① What is Spring framework ; why it is popular ?

framework → pre-written code

→ open source, application framework
→ used for enterprise app development
→ lightweight & loosely coupled
→ pre-defined templates - faster development
→ easy to test

② Name some of the Spring Modules ;

Each module serve some purpose

→ Spring Context - for dependency injection
→ Spring MVC - Model-view-controller used for creating web-application, web services etc.
→ Spring Web → used for creating web appl
→ Spring DAO → for database operations &
→ Spring JDBC - for JDBC & Datasource support
→ Spring ORM - for ORM tools support such as Hibernate
→ Spring AOP - for aspect-oriented programming.

③ What is Spring MVC ?

→ It is a module in Spring framework

→ Spring MVC provides Model-view-controller architecture for developing flexible & loosely coupled web applications

Model → Application data [POJO classes]

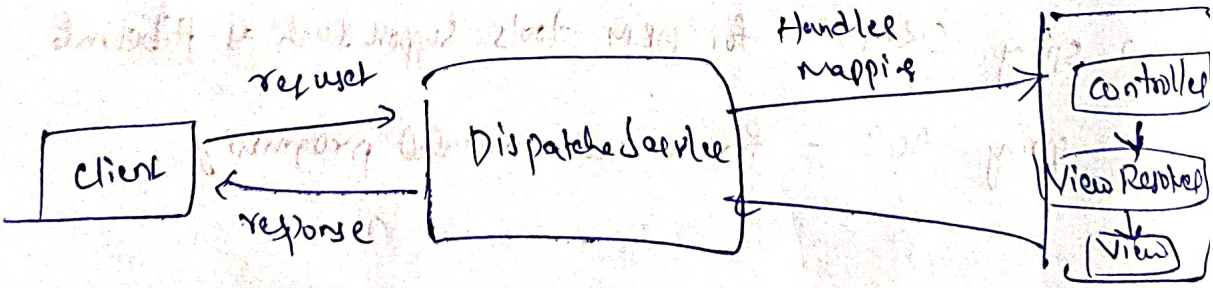
View → HTML page seen by the end user

Controller → Responsible for processing user requests and building model and passes it to the view for rendering.

④ What is DispatcherServlet / front controller ?

→ DispatcherServlet is the front controller that handles all the incoming ~~requests~~ HTTP requests and send back the response to the client

→ The file name shd be [servlet-name]-servlet



③ What is Spring MVC ?

→ It is a module in Spring framework

→ Spring MVC provides model-view-controller architecture for developing flexible & loosely coupled web applications

Model → Application data [POJO classes]

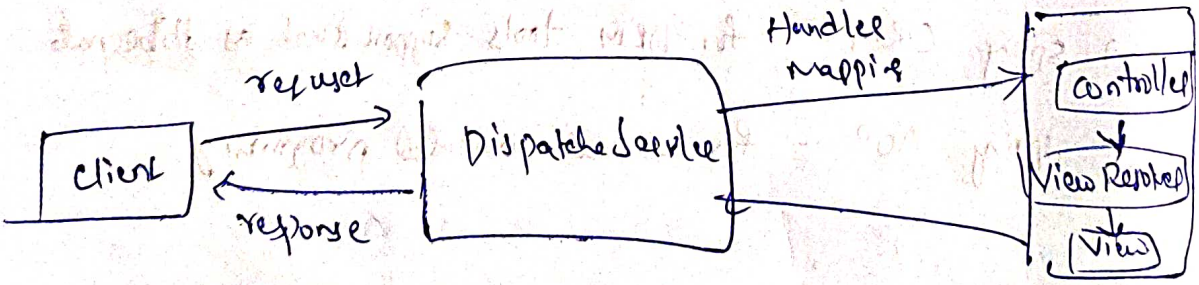
View → HTML page seen by the end user

Controller → Responsible for processing user requests and building model and passes it to the view for rendering.

④ What is DispatcherServlet / front Controller ?

→ DispatcherServlet is the front controller that handles all the incoming ~~requests~~ HTTP requests and send back the response to the client

→ The file name shd be [servlet-name]-servlet



(5) What is Spring IOC container & Explain its adv.

