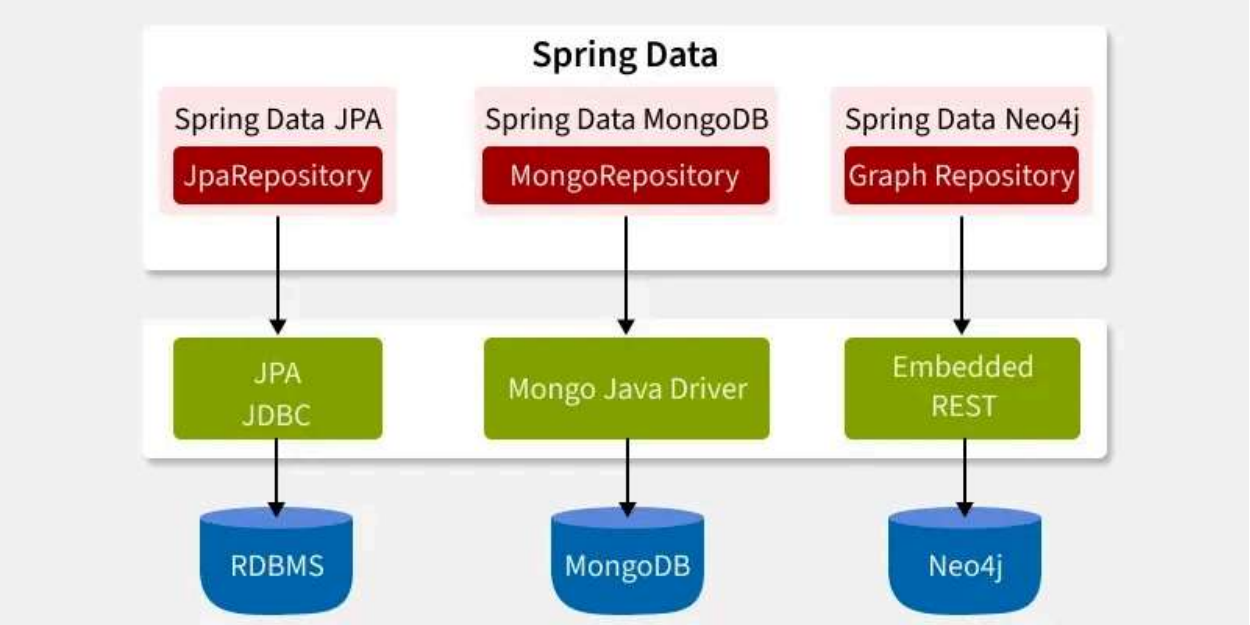
Search...

Introduction to the Spring Data Framework

Last Updated : 28 Aug, 2025

Spring Data is a data access framework in the Spring ecosystem that simplifies database interactions for relational (SQL) and non-relational (NoSQL) databases. It eliminates boilerplate code and provides an easy-to-use abstraction layer for developers working with JPA, MongoDB and more.



Spring-Data

Need for Spring Data Framework

- **Scalability limitations:** Difficult and expensive to scale writes in RDBMS.
- **Performance:** Vertical and horizontal scaling require costly infrastructure.
- **Flexibility concerns:** The schema of RDBMS does not suit dynamic applications.

Spring Data Framework was introduced to bridge the gap between traditional and modern data storage solutions. It provides:

- Unified access to SQL, NoSQL and Big Data technologies.
- Simplified CRUD operations via repositories.
- Automatic query generation and optimized database interactions.
- Scalable and flexible architecture for cloud-based and microservices applications.

Spring Data Framework Components

Spring Data is a parent project that consists of multiple submodules, each designed for specific database types.

1. Relational Database Support (SQL-based)

- **Spring Data JPA:** Simplifies Hibernate and JPA-based persistence.
- **Spring Data JDBC**: Direct JDBC access with minimal overhead.
- **Spring Data R2DBC**: Reactive database access for SQL databases.

2. NoSQL Database Support

- **Spring Data MongoDB**: Document-oriented database with JSON-based storage.
- **Spring Data Redis:** In-memory key-value store with caching support.
- **Spring Data Cassandra**: Highly scalable, distributed NoSQL database.
- **Spring Data Elasticsearch:** Full-text search and analytics database.
- **Spring Data Neo4j**: Graph-based database with complex relationship modeling.

3. Big Data Support

- **Spring Data Hadoop**: Integration with Hadoop ecosystem for large-scale data processing.
- **Spring Data Hive, Pig and Cascading**: Support for distributed computing.

Popular Spring Data Modules

Spring Data includes several widely used modules:

- **Spring Data JPA:** ORM-based persistence layer for SQL databases.
- **Spring Data JDBC:** Lightweight alternative to JPA.
- **Spring Data Redis**: In-memory caching and messaging.
- **Spring Data MongoDB**: NoSQL document storage.
- **Spring Data Elasticsearch**: Advanced search and analytics.
- **Spring Data REST**: Exposes repository-based data as RESTful APIs.

Below modules are under development:

Currently, Spring Data R2DBC (Reactive Relational Database Connectivity) is evolving to provide:

- Reactive repository support
- Asynchronous query execution
- Enhanced SQL and NoSQL integration

Key Terms in Spring Data

1. Repositories

A repository is an abstraction layer between domain models and the database. It enables CRUD operations without writing SQL queries manually.

Steps to Use a Repository:

1. Create a POJO entity representing the data model.
2. Extend JpaRepository, CrudRepository or MongoRepository.
3. Define custom query methods using Spring Data Query Methods.
4. Inject the repository into services and use it for database operations.

2. QueryDSL

A type-safe way to build dynamic queries for SQL, JPA, MongoDB and Lucene.

3. NoSQL Data Models

NoSQL databases store data in key-value, document, column-family or graph-based formats:

- **MongoDB:** Document database using JSON-like BSON format.
- **Redis:** Key-value store supporting Lists, Sets and HashMaps.
- **HBase**: Column-oriented NoSQL database.
- **Neo4j:** Graph-based database for relationship-driven applications

Related Article

- Spring Data JPA
- Spring Data JDBC
- Spring Data R2DBC
- Spring Data MongoDB
- Spring Data Redis
- Spring Data Elasticsearch
- Spring Data REST

GeeksforGeeks
Sanchhaya Education Private Limited
Corporate & Communications Address:
A-143, 7th Floor, Sovereign Corporate Tower, Sector- 136, Noida, Uttar Pradesh (201305)