

Spring Framework

* Spring →

It is a dependency injection Framework.

* Dependency Injection →

- It is a design pattern that removes the dependency from the programming code so that it can be easy to manage and test the application. It makes our programming code loosely coupled.
- It is a fundamental aspect of the Spring framework, through which the Spring container "injects" objects into other objects or "dependencies".

@Component

```
public class ComplexBusinessService {
    @Autowired
    SortAlgorithm sortAlgorithm;
```

@Component

```
public class BubbleSortAlgorithm implements SortAlgorithm {
```

* Terminologies →

- Bean
- Autowiring
- Dependency Injection
- Inversion of control
- IOC Container
- Application Context

Haq, ek behtar zindagi ka.

* Spring Level - 1 → Introduction to Spring Framework in 10 steps.

- ① Step-1 → Setting up a Spring Project using <http://start.spring.io>.
- ② Step-2 → Understanding Tight Coupling using the Binary search Algorithm Example.
- ③ Step-3 → Making the Binary search Algorithm example loosely couple.
- ④ Step-4 → Using Spring Framework to Manage dependencies -
@Component, @Autowired
- ⑤ Step-5 → What is happening in the background.
- ⑥ Step-6 → Dynamic autowiring and Troubleshooting - @Primary.
- ⑦ Step-7 → Constructor and setter Injection.
- ⑧ Step-8 → Spring Modules.
- ⑨ Step-9 → Spring Projects (Spring boot, Spring cloud, Spring batch)
- ⑩ Step-10 → Why is Spring popular?

Haq, ek behtar zindagi ka.

- Enables Testable code
- No plumbing code
- Flexible architecture
- Staying current

★ Spring Level - 2 → Spring Framework in depth

- ① Step-11 → Dependency Injection - A few more examples.
- ② Step-12 → Autowiring in depth - by name and @primary
- ③ Step-13 → Autowiring in depth - @Qualifier annotation.
- ④ Step-14 → Scope of a bean - Prototype and Singleton.
 - Bean Scope
 - Default - singleton
 - Singleton - One instance per Spring Context
 - Prototype - New bean whenever requested
 - Request - One bean per HTTP request
 - Session - One bean per HTTP session.

- ⑤ Step-15A → Complex scope scenarios of a Spring bean - Mix prototype and singleton.

- ⑥ Step-15B → Difference between Spring singleton and GOF singleton.

Haq, ek behtar zindagi ka.
(hang of four)

(16) Step-16 → Using Component Scan to scan for beans.

(17) Step-17 → Lifecycle of a Bean - @PostConstruct and @PreDestroy.

(18) Step-18 → Container and Dependency Injection (CDI) -
@Named, @Inject.
Context

- Java EE Dependency Injection Standard (JSR-330)
- Spring supports most annotations
 - @Inject (@Autowired)
 - @Named (@Component + @Qualifier)
 - @Singleton (Defines a scope of singleton)

(19) Step-19 → Removing Spring Boot in basic Application.

(20) Step-20 → Fixing minor stuff - Add logback and close Application Context.

(21) Step-21 → Defining Spring Application Context using XML-Part 1

(22) Step-22 → Defining Spring Application Context using XML-Part 2.

(23) Step-23 → Mixing XML Context with Component Scan for Beans defined with Annotation

Haq, ek behtar zindagi ka.

② Step-24 → IOC Container vs Application Context vs Bean Factory.

- Disambiguation

- IOC Container
- Application Context - Bean Factory ++
- Bean Factory

③ Step-25 → @Component vs @Service vs @Repository vs @Controller

- @Component - Generic Component
- @Repository - encapsulating storage, retrieval, and search behaviour typically from a relational db.
- @Service - Business Service Facade
- @Controller - Controller in MVC pattern

④ Step-26 → Read values from external properties files.

* Spring Level-3 → Unit testing with Spring Framework.

⑤ Step-27 → Spring Unit Testing with a Java context.

Sunday, December 17

⑥ Step-28 → Spring Unit Testing with a XML Context.

⑦ Step-29 → Spring Unit Testing with Mockito

Haq, ek behtar zindagi ka.

* Spring Level 4 → Spring Boot in 10 steps

① Step-1 → Introduction to Spring Boot - Goals & Imp.