→ The Spring IOC container (Inversion of Control) is at the core of the Spring framework

→ The container will read the configuration file and based on that it will create and store the objects and manages it

Adv

→ Developers can concentrate on business logic since object creation & managing are handled by IOC Containers.

(6) Explain Types of IOC Containers in Spring ?

→ There are two types

① Bean factory : It is an interface that contains collection of beans. It instantiates them whenever asked by clients

② Application Context : It is built on top of the Bean Factory interface. It provides some extra functionality compared to BeanFactory

Application Context is the most used Spring IOC container in Spring.

⑦ How to create Application Content in project ?

→ ApplicationContent is an IOC container and it is an Interface.

→ we need a class for loading the interface

① ClassPathXMLApplicationContext

→ For Standalone java Applications using xml based configuration

② FileSystemXMLApplicationContext

→ Similar to ClassPathXMLApplicationContext but

→ the xml configuration file can be loaded from anywhere in the file system

③ AnnotationConfigApplicationContext

→ for Standalone java applications using annotations based config

④ AnnotationConfigWebApplicationContext and XMLWebApplicationContex for web applications

⑧ What is Spring Bean ?

→ A normal java class that is initialized by Spring IOC container is called Spring Bean

→ The Spring IOC container manages the life cycle of Spring Bean, bean scopes and injecting any required dependencies in the bean.

⑨ How many bean scopes are supported in spring?

Singleton: only one instance will be created for each container (default)

prototype: a new instance will be created every time the bean is requested

Request: for ~~every~~ each HTTP request, a new instance of the bean will be created

Session: for each HTTP session, a new bean will be created

Global session: This is used to create global session beans

To set the bean scope → we can use "scope" attribute
we can use @Scope.

⑩ What is spring Bean life cycle?

→ Spring Beans are created in Container
→ Dependencies are injected
→ Init method is called to initialize the beans
→ Bean is ready to use & operations are done
→ Container shutdown
→ Destroy method is called to destroy the bean

(11) What is spring Configuration file? How many ways we can do that?

→ A spring configuration file is used to define all the beans that will be initialized by spring conte

→ Once the context is initialized, we can use it to get different bean instances

→ We can configure spring beans in below ways

① XML configuration - xml files

② Java Based Configuration - @Configuration

③ Annotation Based Configuration - @Service, @Component ... etc

(12) What do you mean by Dependency Injection?

→ Dependency injection is the process of passing/injecting dependency object to the dependent object

Eg: class Car, class Engine

car - Dependent object Engine - Dependency object

Types of Dependency Injections

① Constructor

② Setter

③ Field

(13) Difference b/w Setter and Constructor

Constructor	Setter
→ Complete dependency injection of the objects	→ partial Dependency injection of the objects
→ order of injecting dependencies is fixed	→ order of injecting dependencies is not fixed
→ It always creates new bean instance	→ It will not create new bean instance

(14) What is Bean wiring and @Autowired Annotation?

→ The process of injecting Spring Bean Dependencies while intializing it called Spring Bean wiring

→ This can be achieved by using @Autowired annotation.

Types of Autowiring

→ Autowire ~~by Type~~ byType
Based on 'class' value in config file

→ autowire byName
Based on 'id' value in config file.

→ autowire by Constructor
added before no arg constructor

(13) Difference b/w Setter and Constructor

Constructor	Setter
→ Complete dependency injection of the objects	→ partial Dependency injection of the objects
→ order of injecting dependencies is fixed	→ order of injecting dependencies is not fixed
→ It always creates new bean instance	→ It will not create new bean instance

(14) What is Bean wiring and @Autowired Annotation?

→ The process of injecting Spring Bean Dependencies while intializing it called Spring Bean wiring

→ This can be achieved by wing @Autowired annotation.

Types of Autowiring

→ Autowire ~~by Type~~ byType
Based on 'class' value in config file

→ autowire byName
Based on 'id' value in config file

→ autowire by Constructor
added before no arg constructor

(15) What is an ~~ViewR~~ ~~InternalView~~ InternalResourceViewResolver in spring mvc ?

