

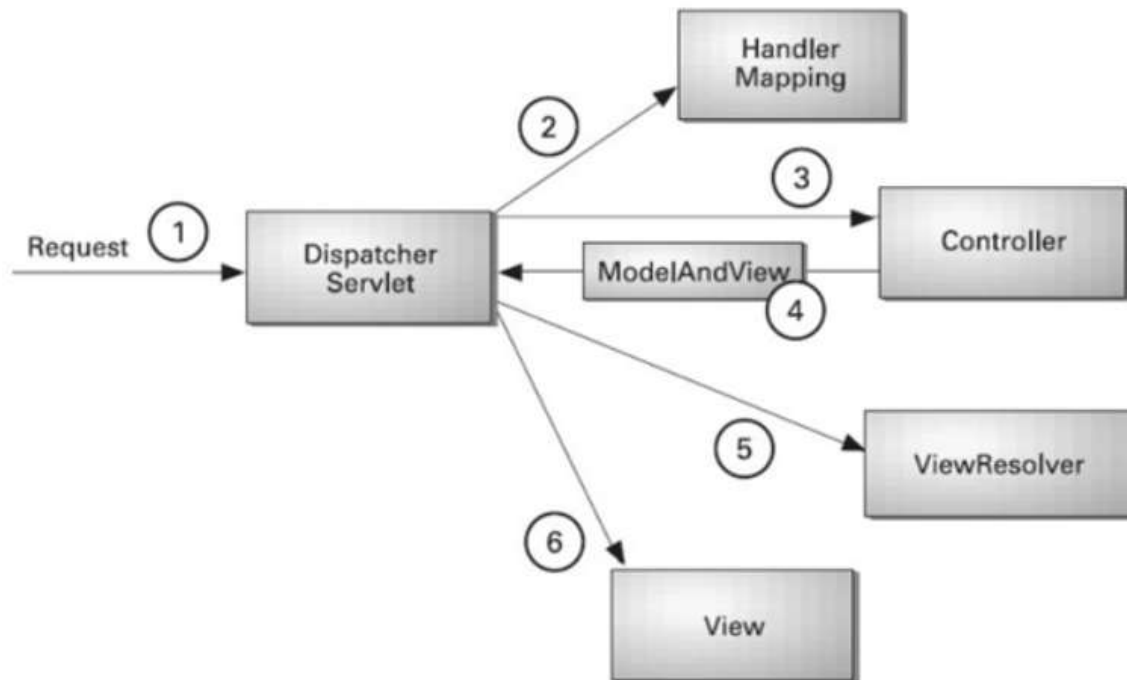
PPT JAVA Assignment-10

Q1.What is the Spring MVC framework?

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

A Spring MVC provides an elegant solution to use MVC in spring framework by the help of **DispatcherServlet**. Here, **DispatcherServlet** is a class that receives the incoming request and maps it to the right resource such as controllers, models, and views.

- **Model** - A model contains the data of the application. A 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. Generally, JSP+JSTL is used to create a view page. Although spring also supports other view technologies such as Apache Velocity, Thymeleaf and FreeMarker.
- **Front Controller** - In Spring Web MVC, the DispatcherServlet class works as the front controller. It is responsible to manage the flow of the Spring MVC application.



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

- *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.

