🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi

To exit full screen, press esc

Q1) What is Spring Boot

1. Spring Boot is a Java framework that makes it easier to create and run Java applications.
2. It simplifies the configuration and setup process, allowing developers to focus more on writing code for their applications.
3. Spring Boot, a module of the Spring framework, facilitates Rapid Application Development (RAD) capabilities.
4. Spring Boot solve many developers problem
    a. Configurations
    b. Dependency Management
    c. Embedded Server

🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi

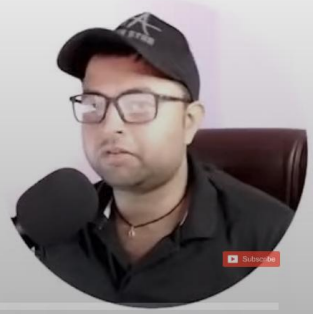
Q2) Why Spring Boot over Spring?

1. Spring Boot provide many advantages over normal spring framework
   a. **Easy to use**- Remove boilerplate codes
   b. **Production Ready applications**- Metrix, Health Check and many features are designed for production ready application.
   c. **Rapid Development-** Opinionated approach and auto-configuration enable developers to quickly develop apps
   d. **Provide Dependency Management**- No need to manager separate dependencies manually.
   e. **Autoconfiguration** - Spring Boot AutoConfigure things by default.
   f. **Embedded Server** : provide tomat server by default.

🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi

Q3) Working of Spring Boot

1. Spring Boot starts by scanning the starter dependencies in pom.xml
   Then **download** and **auto-configure** the module as you included in pom.xml
2. For example we have to create web application then we have to put **spring-boot-starter-web** dependency in pom.xml.
   When we start the project spring boot downloads all the dependency required for web and configure the things like spring mvc.



Q4) How Spring Boot Starts?

1. Starts by calling **main()** method of you main class.
2. The **run()** method of SpringApplicaiton is called. This method starts the application by creating an application **context** and **initializing** it.
3. Once the application context is initialized, the run() method starts the application's embedded web server.

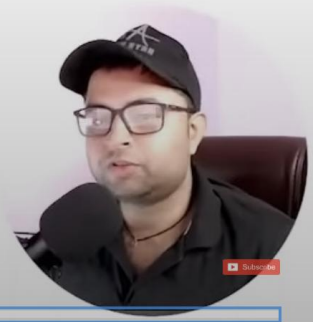
```
@SpringBootApplication
public class SpringBootWorkApplication {

    public static void main(String[] args) {
        SpringApplication.run(SpringBootWorkApplication.class, args);
    }
}
```

🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi

Q5) Top Spring Boot Annotations

- **@SpringBootApplication:** It combines three annotations: **@Configuration**,**@EnableAutoConfiguration**, and **@ComponentScan**. It is typically placed on the main class of the application.
- **@Component:** It is used to mark a class as a Spring bean that will be managed by the Spring container.
- **@Autowired:** This annotation is used to automatically inject dependencies into a Spring-managed bean.
- **@Service**: This annotation is used to indicate that a class represents a service component in the application. It is typically used to annotate classes that contain business logic.
- **@RestController**:Mark class as REST Controller. It is a specialized version of the **@Controlle**r annotation that includes the **@ResponseBody** annotation by default.

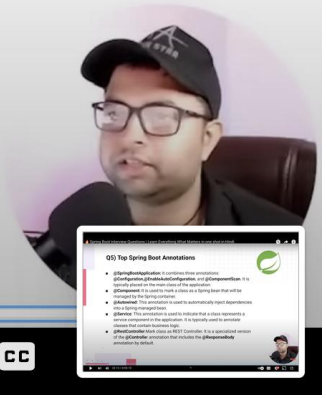


6

33:15 / 4:59:19

🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi

- **@RequestMapping:** used to map specific **url** to **method.**Used on class as well as method level.
- **@Repository:** mark class as DAO. mostly used on class that has database persistent logic.



34:25 / 4:59:19

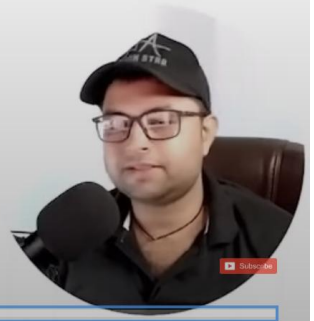
🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi

Q6) What are the Spring Boot Starters

- Starters are a collection of **pre-configured dependencies** that make it easier to develop particular kinds of applications.
- These starters include all of the **dependencies**, **version control**, and **configuration** needed to make certain features of a Spring Boot application functional.