→ ViewResolver implementations are used to add prefix and suffix of the view files location.

→ It is configured using spring Bean configuration file

→ InternalResourceViewResolver is one of the implementation of ViewResolver interface

(16) What is the diff blw @Component, @Service @Controller, @Repository annotations in Spring ?

→ @Component : This marks a java class as a bean It is a generic stereotype for any spring-managed component

→ @Controller : This annotation is a specialization of the component annotation and it makes a class as a spring web mvc controller

→ @Service : This annotation is a specialization of the component annotation and it is used in service-layer

→ @Repository : This annotation is a specialization of the component annotation and provides additional benefits specifically for DAO's

(12) What is @ComponentScan annotation in Spring ?

→ The process of scanning spring components within classpath of the application is called component scanning in spring

→ Later spring will use them to add in ApplicationContext (IOC Container)

→ @ComponentScan annotation tells spring that where to look for spring components explicitly

→ @Component, @Controller, @Service, @Repository, @Configuration annotated classes will be treated as spring components

① What is Spring Boot and its advantages

→ Spring Boot is a spring module which provides RAD (Rapid Application Development) feature to Spring framework.

→ It is used to create standalone spring based application that you can just run because it needs very little spring configuration

Adv

→ Auto Configuration: helps to quick start the application

→ Embedded tomcat, jetty, undertow (inbuilt)

→ It provides opinionated starter POMs to simplify the maven configuration

→ Increases productivity & reduces development time.

② Diff b/w Spring vs Spring Boot

Spring	Spring Boot
→ A web application framework based on java	→ spring boot is a module in spring
→ used to create loosely coupled applications	→ used to create spring app project which can just run
→ Requires lot of boiler plate code (more effort for developer)	→ Requires less boilerplate code (reduces develoga effor)

① What is Spring Boot and its advantages

→ Spring Boot is a Spring module which provides RAD (Rapid Application Development) feature to Spring framework.

→ It is used to create standalone spring based application that you can just run because it needs very little spring configuration

Adv

→ Auto Configuration: helps to quick start the application

→ Embedded tomcat, jetty, undertow (inbuilt)

→ It provides opinionated starter POMs to simplify the maven configuration

→ Increases productivity & reduces development time.

② Diff b/w Spring vs Spring Boot

Spring	Spring Boot
→ A web application framework based on java	→ Spring boot is a module in Spring
→ used to create loosely coupled applications	→ used to create Spring app project which can just run
→ Requires lot of boiler plate code (more effort for developer)	→ Requires less boilerplate code (reduces developer effort)

Spring	Spring Boot
→ Developers manually define dependencies for the spring project in pom.xml	→ Spring Boot comes with the concept of starter in pom.xml file (Auto configuration feature)
→ Server should be setup separately	→ It has inbuilt tomcat server in it.

③ What are ~~the~~ some of the features of Spring Boot ?

→ Starter Dependency

→ Auto Configuration

→ Spring initializer

→ Spring Actuator

→ Logging and Security.

④ Mention the minimum requirements for a Spring Boot System

Spring Boot →

Java →

Spring framework →

Build Tool

Maven →

Gradle →

Servlet Containers

tomcat →

Jetty →

Undertow →

1- 300 + 150 + 500

2- 500 + 300 + 300

3) - 150 +

4

5

⑤ How to create Spring boot application using Spring initializer.

→ Spring initializer is a web tool provided by spring. https://start.spring.io/

⑥ How does spring Boot works?

→ Spring Boot automatically configures our application based on the dependencies added to the project in pom.xml file

→ The entry point of the spring boot application is the class that contains @SpringBootApplication annotation and the main method

→ Spring boot automatically scans all the components included in the project by using @ComponentScan annotation.

(7) How to configure Spring Boot externally?

→ Application properties: By default, spring boot searches for the application properties file or its YAML to load the properties

→ Command line properties: Spring boot provides command line arguments and converts these arguments to properties

→ profile-specific properties: These properties are loaded from the application-{profile}.properties file or its YAML file.

(8) What are Spring Boot starters?

