

Summary

1. Spring IOC

- a. IOC: Inversion of Control is a mechanism that transfer the control of objects to a container in order to achieve loose-coupling between objects. It makes programs more scalable and easier to switch between implementations. We can achieve Inversion of Control through various mechanisms such as: Strategy design pattern, Service Locator pattern, Factory pattern, and Dependency Injection (DI).
- b. DI: Dependency injection means passing a dependent object as a parameter to a method, rather than having the method create the dependent object. It is used to achieve IOC. Dependency Injection in Spring can be done through constructors, setters or fields.
 - i. Constructor-Based Dependency Injection
The container will invoke a constructor with arguments that representing a dependency we want to set.
 - ii. Setter-Based Dependency Injection
The container will call setter methods of our class after invoking a no-argument constructor or no-argument static factory method to instantiate the bean.
 - iii. Field-Based Dependency Injection
we can inject the dependencies by marking them with an `@Autowired` annotation. The container will use reflection to inject dependency.
- c. Bean Scope: to make spring produce a new bean we need to define its scope, it has following attributes.
 - i. Singleton
A single instance per Spring IoC container.
 - ii. Prototype
Can have any number of object instances.
 - iii. Request
An HTTP request. Only valid in the context of a web-aware Spring ApplicationContext.
 - iv. Session
An HTTP session. Only valid in the context of a web-aware Spring ApplicationContext.
 - v. Global-Session
A global HTTP session. Only valid in the context of a web-aware Spring ApplicationContext.
- d. Bean Lifecycle

Bean life cycle is managed by the spring container. When we run the program then, first of all, the spring container gets started. After that, the container creates the instance of a bean as per the request, and then dependencies are injected. And finally, the bean is destroyed when the spring container is closed.

2. Spring AOP

- a. AOP: stands for Aspect Oriented Programming. The Spring AOP helps in breaking down the logic of the program into several distinct parts called as concerns. Cross-cutting concerns are the functions which span multiple points of an application. The cross-cutting concerns help in increasing the modularity and separate it from the business logic of an application.
 - i. Aspect: This is a module which has a set of APIs providing cross-cutting requirements.
 - ii. Advice: This is the actual action to be taken either before or after the method execution.
 - iii. Join Point: a point during the execution of a program, such as the execution of a method or handling an exception
 - iv. Point Cut: This is a set of one or more join points where an advice should be executed.
 - v. Target: The object being advised by one or more aspects.

- b. @Transactional

We can use @Transactional to wrap a method in a database transaction. It allows us to set propagation, isolation, timeout, read-only, and rollback conditions for our transaction.

3. Spring MVC

- a. MVC: 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.
 - i. Model - A model contains the data of the application. A data can be a single object or a collection of objects.
 - ii. Controller - A controller contains the business logic of an application.
 - iii. View - A view represents the provided information in a particular format.
 - iv. Front Controller - The DispatcherServlet class works as the front controller. It is responsible to manage the flow of the Spring MVC application.
- b. Spring MVC Workflow
 - i. Client requests for a page.
 - ii. Client request is intercepted by the Dispatcher Servlet also known as Front Controller. Dispatcher Servlet is a servlet specified in Web.XML file or in the Web Configuration class.
 - iii. Dispatcher Servlet uses URL Mapping Handler to find out the relevant controller class to which request should be passed.

- iv. Once Dispatcher Servlet has identified the Controller to be considered, it passes the client request to the controller.
- v. The controller class is the main class controlling the business logic flow once request has been dispatched it by dispatcher servlet. This class will implement the methods for different type of http requests (e.g. GET, POST) and all logic to call Service layer methods will reside in this controller class.

4. Spring Boot

Spring Boot is an open source Java-based framework used to create a micro Service.

a. Spring Boot Advantages

- i. Easy Deployment
- ii. Simple Scalability
- iii. Compatible with containers
- iv. Minimum Configuration
- v. Lesser Production Time

b. Spring Boot Starter

Spring Boot provides a number of starters that allow us to add jars in the classpath. Spring Boot built-in starters make development easier and rapid. Spring Boot Starters are the dependency descriptors.

c. Auto Configuration

Spring Boot auto-configuration attempts to automatically configure Spring application based on the jar dependencies that developers have added.

d. Rest API

Any REST request includes four essential parts: an HTTP method, an endpoint, headers, and a body.

An HTTP method describes what is to be done with a resource. There are four basic methods also named CRUD operations:

- POST to Create a resource,
- GET to Retrieve a resource,
- PUT to Update a resource, and
- DELETE to Delete a resource.

5. Error Handling in Spring Rest API

a. Controller Level @ExceptionHandler

The @ExceptionHandler annotated method is only active for that particular Controller, not globally for the entire application

b. Controller Advice

The@ControllerAdvice annotation allows us to consolidate our multiple, scattered @ExceptionHandler from before into a single, global error handling component.

6. Validation

We use spring boot starter validation to validate user input such as not null, max, pattern, Email.

7. Swagger

Swagger is widely used for visualizing APIs, and with Swagger UI it provides online sandbox for frontend developers.