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


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



🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi

Q7) What are the key dependencies of Spring Boot?

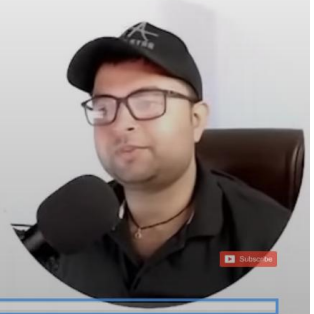
1. spring-boot-starter-parent
2. spring-boot-maven-plugin
3. spring-boot-starter-test
4. spring-boot-starter-security
5. spring-boot-starter-actuator
6. Spring-boot-starter-web
7. ……

38:37 / 4:59:19

🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi

Q8) what is Spring-Boot-Starter-Parent?

1. Spring Boot Starter Parent is a starter project that provides the default configuration for spring-based applications.
   a. The dependency management feature manages the versions of common dependencies.
   b. Provide the default compiler level as Java 1.8 and UTF-8 source encoding.
   c. Provides a default configuration for Maven plugins such as maven-surefire-plugin, maven-jar-plugin, and maven-failsafe-plugin.
   d. Executes a repackage goal with a repackage execution id.
   e. Resource filtering and configuring profile-specific files.

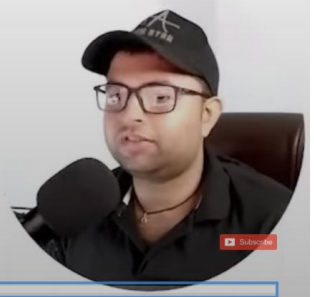


🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi

Q9) Can we use Only Spring Boot Dependency feature and configure maven plugin manually?

1. **YES**
2. We don't inherit from the **spring-boot-starter-parent** pom.
3. Include the **spring-boot-dependencies** dependency inside the dependencyMangement section as an import scope.