→ Spring boot provides a number of starters that allow us to add jars in the classpath by adding this in pom.xml file

→ Spring boot built-in starters make development easier & rapid

```
<dependency>
    <groupId> org.springframework.boot </groupId>
    <artifactId> spring-boot-starter-web </artifactId>
</dependency>
```

(9) Some available spring boot starters:

→ spring-boot-starter-test

→ spring-boot-starter-tomcat

→ spring-boot-starter-jdbc

→ spring-boot-starter-data-jpa

→ spring-boot-starter-actuator

→ spring-boot-starter-logging

(10) What is auto configuration in spring boot & how to disable the auto-configuration ?

→ Spring boot auto configuration represents a way to automatically configure a spring application based on the dependencies that are present on the classpath

→ This can make development faster and easier

→ AutoConfiguration can be disabled by using annotation or through properties file

@EnableAutoConfiguration(exclude = { ---.class })

spring.autoconfigure.exclude=

(11) Is it possible to change the port of embedded Tomcat server in spring boot

→ yes.

→ server.port =

→ Default 8080

(12) What is Spring boot Actuator ?

→ Spring Boot actuator helps to monitor and manage the application

→ It helps in control our application by using HTTP endpoints.

(13) What is the need for Spring Boot Dev tools ?

→ The aim of this module is to fay and improve the development time while working with the Spring Boot application.

→ Spring boot DevTools pick up the changes & restart the application.

SOLID

① Single Responsibility principle.

"class should have only one reason to change"

Violation

```
public interface EmployeeService
{
    Employee getEmployeeById(long id);
    void addEmployee(Employee e);
    void sendEmail(Employee e, String content);
}
```

Solution.

```
public interface EmployeeService
{
    Employee getEmployee()
    void addEmployee(Employee e);
}

public interface EmailSender
{
    void sendEmail(Employee e, Content content);
}

public interface Content
{
}
```

① Open/Closed Principle.

"open for extension, but closed for modification"

Violation

```
public class SimpleCalculator
{
    public double addition(Addition addition)
    {
    }
    public double substraction(Substraction substraction)
    {
    }
}
```

Solution

```
public interface Calculator
{
    void calculate(Operation operation);
}

public class Addition implements Operation
{
    =
}

public class Addition implements Operation
{
    =
}
:

public class SimpleCalculator implements Calculator
{
    void calculate(Operation operation)
    {
        operation.performOperation();
    }
}
```

```
Public interface Operation
{
    void performOperation();
}
```

③ Liskov Substitution principle

" Subtypes must be substitutable for their base types "

Violation

```
public class MediaPlayer
{
    public void playAudio()
    {
    }
    public void playVideo()
    {
    }
}
```

```
public class WinampMediaPlayer extends MediaPlayer
{
    public void playVideo()
    {
        throw new Exception("");
    }
}
```

Solution

```
public class MediaPlayer
{
    public void playAudio()
    {
    }
}

public class VideoPlayer extends MediaPlayer
{
    public void playVideo()
    {
    }
}

public class WinMediaPlayer extends MediaPlayer
{
}
```

(4) Interface Segregation Principle

" clients should not be forced to implement interfaces they don't use "

12 Conventions for writing clean code

→ Magic Numbers

→ Deep Nesting

→ Comments

→ Avoid large functions

→ Code Repetition

→ Variable Naming [camel case]

→ Meaningful Names

→ Favor Descriptive over Concise

→ Use consistent Verbs per Concept

→ Use Nouns for class Name & Pascal case

→ Capitalize Constant Value

→ Avoid one-letter variable Names

Benefits

* Clean code
* Quality of code
* Readability of code
* Makes code maintenance easier

① What is the difference b/w POST, PUT and PATCH?

→ POST is always for creating a resource (does not matter if it was duplicated)

→ PUT is for checking if resource exists then update, else create new ~~request~~ resource

→ PATCH is always for updating the resource

→ PUT, if identifier doesn't exist, create new resource and assign a new identifier

→ PATCH, if identifier doesn't exist, we will throw an exception

→ PUT, it is mandatory to send all ~~a~~values again, the full payload

→ PATCH, we only send the data which we want to update

① What is monolithic Architecture?

