Spring Boot JDBC

Spring Boot JDBC is used to connect the Spring Boot application with **JDBC** by providing libraries and starter dependencies.

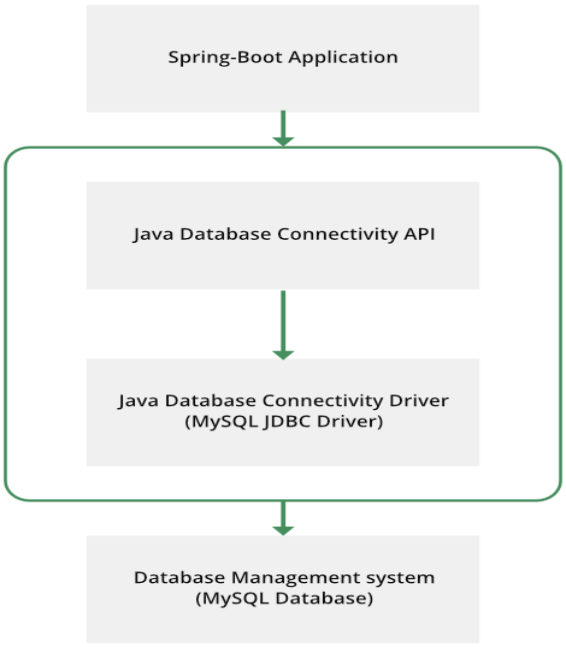
Spring Boot JDBC has a level of control over the SQL queries that are being written.

Spring Boot JDBC has simplified the work of Spring JDBC by automating the work and steps by using the concept of **Auto Configuration**. **JdbcTemplate**, **Data Source,** etc. all required objects are auto configured, and object creation is done by Spring Boot.

As a programmer, you need to auto-wire the **JdbcTemplate** and perform the database operations like **Create, Retrieve, Update, and Delete**. Inputs need to be passed by **application. Properties** or **application.yml file.** In application. Properties file, we configure **Data Source.**

JDBC consists of two parts as depicted below:

- **JDBC interfaces:** *java.sql / javax.sql* packages have classes/interfaces of JDBC API.
- **JDBC drivers:** *JDBC Driver* allows Java programs to interact with the database.



To work with a database using Spring Boot, we need to add the following dependencies:

A. JDBC API

Database connectivity API specifies how the client connects and queries a database.

pom.xml Maven Dependency:

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-jdbc</artifactId>
</dependency>
```

B. MySQL Driver

MySQL JDBC and R2DBC driver to work with the database.

```
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
  <scope>runtime</scope>
</dependency>
```

After this, we will add **datasource** properties in **application.properties** file:

For MySQL database:

```
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:8080/test
spring.datasource.username=root
spring.datasource.password=root
```