- ❖ **Spring Boot** : 1)It is built on the top of Spring framework.it is easier and faster way to setup ,Configure and run simple and web based applications.
  2)It provide RAD(Rapid Application Development) to Spring framework which is used to develop Enterprise Applications.
  3)its help to faster application.
  4)It help to develop REST API(Representationl State Transfer)

  - **Spring Boot** :- Spring framework + Embeded Servers (server that is scalable and can hot application that is used by million of user[Tomcat])
  - **Plugin** :-It is add on or software program used to run existing computer program (e.g Tomacat :used to host java application ,JDBC driver to connect java with SQL)

  e.g Spring initializer can be used to develop Spring Boot application)

  - **Need of Spring boot**:

  1)Dependancy Injuction
  2)Powerful database Transaction Management Capabilities.
  3)Security
  4)allow integration of other framework like Hibernate ,Struct
  5)Reduce Development time.

  - **Spring Sister Projects** :

    - **Spring Data**: It simplifies data access from the relational and NoSQL databases.
    - **Spring Batch**: It provides powerful batch processing.
    - **Spring Security**: It is a security framework that provides robust security to applications.
    - **Spring Social**: It supports integration with social networking like LinkedIn.
    - **Spring Integration**: It is an implementation of Enterprise Integration Patterns. It facilitates integration with

other enterprise applications using lightweight messaging and declarative adapters.

- **Advantages of Spring Boot** :

1) It tests web applications easily with the help of different **Embedded** HTTP servers such as **Tomcat, Jetty,**
2) It provides production-ready features such as metrics, health checks, and externalized configuration.
3) There is no requirement for XML configuration.
4) It offers a CLI tool for developing and testing the Spring Boot application.
5) It offers the number of plug-ins.
6) It also minimizes writing multiple boilerplate codes (the code that has to be included in many places with little or no alteration), XML configuration, and annotations.
7) It increases productivity and reduces development time.

- **Features :**

1)Web Development

It is a well-suited Spring module for web application development. We can easily create a self-contained HTTP application that uses embedded servers like **Tomcat, Jetty,** or Undertow. We can use the **spring-boot-starter-web** module to start and run the application quickly.

2)SpringApplication

The SpringApplication is a class that provides a convenient way to bootstrap a Spring application. It can be started from the main method. We can call the application just by calling a static run() method.

3)Application Events and Listeners

Spring Boot uses events to handle the variety of tasks. It allows us to create factories file that is used to add listeners. We can refer it to using the **ApplicationListener key**.

Always create factories file in META-INF folder like **META-INF/spring.factories**.

4)Admin Support

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

5)Externalized Configuration

Spring Boot allows us to externalize our configuration so that we can work with the same application in different environments. The application uses YAML files to externalize configuration.

6)Properties Files

Spring Boot provides a rich set of **Application Properties**. So, we can use that in the properties file of our project. The properties file is used to set properties like **server-port =8082** and many others. It helps to organize application properties.

7)YAML Support

It provides a convenient way of specifying the hierarchical configuration. It is a superset of JSON. The SpringApplication class automatically supports YAML. It is an alternative of properties file.

8)Type-safe Configuration

The strong type-safe configuration is provided to govern and validate the configuration of the application. Application configuration is always a crucial task which should be type-safe. We can also use annotation provided by this library.

9)Logging

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

10)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 is available to develop a secure Spring Boot application.

- **Goals of Spring Boot** :

The main goal of Spring Boot is to reduce development, unit test, and integration test time.

1)Provides Opinionated Development approach

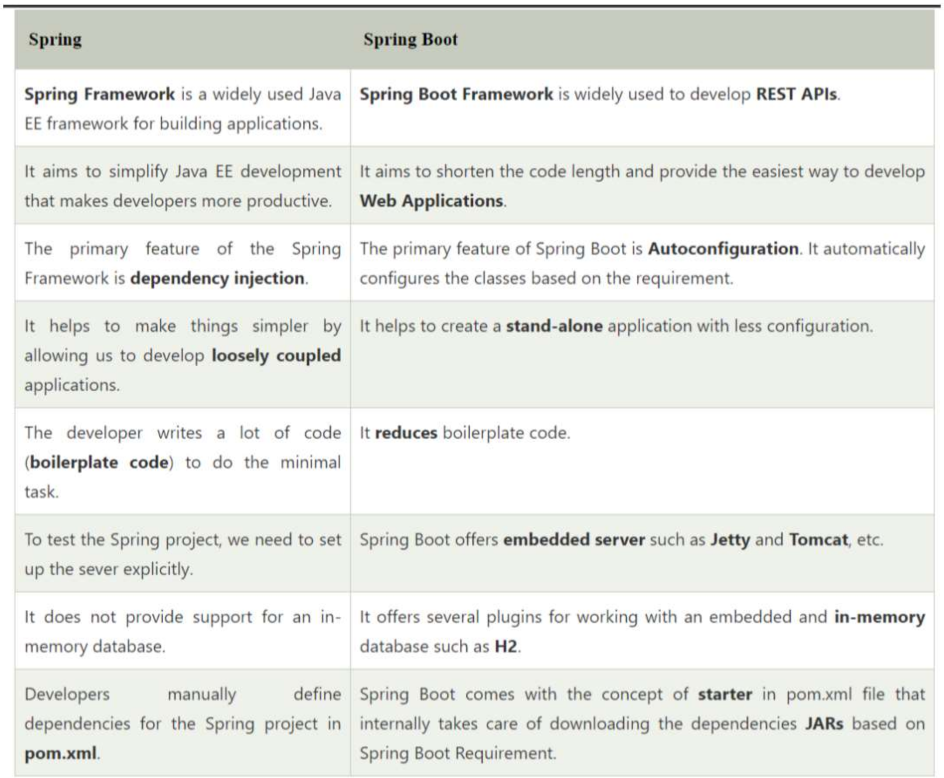
2)Avoids defining more Annotation Configuration

3)Avoids writing lots of import statements

4)Avoids XML Configuration.