→ It is a traditional way of developing applications where application is developed and deployed as a single unit

→ Application code size will be large as all the modules are clubbed together.

② What are adv & DB adv of monolithic arch?

→ Adv

→ simple to code & test

→ Easier to deploy

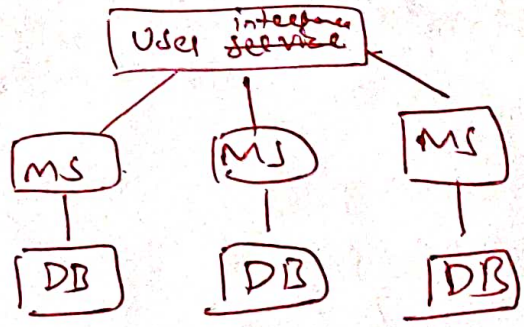
→ DB adv

→ tightly coupled

→ Need to redeploy entire app for even small change

→ Large in size, so difficult to manage

③ What are Microservices?

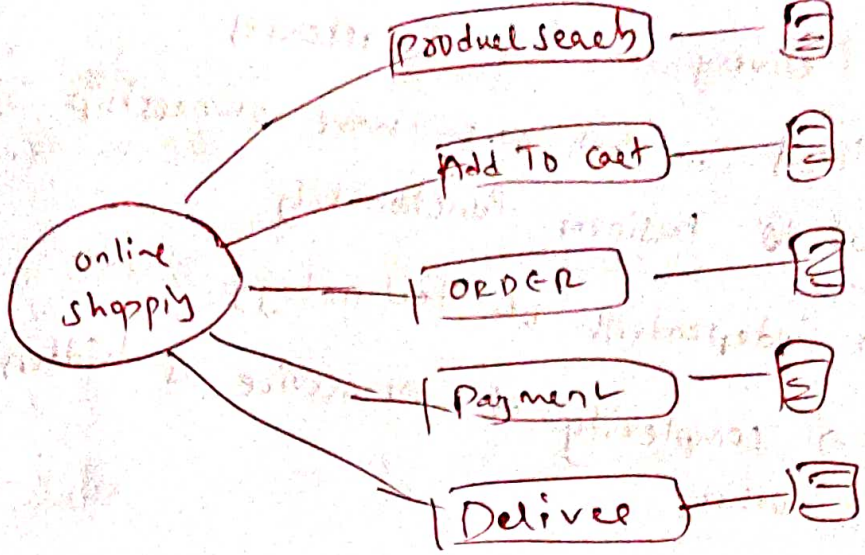


→ It is an architecture where a large application is divided into loosely coupled smaller services

→ Each service will serve one business goal

→ These services can be independly developed deployed and maintained.

Ex:



(4) What are the adv & disadv of Microservices

Adv

→ Loosely coupled
→ can use diff technologies
→ Easy to upgrade / bug fix
→ parallel Development & Delivery
→ faster Release cycles

Disadv

→ Difficult to Manage many services
→ Testing is Difficult
→ communication blw Microservices is complex
→ Needs large team size to work in parallel for faster development

(5) What are features of microservices

→ Decoupling [easily built, Modified, and scaled]
→ Business capability [focus on single Capability]

Continuous Delivery - frequent releases

De-centralized - Each M.s have ownership of data related to business functionality

polyglot → independent of technology

Black box → complexity of one service is hidden to other

⑥ When to use microservice?

→ When legacy app need to be modernized

→ Reduce time to market

→ Faster Development

→ if diff technologies can be used

→ Improved scalability

→ Having big team with required skillset & knowledge abt microservices

⑦ Diff b/w Monolithic, SOA and microservices architecture

Monolithic	SOA	Microservices
Application is built as single unit	Breaks up Monolithic app into smaller services	Breaks up SOA services further into independent services
Tightly coupled	loosely coupled	loosely coupled than SOA
Traditional way	Designed for service reusability & Data sharing	Designed for hosting services independently

single Data Storage for entire app	shared Data storage b/w services	can have shared single Data storage.
online shoping App	GoogleMapservices	CreditService DebitService

(8) Explain