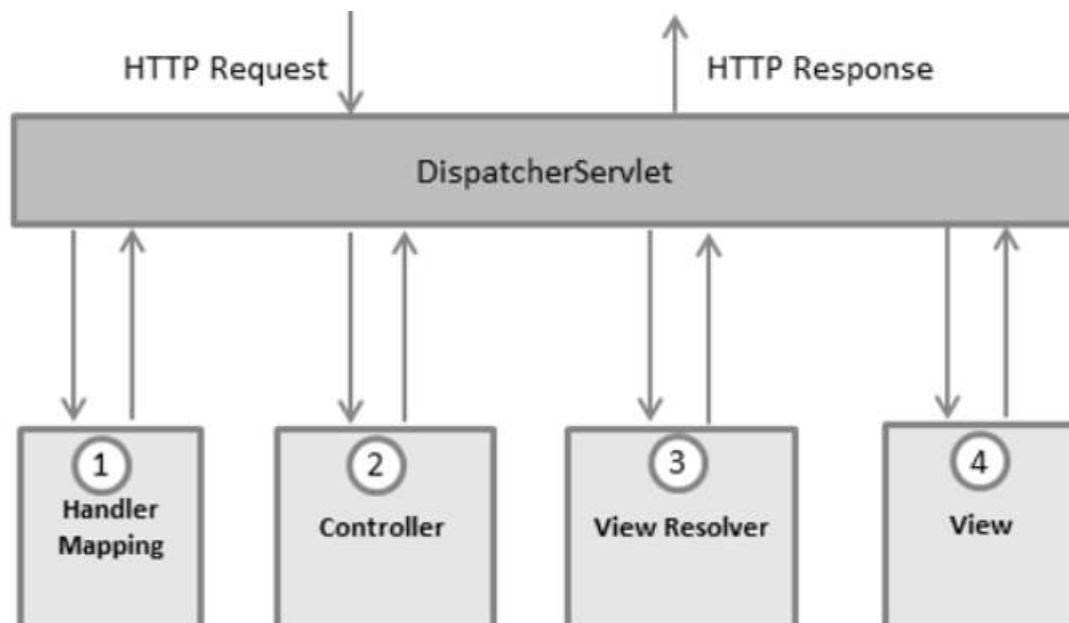
Disadvantages of Spring MVC Framework

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

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.

The Spring Web MVC framework provides Model-View-Controller (MVC) architecture and ready 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 (input logic, business logic, and UI logic), 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.



Q.4 What is a View Resolver pattern and explain its significance in Spring MVC?

- It is a J2EE pattern that allows the applications to dynamically choose technology for rendering the data on the browser (View).
 - Any technology like HTML, JSP, XSLT, JSF, or any other such technology can be used as View.
- The View Resolver has the information of different views. The Controller returns the name of the View which is then passed to

View Resolver by the DispatcherServlet for selecting the appropriate View technology and then the data is displayed.

- The default ViewResolver used in Spring MVC is InternalResourceViewResolver.

Q.5 What are the differences between @RequestParam and @PathVariable annotations?

Feature	@RequestParam	@PathVariable
Usage	Query parameters	Path variables
Mapping	Maps to request parameters with matching names	Maps to URI template variables
Position	Can be placed anywhere in the method parameter list	Must be placed directly on the method parameter
Required	Can be optional or required (default is required)	Always required
Binding	Optional binding to a default value if not present	Binds directly to the URI template variable
URL Encoding	Required for special characters in parameter values	Automatically decoded
Example	<pre>@RequestParam("paramName") String paramValue</pre>	<pre>@PathVariable("varName") String varValue</pre>

Q6.What is the Model in Spring MVC?

In Spring MVC, the model works 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?

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.

