FRAMEWORK:

A Framework or software framework is a platform for developing applications for example a framework may include pre-defined classes and functions that can be used to process input manage hardware devices and interact with system software.

SPRING FRAMEWORK:

The Spring Framework provides a comprehensive programming and configuration model for modern Java-based enterprise applications - on any kind of deployment platform.

As we know if we want create real-time applications by using spring framework it includes many XML configurations , server setting adding dependencies.

FEATURES OF SPRING FRAMEWORK:

* Lightweight. The Spring Framework is very lightweight with respect to its size and functionality.
* ii. Aspect-Oriented Programming (AOP)
* v. Dependency Injection. ...
* vi. Integration With Other Frameworks.

DEPENDENCY INJECTION:

Dependency Injection is a fundamental aspect of the Spring framework, through which the Spring container “injects” objects into other objects or “dependencies”. Simply put, this allows for loose coupling of components and moves the responsibility of managing components onto the container.

ASPECT-ORIENTED PROGRAMMING (AOP):

Aspect-Oriented Programming entails breaking down program logic into distinct parts called so-called concerns. Spring AOP module provides interceptors to intercept an application. For example, when a method is executed, you can add extra functionality before or after the method execution.

SPRING FRAMEWORK INTEGRATION:

Spring framework is integrate with other framework like HIBERNATE, STRUTS.

NOTE**:**

Spring boot is framework developed on top of core spring framework.

SPRING BOOT:

* Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run".
* The main aim of the spring boot is create real time applications in very less time.
* Spring boot comes with inbuilt server we no longer have to use any external servers like tomcat, glassfish or anything else. So don’t need to deploy WAR files.
* Supporting tools are maven and gradle.
* Spring Boot is an open-source micro framework maintained by a company called Pivotal. It provides Java developers with a platform to get started with an auto configurable production-grade Spring application.

FEATURES OF SPRING BOOT:

* Web Development
* Spring Application
* It allows standalone applications
* Admin features
* No need deploy WAR files
* Embedded Tomcat
* Logging
* Security

WEB DEVELOPMENT:

It is well suited Spring module for web application development. We can easily create a self-contained HTTP server using embedded Tomcat, Jetty or Undertow. We can use the spring-boot- starter-web module to start and running application quickly.

SPRING APPLICATION:

It is a class which provides the convenient way to bootstrap a spring application which can be started from main method. You can call start your application just by calling a static run() method.

ADMIN FEATURES:

Spring Boot provides the facility to enable admin related features for the application. It is used to access and manage application remotely. We can enable it by simply using *spring.application.admin.enabled* property.

LOGGING:

Spring Boot uses Common logging for all internal logging. Logging dependencies are managed by default. We should not change logging dependencies, if there is no required customization is needed.

SECURITY:

Spring Boot applications are spring bases web applications. So, it is secure by default with basic authentication on all HTTP endpoints. A rich set of Endpoints are available for develop a secure Spring Boot application.

SPRING BOOT ARCHITECTURE

Spring Boot is a module of the Spring Framework. It is used to create stand-alone, production-grade Spring Based Applications with minimum efforts. It is developed on top of the core Spring Framework.

Spring Boot follows a layered architecture in which each layer communicates with the layer directly below or above (hierarchical structure) it.

Before understanding the Spring Boot Architecture, we must know the different layers and classes present in it. There are four layers in Spring Boot are as follows:

* Presentation Layer
* Business Layer
* Persistence Layer
* Database Layer



PRESENTATION LAYER:

 The presentation layer handles the HTTP requests, translates the JSON parameter to object, and authenticates the request and transfer it to the business layer. In short, it consists of views i.e., frontend part.

BUSINESS LAYER:

The business layer handles all the business logic. It consists of service classes and uses services provided by data access layers. It also performs authorization and validation.

PERSISTENCE LAYER:

The persistence layer contains all the storage logic and translates business objects from and to database rows.

DATABASE LAYER:

 In the database layer, CRUD (create, retrieve, update, delete) operations are performed.

The **CRUD** stands for **Create, Read/Retrieve, Update,** and **Delete**. These are the four basic functions of the persistence storage.

The CRUD operation can be defined as user interface conventions that allow view, search, and modify information through computer-based forms and reports. CRUD is data-oriented and the standardized use of **HTTP action verbs**. HTTP has a few important verbs.

* **POST:** Creates a new resource
* **GET:** Reads a resource
* **PUT:** Updates an existing resource
* **DELETE:** Deletes a resource

Within a database, each of these operations maps directly to a series of commands. However, their relationship with a RESTful API is slightly more complex.

## STANDARD CRUD OPERATION:

* **CREATE Operation:** It performs the INSERT statement to create a new record.
* **READ Operation:** It reads table records based on the input parameter.
* **UPDATE Operation:** It executes an update statement on the table. It is based on the input parameter.
* **DELETE Operation:** It deletes a specified row in the table. It is also based on the input parameter.

## HOW CRUD OPERATIONS WORKS:

CRUD operations are at the foundation of the most dynamic websites. Therefore, we should differentiate **CRUD** from the **HTTP** **action verbs**.

Suppose, if we want to **create** a new record, we should use HTTP action verb **POST**. To **update** a record, we should use the **PUT** verb. Similarly, if we want to **delete** a record, we should use the **DELETE**verb. Through CRUD operations, users and administrators have the right to retrieve, create, edit, and delete records online.

The primary comparison between Spring and Spring Boot are discussed below:

|  |  |
| --- | --- |
| **Spring** | **Spring Boot** |
| **Spring Framework** is a widely used Java EE framework for building applications. | **Spring Boot Framework** is widely used to develop **REST APIs**. |
| It aims to simplify Java EE development that makes developers more productive. | It aims to shorten the code length and provide the easiest way to develop **Web Applications**. |
| The primary feature of the Spring Framework is **dependency injection**. | The primary feature of Spring Boot is **Autoconfiguration**. It automatically configures the classes based on the requirement. |
| It helps to make things simpler by allowing us to develop **loosely coupled** applications. | It helps to create a **stand-alone** application with less configuration. |
| The developer writes a lot of code (**boilerplate code**) to do the minimal task. | It **reduces** boilerplate code. |
| To test the Spring project, we need to set up the sever explicitly. | Spring Boot offers **embedded server** such as **Jetty** and **Tomcat**, etc. |
| It does not provide support for an in-memory database. | It offers several plugins for working with an embedded and **in-memory** database such as **H2**. |
| Developers manually define dependencies for the Spring project in **pom.xml**. | Spring Boot comes with the concept of **starter** in pom.xml file that internally takes care of downloading the dependencies **JARs** based on Spring Boot Requirement. |

SREFERENCE LINKS:

1. **https://www.javatpoint.com/spring-boot-features**

hg