• Goals ⇒

- 1) Enable building production ready applications quickly.
- 2) Provide common non-functional features.
 - Embedded servers
 - Metrics
 - Health checks
 - Externalized configuration.

• What Spring Boot is Not! ⇒

- 1) zero code generation
- 2) Neither an application server nor a web server.

• Features ⇒

- 1) Quick starter projects with auto configuration.

- Web
- JPA

- 2) Embedded servers - Tomcat, Jetty or Undertow

- 3) Production-ready features

- Metrics and health check
- Externalized configuration.

Haq, ek behtar zindagi ka.

② Step-2 → Developing Spring Application before Spring Boot.

③ Step-3 → Using Spring Initializr to create a Spring Boot Application.

④ Step-4 → Creating a Simple REST Controller.

⑤ Step-5 → What is Spring Boot Auto Configuration?

- It attempts to automatically configure your Spring application based on the jar dependencies that you have added. For ex., if HSQLDB is on your classpath, and you have not manually configured any database connection beans, then we will auto-configure an in-memory database.

⑥ Step-6 → Spring Boot vs Spring vs Spring MVC

• Spring vs Spring Boot →

- Spring Framework is the most popular app. dev. framework in java. Its main feature is dependency injection or IoC. With the help of it, we can develop a loosely coupled app. It is better to use if app. type or char. are purely defined.

• Spring boot is a module of Spring Framework. It allows us to build a stand-alone app. with minimal or 0 config's.

It is better to use if we want to develop a simple spring-based app. or RESTful services. Haq, ek behtar zindagi ka.

- Spring Boot vs Spring MVC →

- Spring Boot Makes it easy to quickly bootstrap and start developing a Spring-based app. It avoids a lot of boilerplate code. It hides a lot of complexity behind the scene, so that the developer can quickly get started and develop Spring-based applications easily.
- Spring MVC is a web MVC Framework for building web applications. It contains a lot of configuration files for various capabilities. It is an HTTP oriented web app. dev. framework.

⑦ Step-7 → Spring Boot Starter Projects - Starter web and Starter JPA

⑧ Step-8 → Overview of different Spring Boot Starter projects.

⑨ Step-9 → Spring Boot Actuator

- An actuator is a manufacturing term, referring to a mechanical device for moving or controlling something. Actuators can generate a large amount of motion from a small change.
- Actuator HTTP endpoints are only available with the Spring MVC-based app. In particular, it will not work with Jersey unless you enable Spring MVC as well.

⑩ Step-10 → Spring Boot developer tools

* Spring Level-5 → Spring AOP (Aspect Oriented Prog.).

① Step-1 → Setting up AOP example - Part 1

② Step-2 → Setting up AOP example - Part 2

③ Step-3 → Defining an @Before advice

④ Step-4 → Understand AOP Terminology - Pointcut, Advice, Aspect and Join Point.

⑤ Step-5 → Using @After, @AfterReturning, @AfterThrowing advices.

⑥ Step-6 → Using @Around advice to implement performance tracing.

⑦ Step-7 → Best Practice : Use common pointcut configuration.

⑧ Step-8 → Quick summary of other pointcuts.

⑨ Step-9 → Creating custom Annotation and an Aspect for Tracking Haq, ek behtar zindagi ki

* **Spring Level - 6 →** Interacting with Databases - Spring JDBC, JPA and Spring Data.

① Step-1 → Setting up a project with JDBC, JPA, H2 and Web dependencies.

② Step-2 → Launching up H2 Console.

③ Step-3 → Creating a Database Table In H2

④ Step-4 → Populate data into Person Table.

⑤ Step-5 → Implement findAll persons Spring JDBC query Method.

⑥ Step-6 → Execute the findAll method using CommandLineRunner

⑦ Step-7 → A quick review - JDBC vs Spring JDBC

Advantages -

- The Spring JDBC template allows to clean up the resources automatically, e.g; release the db connection,
- It converts the standard JDBC SQLException into RuntimeException.
- This allows the programmer to react more flexible to the errors.
- The code is also less.

Haq, ek behtar zindagi ka.

⑧ Step-8 → What's in the background? Understanding Spring Boot Configuration.

⑨ Step-9 → Implementing findById Spring JDBC query Method.

⑩ Step-10 → Implementing deleteById Spring JDBC update Method.