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <!-- Import dependency management from Spring Boot -->
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-dependencies</artifactId>
      <version>2.5.2</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
```

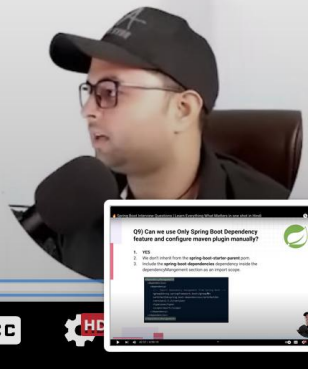


🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi

Q10) What is Spring Boot CLI and what are its benefits?

Command line tool to create,run and manage spring boot Applications

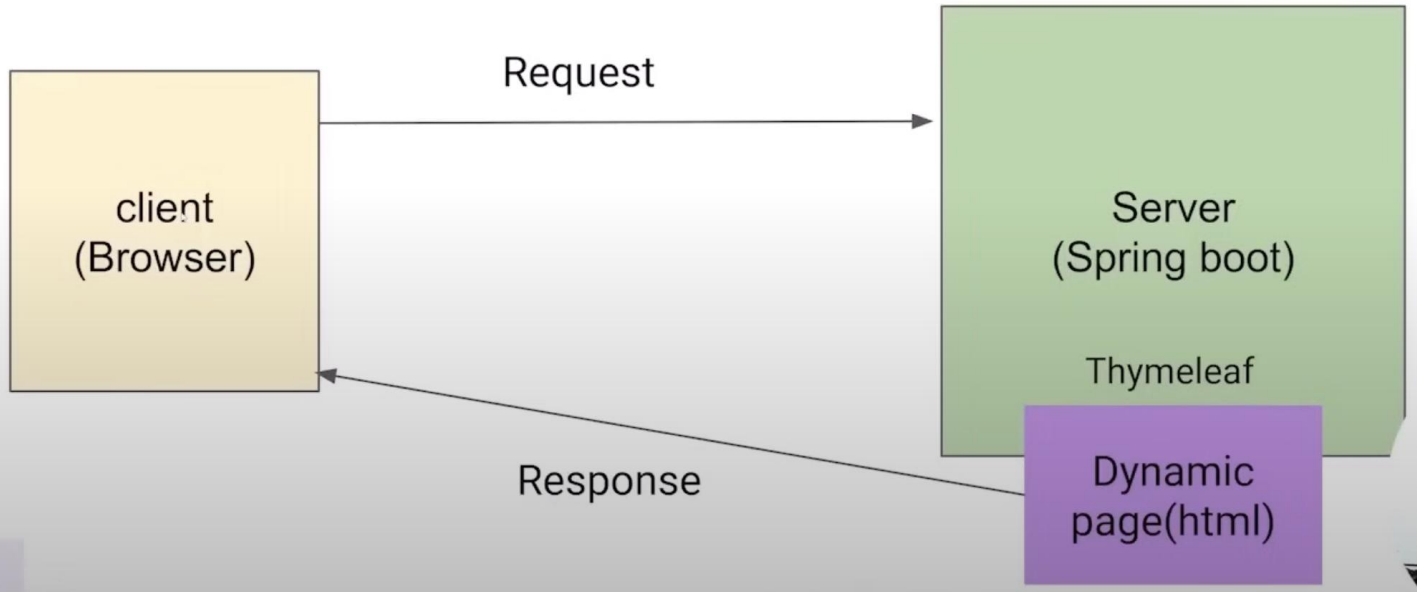
>>spring
>>spring help init
>>spring version
>>spring init --dependencies=web,data-jpa my-project
>> spring init --list



🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi

Q11) What is thymeleaf?

Java-based server-side templating engine used in Java web applications to render dynamic web pages.



🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi

Q12) What is IOC or inversion of control?

Inverting the control of creating object using new keyword to container or framework.

🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi