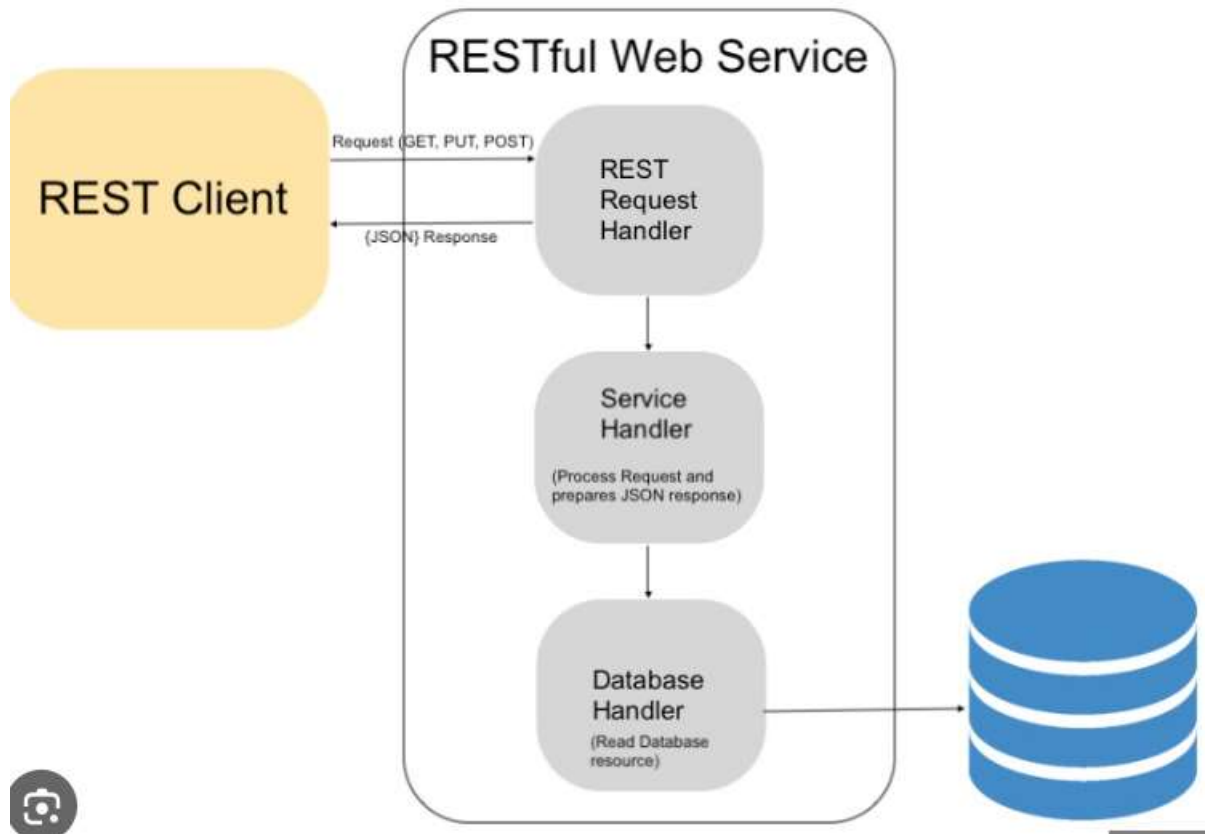
Q8.What is the significance of @Repository annotation?

Spring @Repository annotation is used to indicate that the class provides the mechanism for storage, retrieval, search, update and delete operation on objects.

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

REST, or REpresentational State Transfer, is an architectural style for providing standards between computer systems on the web, making it easier for systems to communicate with each other. REST-compliant systems, often called RESTful systems, are characterized by how they are stateless and separate the concerns of client and server.

RESTful Web Services are basically REST Architecture based Web Services. In REST Architecture everything is a resource. RESTful web services are light weight, highly scalable and maintainable and are very commonly used to create APIs for web-based applications.



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

SOAP Protocol	RESTful Web Services
SOAP is a protocol.	REST is an architectural approach.
SOAP acronym for Simple Object Access Protocol.	REST acronym for REpresentational State Transfer.
In SOAP, the data exchange format is always XML.	There is no strict data exchange format. We can use JSON, XML, etc.
XML is the most popular data exchange format in SOAP web services.	JSON is the most popular data exchange format in RESTful web services.
SOAP uses Web Service Definition Language (WSDL).	REST does not have any standard definition language.
SOAP does not pose any restrictions on transport. We can use either HTTP or MQ.	RESTful services use the most popular HTTP protocol.
SOAP web services are typical to implement.	RESTful services are easier to implement than SOAP.
SOAP web services use the JAX-WS API.	RESTful web services use the JAX-RS API.
SOAP protocol defines too many standards.	RESTful services do not emphasis on too many standards.
SOAP cannot use RESTful services because it is a protocol.	RESTful service can use SOAP web services because it is an architectural approach that can use any protocol like HTTP and SOAP.
SOAP reads cannot be cached.	REST reads can be cached.