Spring Boot - Architecture

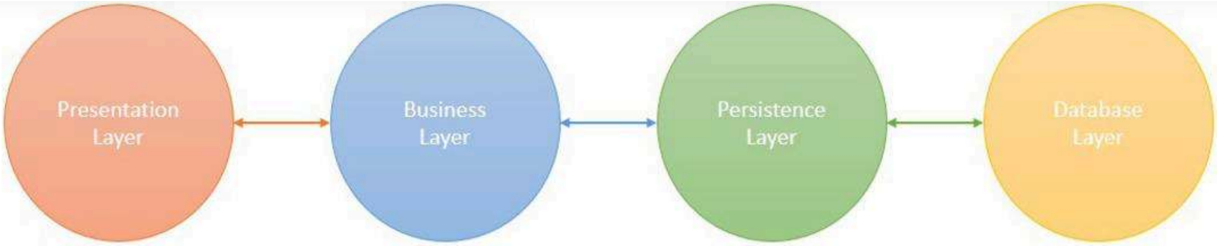
Last Updated : 21 Aug, 2025

Spring Boot is built on top of the Spring Framework and follows a layered architecture. Its primary goal is to simplify application development by providing auto-configuration, embedded servers and a production-ready environment out of the box.

The architecture of Spring Boot can be divided into several layers and components, each playing an important role in building and running modern applications.

Spring Boot Architecture Layers

Spring Boot consists of the following four layers:



1. Presentation Layer

- Handles HTTP requests through REST controllers (GET, POST, PUT, DELETE).
- Manages authentication, request validation and JSON serialization/deserialization.
- Forwards processed requests to the Business Layer for further logic.

2. Business Layer

The Business Layer is responsible for implementing the application's core logic. It consists of service classes that:

- Process and validate data.
- Handle authentication and authorization (integrating Spring Security if needed).
- Apply transaction management using @Transactional.
- Interact with the Persistence Layer to store or retrieve data.

3. Persistence Layer

The Persistence Layer manages database transactions and storage logic. It consists of repository classes using Spring Data JPA, Hibernate or R2DBC for data access. It is responsible for:

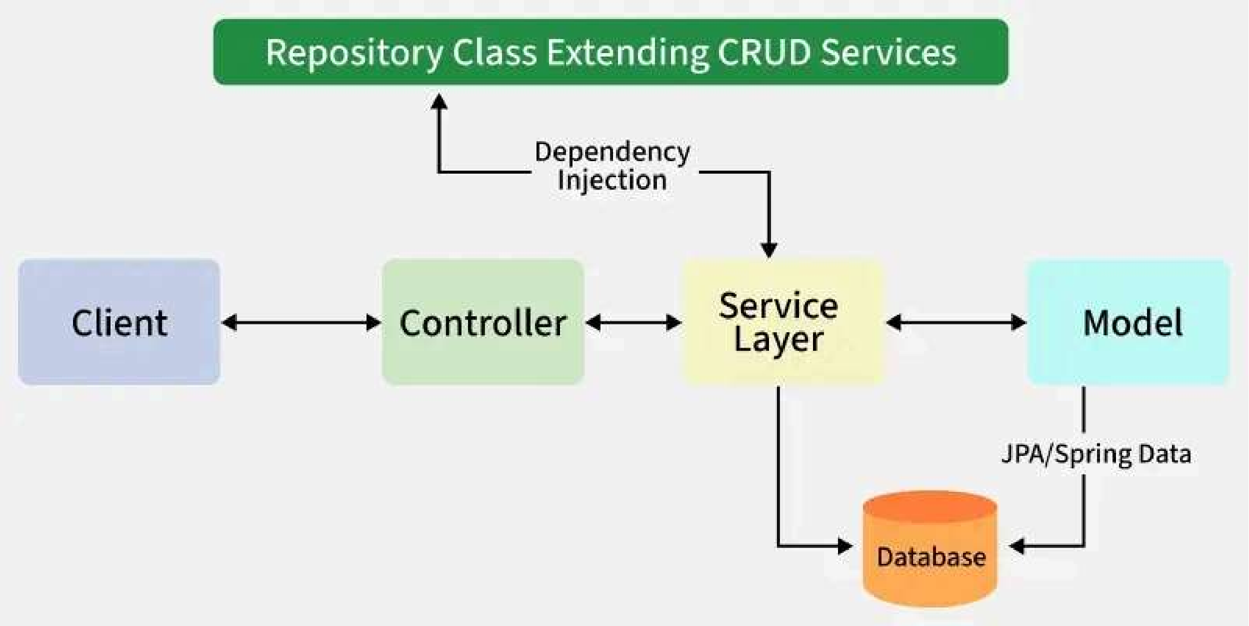
- Mapping Java objects to database records using ORM frameworks.
- Managing CRUD (Create, Read, Update, Delete) operations.
- Supporting relational and NoSQL databases.

4. Database Layer

The Database Layer contains the actual database where the application data is stored. It can support:

- Relational Databases (MySQL, PostgreSQL, Oracle, SQL Server).
- NoSQL Databases (MongoDB, Cassandra, DynamoDB, Firebase).
- Cloud-based databases for scalability.

Spring Boot Flow Architecture



Architecture flow

Explanation:

- The client (frontend or API consumer) sends an HTTP request (GET, POST, PUT, DELETE) to the application.
- The request is handled by the Controller Layer, which maps the request to a specific handler method.
- The Service Layer processes business logic and communicates with the Persistence Layer to fetch or modify data.
- The Persistence Layer interacts with the Database Layer using Spring Data JPA or R2DBC, often through a **Repository Class** that extends CRUD services.
- The processed response is returned as JSON.
- Spring Boot Actuator can be used for monitoring and health checks.

Comment More info Advertise with us

GeeksforGeeks
Sanchhaya Education Private Limited