Q13) Explain the Spring Bean-LifeCycle.

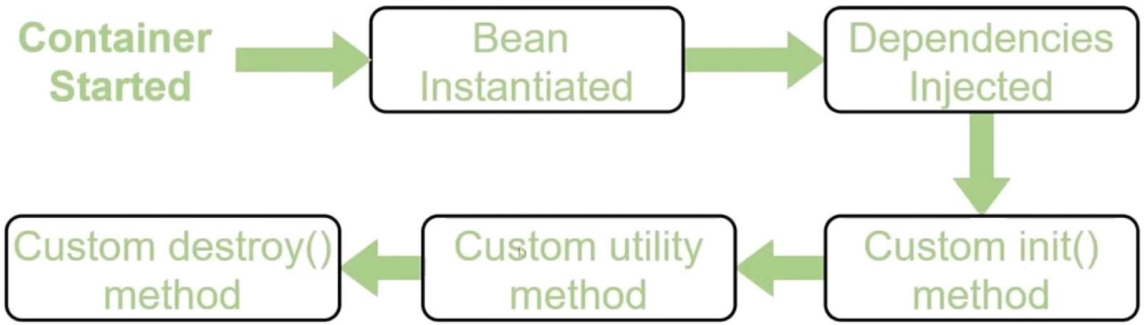
Life cycle means

1. How object is born
2. How it behaves (whole lifecycle)
3. How it dies.

Spring bean life cycle is maintained by IOC Container

1. Container gets started.
2. Container creates the object of bean as per request.
3. Dependencies is created
4. Dependencies is injected
5. Destroyed when container closed.

🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi



1:03:03 / 4:59:19

Q14) What is Bean Factory, have you used **XMLBeanFactory**?

1. This is the root interface for accessing a Spring bean container.
2. It is the actual container that instantiates, configures, and manages a number of beans.
3. **BeanFactory**(I) – Available in **org.springframework.beans.factory** package.

🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi

Q15) What are the difference between **BeanFactory** and **ApplicationContext** in Spring?

Bean Factory

- Bean instantiation/wiring

Application Context

- Bean instantiation/wiring
- Automatic BeanPostProcessor registration
- Automatic BeanFactoryPostProcessor registration
