1) What is Spring WEB Module ?

This module is given by Spring Framework.

It is used to develop both MVC and REST applications.

It works based on HTTP Protocol Standard.

It is used to develop client-server and server-server apps.

2) 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.

3) What is Spring MVC?

Java Framework which is used to develop dynamic web applications.

It follows the Model-View-Controller design pattern.

Model - A model contains the data of the application.

Controller - A controller contains the business logic of an application.

View - A view represents the provided information in a particular format. Like JSP+JSTL, Apache Velocity, Thymeleaf, and FreeMarker.

4) 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.

5) 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.

6) What are the advantages of Spring MVC Framework?

The following are the advantages of Spring MVC Framework:

Light-weight - It uses light-weight servlet container to develop and deploy your application.

Rapid development - The Spring MVC facilitates fast and parallel development.

Flexible Mapping - It provides the specific annotations that easily redirect the page.

7) Which annotations are used for HTTP request methods?

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

@GetMapping

@PostMapping

@PutMapping

@PatchMapping

@DeleteMapping

8)Explain the Difference Between @Controller and @RestController?

The main difference between the @Controller and @RestController annotations is that the @ResponseBody annotation is automatically included in the @RestController.

This means that we don't need to annotate our handler methods with the @ResponseBody.

We need to do this in a @Controller class if we want to write response type directly to the HTTP response body.

9) What Are the @RequestBody and the @ResponseBody Annotations?

The @RequestBody annot tion, used as a handler method parameter, binds the HTTP Request body to a transfer or a domain object.

Spring automatically deserializes incoming HTTP Request to the Java object using Http Message Converters.

@ResponseBody annotation indicates that we'll write the return type of the method directly to the HTTP response body

10)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.

```
@Controller
@RequestMapping("/form") class DemoController
{
@RequestMapping("/show") public String display() {}
}
```

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.

@RequestMapping("/show/{id}") public String handler(@PathVariable("id") String uid) { }

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.

For Example:-

```
@RequestMapping("/show")
@ResponseBody
public ProductPayload disply()
{ return new ProductPayload(id,code, amount);
}
```

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

It 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: @NotNull, @Pattern, @Email, @Past, @NotBlank ..etc

@Valid must be used at Method Parameter along with @RequestBody or @ModelAttribute

14) Which annotations are used to define Global Exception Handler class?

@ControllerAdvice (or) @RestControllerAdvice along with @ExceptionHandler

We need to define one user class with annotation @ControllerAdvice or @
RestControllerAdvice

Then Define one method and @ExceptionHandler with Exception Type.

15) Explain some Response Status codes ?

200 OK indicates Success

400 Bad Request - Input data is invalid

401-UnAuthorized - Login Request is failed

Forbidden - User not allowed to access Resource

Not Found URL is not exist

405-Method Not Allowed - Request Method is invalid 500-Internal server Error.

16) What is cross-origin concept in Spring WEB?

Cross-Origin Resource Sh ring (CORS) is a security concept that allows restricting the resources implemented in web browsers.

It prevents the JavaScript code producing or consuming the requests against different origin.

If your apps needs to be connected using JavaScript apps like Angular or ReactJS, then we need to allow those by adding this annotation • @CrossOrigin(origins = "http://localhost:8080")

17) What is init-binder in Spring WEB?

@InitBinder: This annotation is used with the methods which initializes WebDataBinder and works as a preprocessor for each request coming to the controller.

Like String to Date Format Convertsions.

Even String to any Special ClassType conversions.

18) Explain @PathVariable @RequestParam Annotations?

With the @PathVariable annotation, we bind the request URL template path variable to the method variable.

Ex: http://localhost:8080/hello/100/Raghu the 100 value is bound to the id variable and the "Raghu" value to the name variable.

With @RequestParam annotation, we can extract values from the query string. For example http://localhost:8080/helloworld?id=100&name=Raghu

While @RequestParams extract values from the query string, @PathVariables extract values from the URI path.

19) Why do we need View Resolver in Spring MVC?

View Resolver is responsible for rendering of models in the web browser.

The "InternalResourceViewResolver" is the internal view resolver in the Spring MVC.

There are some important view resolver's in Spring MVC as mentioned below:

AbstractCachingViewResolver, XmlViewResolver, UrlBasedViewResolver, ..etc

20) What is the use of @ModelAttribute annotation?

It binds a method parameter or a method return value to a named model attribute and then exposes it to a web view.

If a HTML Form is used and submitted with data then it is converted into one class object (like Employee object) and that can be read into code using @ModelAttribute.

Syntax: @ModelAttribute("objName") ClassName localVariable

21) Is @Controller and @RestController are stereotypes?

Yes, both @Controller and @RestController are stereotypes. The @Controller is actually a specialization of Spring's @Component stereotype annotation.

@RestController is a specialization of @Controller for the RESTful web service.

It not only combines the @ResponseBody and @Controller annotations, but it also gives more meaning to your controller class to clearly indicate that it deals with RESTful requests.

22) What is the main difference between Spring core and Spring MVC?

Spring MVC is built using Spring core.

Spring core is an IOC container that injects dependencies into various bean classes.

Spring MVC leverages the power of that IOC container to implement web based applications.

Spring container concepts like: Container System, Beans, Scheduling, AOP ..etc

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

HttpServletRequest interface - HttpServletRequest is a java interface present in javax.servlet.http package.

@RequestParam annotation - The @RequestParam annotation reads the form data and binds it automatically to the parameter.

@ModelAttribute annotation - The @ModelAttribute annotation binds a method parameter.

24) What is ResponseEntity<T> in Spring REST?

The final output of RestController is ResponseEntity. It must be used for method ReturnType.

It contains Body, Http Headers and Response Code.

It can be converted into JSON, XML formats using @ResponseBody, for non-String return types.

25) Why @RequestHeader is used in Spring WEB?

This annotation is used to read Header Parameters sent by client application

Like Authorization Header, even any custom headers too.

Example:

@RequestHeader("Authorization")String authkey