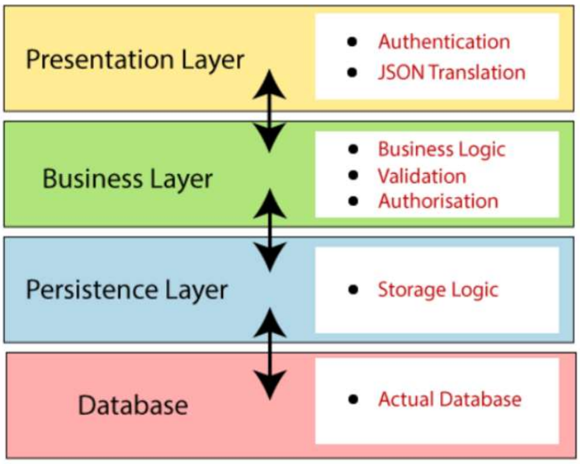
- **Limitation** :

Dependancy increase Size of Application.



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.

❖ Spring Boot Architecture :

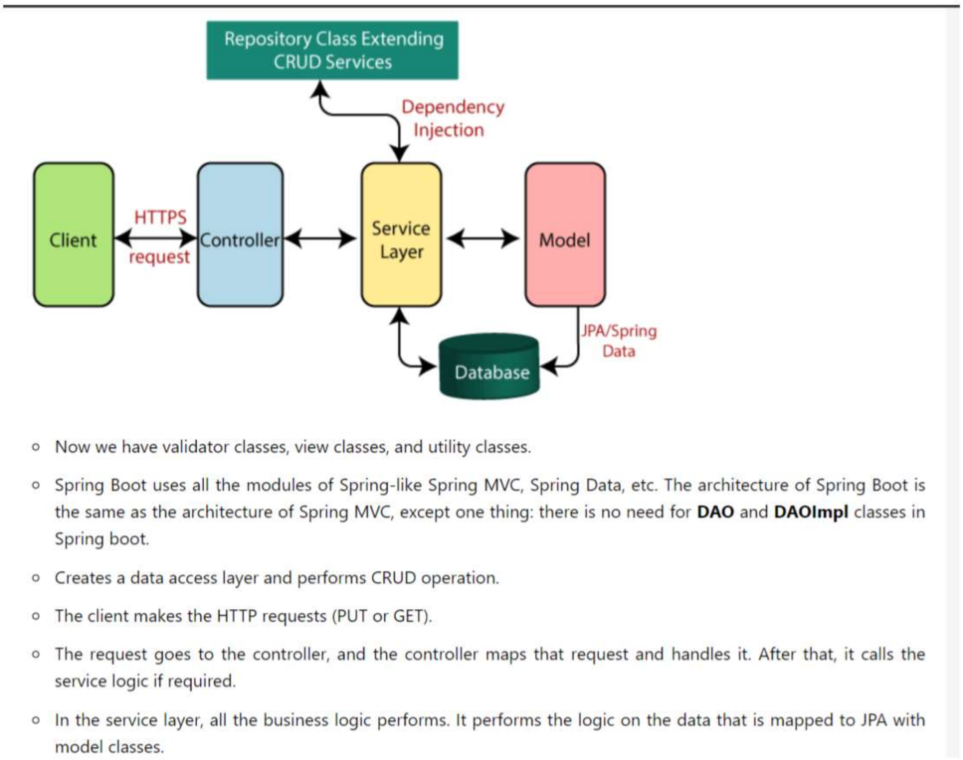


Presentation Layer: The presentation layer handles the HTTP requests, translates the JSON parameter to object, and authenticates the request and transfer it to the business layer. In short, it consists of **views** i.e., frontend part.

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.



- Now we have validator classes, view classes, and utility classes.
- Spring Boot uses all the modules of Spring-like Spring MVC, Spring Data, etc. The architecture of Spring Boot is the same as the architecture of Spring MVC, except one thing: there is no need for **DAO** and **DAOImpl** classes in Spring boot.
- Creates a data access layer and performs CRUD operation.
- The client makes the HTTP requests (PUT or GET).
- The request goes to the controller, and the controller maps that request and handles it. After that, it calls the service logic if required.
- In the service layer, all the business logic performs. It performs the logic on the data that is mapped to JPA with model classes.

- Dependancy Injection : It is just process where ,if one class require object of other class so we don't need to give dependency Spring Boot give it.
- Spring AOP (Aspect Oriented Programming) : it help to tackle cross cutting things ,as in application some things are repetitively performed so we write method that invoked .(login ,logout).
- Spring MVC : Allow us to create web application.
- Spring data libraries : when we want to caonnect java to mysql we need JDBC and working with it make it complicated so spring provide data libraries for the same.for different database like mysql ,oracle spring data project available that can be used.
- Spring boot also called as framework of framework because it allow integration of other framework like Hibernate.
- **Tightly coupling** : suppose we have doctor class and have some method like assist() and again there is class classification that is present in doctor class [so that we can test it together]
- **Loose Coupling** : it comes when dependency injenction arise because contain different classes that are created an separately and obviously they are referring one anouther so spring will have reference graph we don't need to explicitly do this.[we just define blueprint and spring will take care of inject object and all]we are giving control to the spring that is inversion of control.

❖ Dependancy injection : can be done

- XML configuration
- Java Annotation
- Java Configuration