- Convenient MessageSource access (for i18n)
- ApplicationEvent publication

🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi

Q16) Difference between the **setter** and **constructor** injection in Spring?

- In constructor injection is important to remember the type and order of constructor parameters.
- Constructor Injection is for Mandatory Dependencies and Setter is for Optional

1:23:44 / 4:59:19

🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi

Q17) What are the different modules in spring?

Spring has seven core modules

1. The Core container module
2. Application context module
3. AOP module (Aspect Oriented Programming)
4. JDBC abstraction and DAO module
5. ORM module (Object/Relational)
6. Web Module
7. Test

🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi

Q18) Difference between **@Autowired** and **@Inject** annotation in Spring?

The @Inject annotation also serves the same purpose as @Autowried ,
The main difference between them is that @Inject is a **standard annotation** for dependency injection and @Autowired is spring specific.

1:43:02 / 4:59:19

🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi

Q19) Difference between @Bean and @Component annotation in Spring?

1. **@Component** Preferable for component scanning and automatic wiring.
2. **@Bean annotation** returns an object that spring should register as bean in application context. The body of the method bears the logic responsible for creating the instance.

1:45:47 / 4:59:19

🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi

Q20) What is autowiring in spring? What are the autowiring modes?

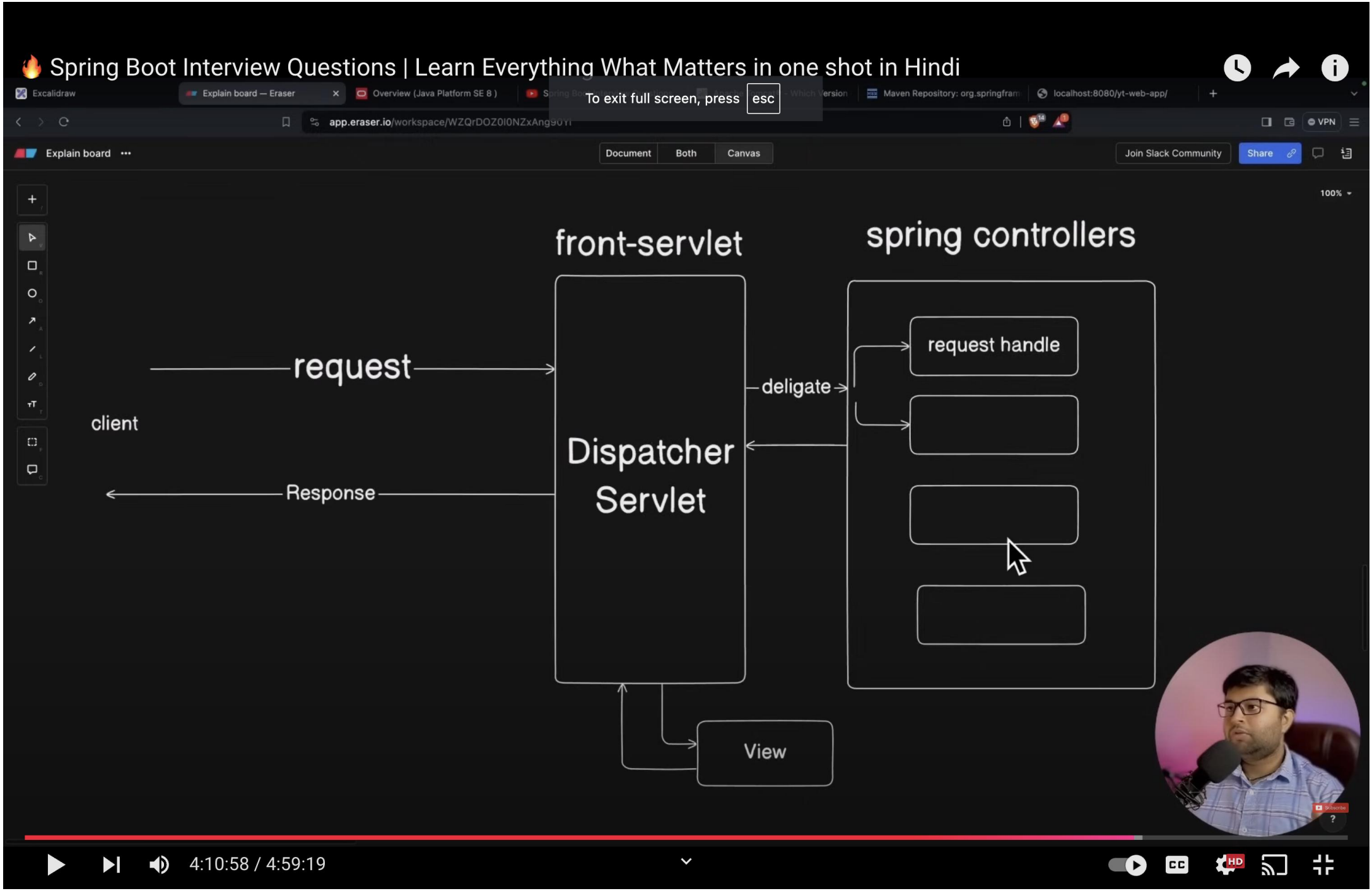
Injecting the beans automatically. We don't need to write explicit injection logic.

1) **no** - this is the default mode, it means autowiring is not enabled.
2) **byName** - injects the bean based on the property name. It uses setter method.
3) **byType** - injects the bean based on the property type. It uses setter method.
4) **constructor**- It injects the bean using constructor

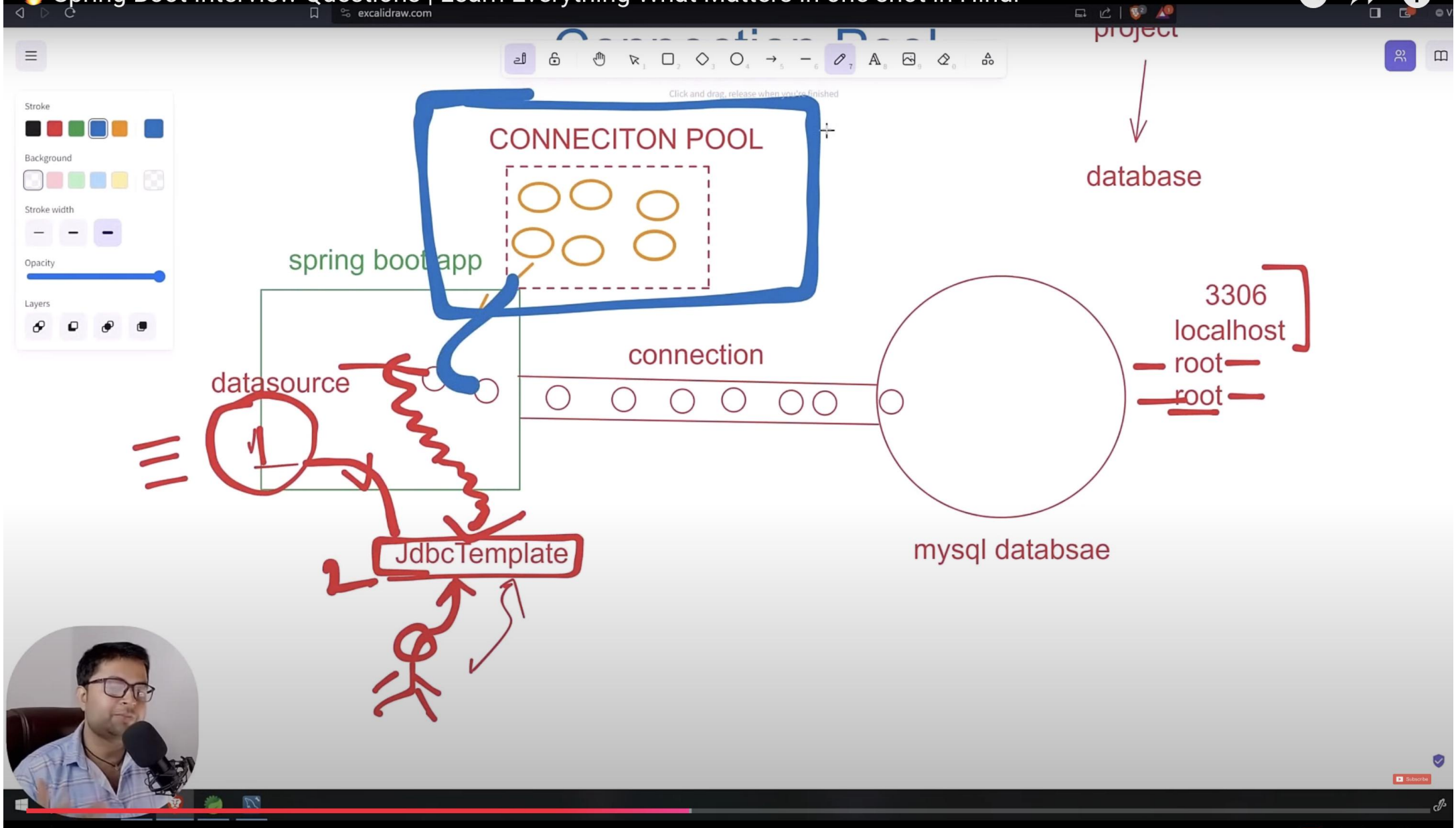
1:49:36 / 4:59:19

Q21) What are the different bean scopes in spring?

