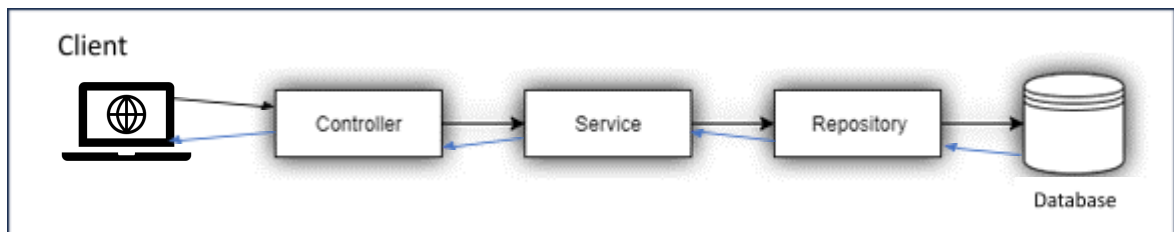


Core Concepts in Spring:

When we build a standalone application or a web app, the client's computer talks to the backend, which then talks to the database to get the needed data. This is done using multiple layers, such as the controller layer, service layer, and repository layer.



In Java, everything is an object, and we use OOP (Object-Oriented Programming). To ensure smooth coordination, each layer must contain objects from the other layers to work with. Managing these objects can be a difficult task for developers. Instead of focusing on business logic, developers often find themselves bogged down by object management. Spring offers a solution to this problem called Spring Boot.

Inversion of Control (IoC)

Inversion of Control (IoC) is a concept where Spring takes charge of creating and managing objects, which is typically done by developers in traditional programming.



In traditional programming, developers control how objects are created and managed within the main program.

With IoC, this control shifts to Spring's framework, where an external container oversees the flow and management of objects.

IoC covers tasks like creating objects, managing their lifecycle (including memory management), and injecting them into other parts of the application as needed.

The Spring IoC Container handles everything from creating objects to eventually cleaning them up.

Developers only need to provide two things to the IoC container: the objects themselves (known as beans) and how they're configured (their dependencies).

Dependency Injection (DI)

Dependency injection (DI) is a way to connect different parts of a program so they can work together effectively. In Spring, DI automates this process by handling how objects are connected and provided to each other.

DI helps keep components of a program loosely connected, which makes the code easier to manage and test.

In Spring, the IoC container is responsible for managing these connections, ensuring that each part of the program gets the objects it needs to do its job correctly.

- **Version Information:**
- Current Spring version: 6
- Current Spring Boot version: 3
- Spring Boot 3 works with Spring 6.