⑪ Step-11 → Implementing insert and update Spring JDBC update Method.

⑫ Step-12 → Creating a custom Spring JDBC RowMapper.

⑬ Step-13 → Quick introduction to JPA. (Java Persistence API)

• It is a specification of Java. It is used to persist data b/w Java project and relational db. JPA acts as a bridge b/w object-oriented domain Models and relational db systems.

• As JPA is a specification, it doesn't perform any ~~any~~ operation by itself. It requires an implementation. Sunday, December 24
So, ORM tools like hibernate, TopLink and iBatis implement JPA specifications for data persistence.

⑭ Step-14 → Defining Person Entity.

Haq, ek behtar zindagi ka.

(13) Step-15 → Implementing findById JPA Repository Method.

(14) Step-16 → Implementing insert and update JPA Repository Methods.

(15) Step-17 → Implementing deleteById JPA Repository Method.

(16) Step-18 → Implementing findAll using JPQL Named query.
• (Java Persistence Query Language) is used to define searches against persistent entities independent of the mechanism used to store those entities.

(17) Step-19 → Introduction to Spring Data JPA.

- Spring data JPA API provides JpaRepository class to integrate spring application with JPA.
- JPA is the sun specification for persisting objects in the enterprise app. It is currently used as the replacement for complex entity beans.

(18) Step-20 → Connecting to other databases.

"9, ek behtar zindagi ka.

* Web application with Spring MVC →

① Step-1 → Setting up your first java web application.

- Theory-1 → Maven and Magic

- It is a powerful project Management tool that is based on POM (Project Object Model). It is used for projects build, dependency and documentation.

- Theory-2 → What is a servlet?

- It is used to create a web application (resides at server side and generates a dynamic web page).
- It is robust and scalable because of java language.

- Theory-3 → Web application request flow

- Theory-4 → Understand Your first servlet - LoginServlet

② Step-2 → Create a LoginServlet from scratch again and your first view.

③ Step-3 → Theory - Play time - Let's try breaking things.

④ Step-4 → Parsing request parameters using Get Method.

⑤ Step-5 → Theory - Introduction and end to scriptlets.

- It is used to execute java source Haq, ek behtar zindagi ka. code in JSP.

⑥ Step-6 → Disadvantages of GET parameters.

- It can't be used to send word documents or images.
- GET requests can be used only to retrieve data.
- The length of URL is limited.
- If we use GET method, the browser appends the data to the URL.

⑦ Step-7 → Your first POST request

⑧ Step-8 → Your first servlet doPost Method

⑨ Step-9 → Let's add a password field

⑩ Step-10 → Setting up Maven, Tomcat and simple JEE app.