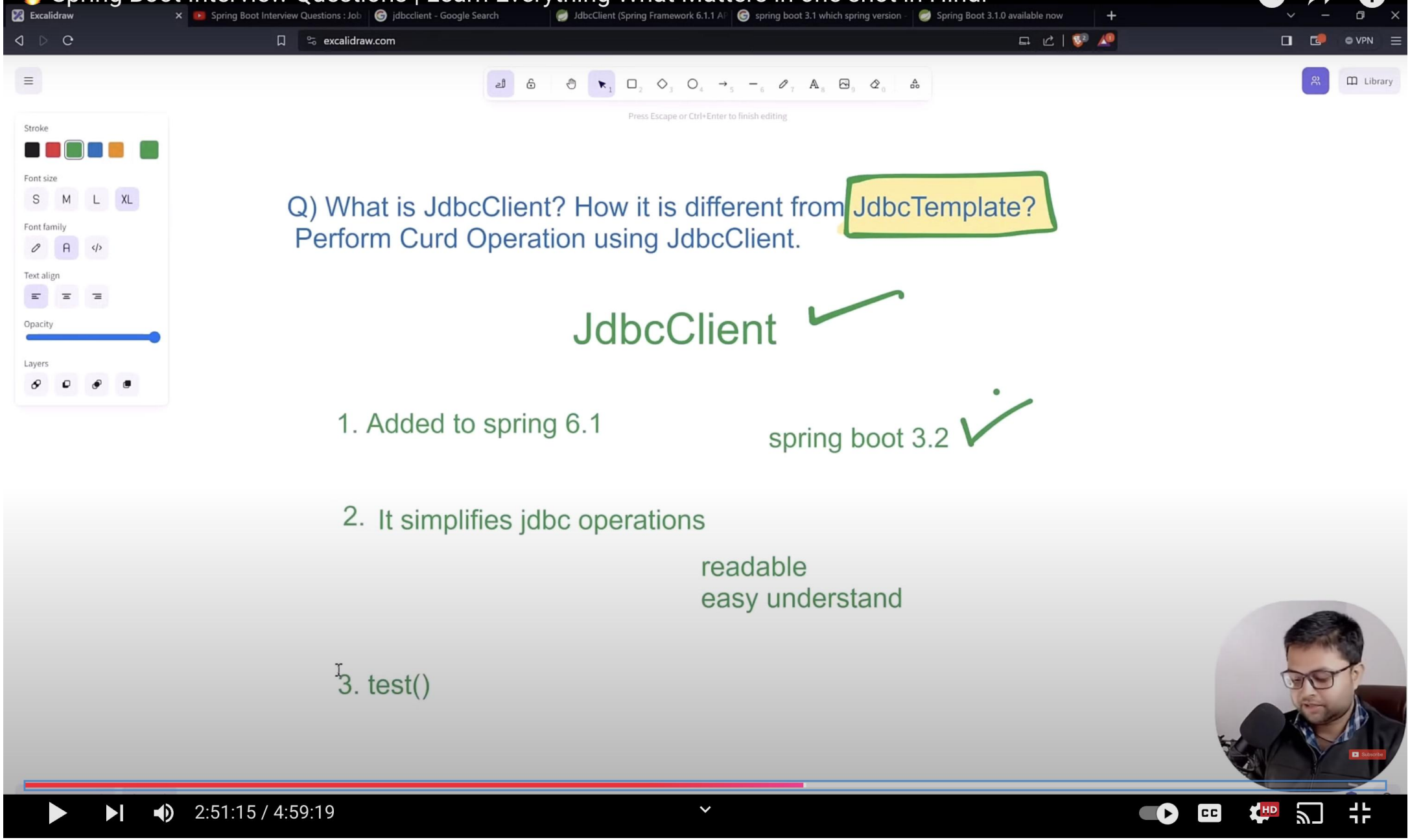
1. **Singleton**- The bean instance will be only once and same instance will be returned by the IOC container. It is the default scope.
2. **Prototype**- The bean instance will be created each time when requested.
3. **Request**- The bean instance will be created per HTTP request.
4. **Session**- The bean instance will be created per HTTP session.
5. **Globalsession**- The bean instance will be created per HTTP global session. It can be used in portlet context only.


🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi
To exit full screen, press esc
app.eraser.io/workspace/WZQrDOZ0I0NZxAng90Yl
Explain board
Document
Both
Canvas
Join Slack Community
Share
100%
front-servlet
spring controllers
request
client
Dispatcher Servlet
deligate
request handle
Response
View
4:10:58 / 4:59:19

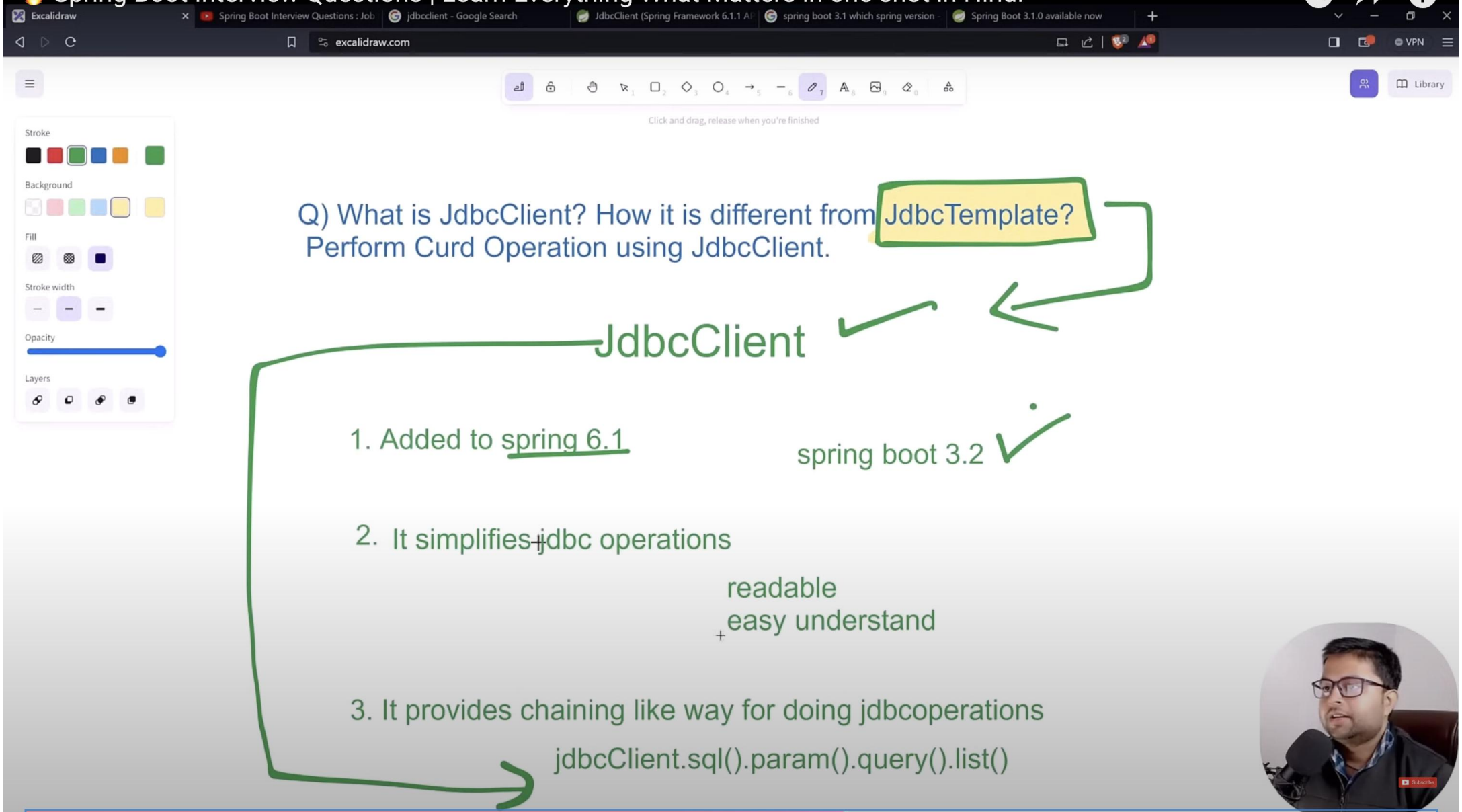
🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi



🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi



🔥 Spring Boot Interview Questions | Learn Everything What Matters in one shot in Hindi



2:53:25 / 4:59:19