View Resolver for selecting an appropriate view. Spring MVC framework supplies inbuilt view resolver for selecting views.

Q5.What are the differences between @RequestParam and @PathVariable annotations?

ans:-

As the name suggest, In Spring MVC, @RequestParam is an annotation used to bind request parameters from a URL or a form submission to method parameters in a Spring MVC controller.

It allows developers to retrieve values from query parameters, form data, or request headers and use them in their controller methods.

Here's an example of using @RequestParam in a Spring MVC controller method:

```
@RequestMapping("/user")
public String getUser(@RequestParam("id") int userId) {
    // Logic to fetch user details by userId
    // ...
    return "user";
}
```

The @RequestParam annotation in Spring MVC is typically used when you want to retrieve values from query parameters, form data, or request headers in a Spring MVC controller method.

The `@PathVariable` annotation in Spring MVC is used to capture and bind a value from a URL path segment to a method parameter in a controller method. It allows you to extract dynamic values from the URL path and use them in your controller logic. Here's an example:

Suppose you have a URL pattern like `"/users/{id}"` where `"{id}"` is a path variable representing the user ID. You can use `@PathVariable` to capture the value of `"{id}"` and bind it to a method parameter in your controller method. Here's an example:

```
@GetMapping("/users/{id}")

public ResponseEntity<User> getUserById(@PathVariable("id") Long userId) {

    // Logic to retrieve user by ID

    User user = userService.getUserById(userId);

    if (user != null) {

        return ResponseEntity.ok(user);

    } else {

        return ResponseEntity.notFound().build();

    }

}
```

Q6.What is the Model in Spring MVC?

ans:-

In Spring MVC, the model works as a container that contains the data of the application. Here, a data can be in any form such as objects, strings, information from the database, etc.

It is required to place the Model interface in the controller part of the application. The object of `HttpServletRequest` reads the information provided by the user and pass it to the Model interface. Now, a view page easily accesses the data from the model part.

Q7.What is the role of `@ModelAttribute` annotation?

ans:- In Spring MVC, the `@ModelAttribute` annotation binds a method parameter or method return value to a named model attribute and then exposes it to a web view. It refers to the property of the Model object.

For example, if we have a form with a form backing object that is called “Student” then we can have Spring MVC supply this object to a Controller method by using the @ModelAttribute annotation:

```
@RequestMapping("/home")
```

```
public String showHomePage(@ModelAttribute("studentInfo") StudentInfoDTO studentInfoDTO) {
```

```
    return "something";
```

```
}
```

Q8.What is the significance of @Repository annotation?

ans:- Spring is one of the most popular Java EE frameworks. It is an open-source lightweight framework that allows Java EE 7 developers to build simple,

reliable, and scalable enterprise applications. This framework mainly focuses on providing various ways to help you manage your business objects. It made the development of Web applications much easier than compared to classic Java frameworks and application programming interfaces (APIs),

such as Java database connectivity (JDBC), JavaServer Pages(JSP), and Java Servlet. This framework uses various new techniques such as Aspect-Oriented Programming (AOP), Plain Old Java Object (POJO), and dependency injection (DI), to develop enterprise applications. Now talking about Spring Annotation

Spring Annotations are a form of metadata that provides data about a program. Annotations are used to provide supplemental information about a program. It does not have a direct effect on the operation of the code they annotate. It does not change the action of the compiled program.

Q9.What does REST stand for? and what is RESTful web services?

ans:- REST (representational state transfer) refers to a set of specific rules that dictate how web service applications communicate over the internet. Therefore, a RESTful API is an API that follows these rules, providing flexibility, bandwidth, and speed advantages to software integrations and communications.

REST APIs serve as the middleman between your client and your host server, fielding resource requests and delivering responses via HTTP. In a REST design, the client initiates the interaction — for example, accessing a webpage via a URL is a commonly used HTTP request.

Q10.What is differences between RESTful web services and SOAP web services?

ans:-

SOAP

SOAP stands for Simple Object Access Protocol

SOAP is a protocol. SOAP was designed with a specification. It includes a WSDL file which has the required information on what the web service does in addition to the location of the web service.

SOAP cannot make use of REST since SOAP is a protocol and REST is an architectural pattern.

SOAP uses service interfaces to expose its functionality to client applications. In SOAP, the WSDL file provides the client with the necessary information which can be used to understand what services the web service can offer.

SOAP requires more bandwidth for its usage. Since SOAP Messages contain a lot of information inside of it, the amount of data transfer using SOAP is generally a lot.

SOAP can only work with XML format. As seen from SOAP messages, all data passed is in XML format.

REST

REST stands for Representational State Transfer

REST is an Architectural style in which a web service can only be treated as a RESTful service if it follows the constraints of being

Client Server

Stateless

Cacheable

Layered System

Uniform Interface

REST can make use of SOAP as the underlying protocol for web services, because in the end it is just an architectural pattern.

REST use Uniform Service locators to access to the components on the hardware device. For example, if there is an object which represents the data of an employee hosted on a URL as <http://demo.guru99> , the below are some of URI that can exist to access them.

REST does not need much bandwidth when requests are sent to the server. REST messages mostly just consist of JSON messages. Below is an example of a JSON message passed to a web server. You can see that the size of the message is comparatively smaller to SOAP.

REST permits different data format such as Plain text, HTML, XML, JSON, etc. But the most preferred format for transferring data is JSON.