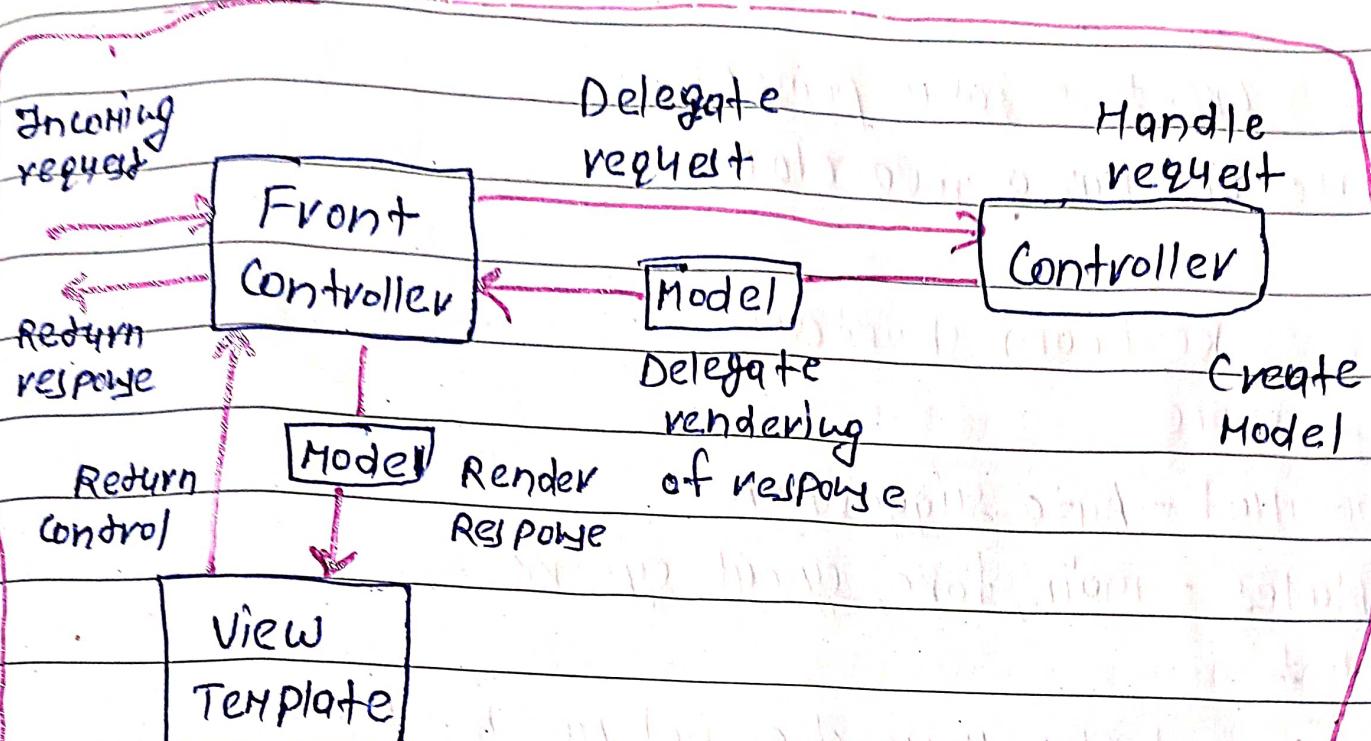
⑪ Step-11 → Setting up Spring MVC with 4 mini steps

- Rename package webapp to com.in28minutes.jee
- Add required jars to the project.
- Add dispatcher servlet to web.xml
- Add spring context.

⑫ Step-12 → Your first spring MVC controller.

⑬ Step-13 → Part -2 → Your first Spring MVC view
: ViewResolver,

- Part - 2 → Theory break - Spring MVC architecture.



JSP Engine

⑩ - Part - 3 → Play break - Try breaking things.

⑪ Step - 14 → Add logging framework Log4j

⑫ Step - 15 → Redirect to welcome page : Model Map and
@RequestParam

⑬ Step - 16 → Use Login Service to authenticate

⑭ Step - 17 → Spring Autowiring and Dependency Injection

* Basic tools & frameworks - Eclipse in 5 steps →

① Step-1 → Create a Java project.

- Create and run a Java class.

② Step-2 → Keyboard shortcuts.

- Ctrl + Space

• BigDecimal - Auto suggestion

• Templates - main, fore, System.out, System.err.

- Ctrl + 1

• File Name and Class Name does not Match - Display errors

• Rename a class - What suggestions are offered?

• new Integer() - What suggestions are offered?

- Ctrl + Shift + R (and T)

- F3 (Goto declaration)

- F4 (Type Hierarchy)

- Ctrl + Shift + L

③ Step-3 → Views and perspectives.

④ Step-4 → Save action.

⑤ Step-5 → Code generation,

- Alt + Shift + S

• Getters and setters

• toString()

• equals()

Haq, ek behtar zindagi ka.

• hashCode()

Basic tools & Frameworks - Maven in 5 steps

① Step-1 → Creating and importing a Maven project.

② Step-2 → Understanding project object model - pom.xml

- Naming a project
- Declaring dependencies

③ Step-3 → Maven build life cycle.

- run "mvn clean install"
- Build lifecycle - Validate, compile, test, package, integration test, verify, install, deploy.
- Convention over configuration - Pre defined folder structure.
 - source code
 - \${basedir}/src/main/java
 - \${basedir}/src/main/resources
 - Test code
 - \${basedir}/src/test

④ Step-4 → How does Maven work?

- local repository
- Maven repository

Sunday, December 31

dependency: sources

mvn compile (Compiles source file)

mvn clean - delete target directory.

mvn --version

mvn test - run unit tests

mvn package - creates the Jar

Haq, ek behtar zindagi ka

dependency: tree | --debug