



CS544 EA

Applications

Spring Data

Spring Data

- We've been writing our own DAOs
 - Seems like we're repeating ourselves everytime
 - Spring can generate DAOs for us
- We'll discuss how Spring Data works
 - Then look at finder methods

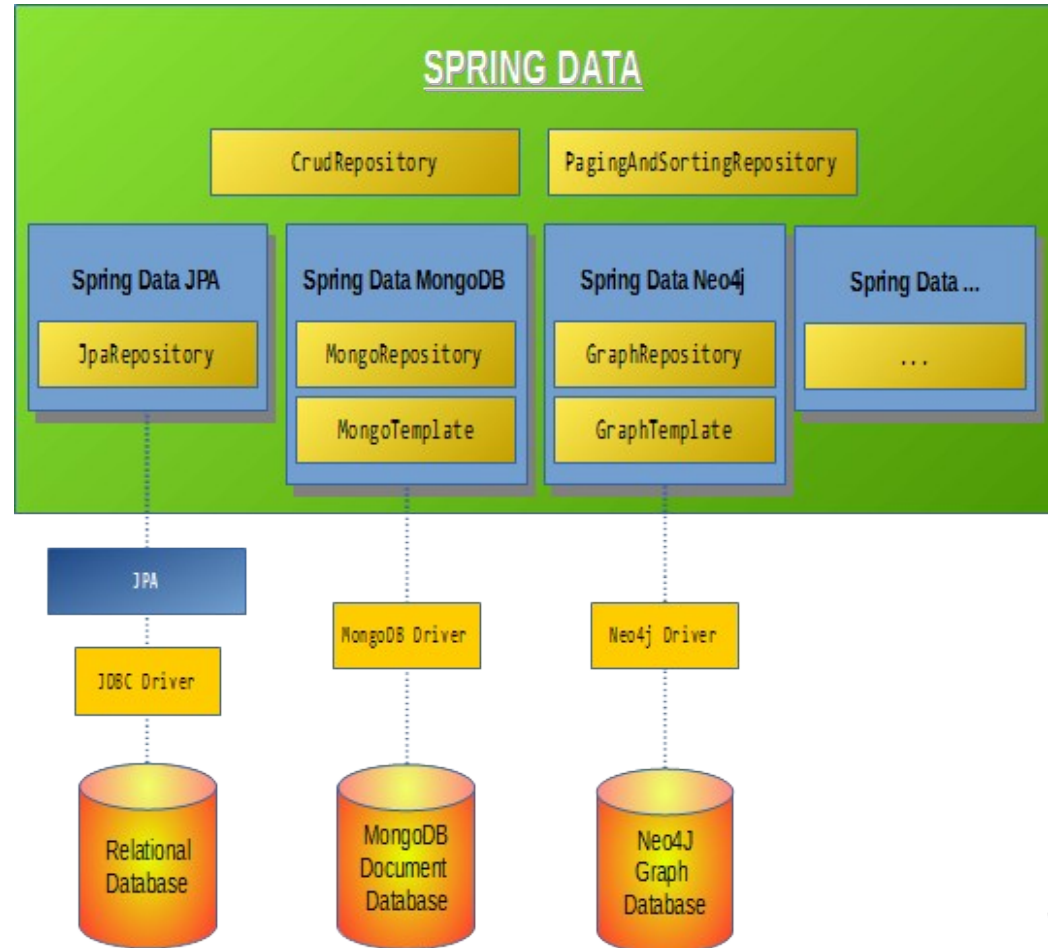


What is Spring Data

- Spring Data offers a flexible abstraction for working with data access frameworks
- Purpose is to **unify and ease the access** to different kinds of persistence stores
- Both relational db systems and NoSQL data stores
- Addresses common difficulties developers face when working with databases in applications
- Spring Data is an umbrella project that provides you with easy to use data access technologies for all kinds of relational and non-relational DBs

Spring Data Modules

- Modules supports:
 - Templating
 - Object/Datastore mapping
 - Repository support
- Every repository offers:
 - CRUD operations
 - Finder Methods
 - Sorting and Pagination
- Spring data module repositories provide generic interfaces:
 - CrudRepository
 - PagingAndSortingRepository



Mapping

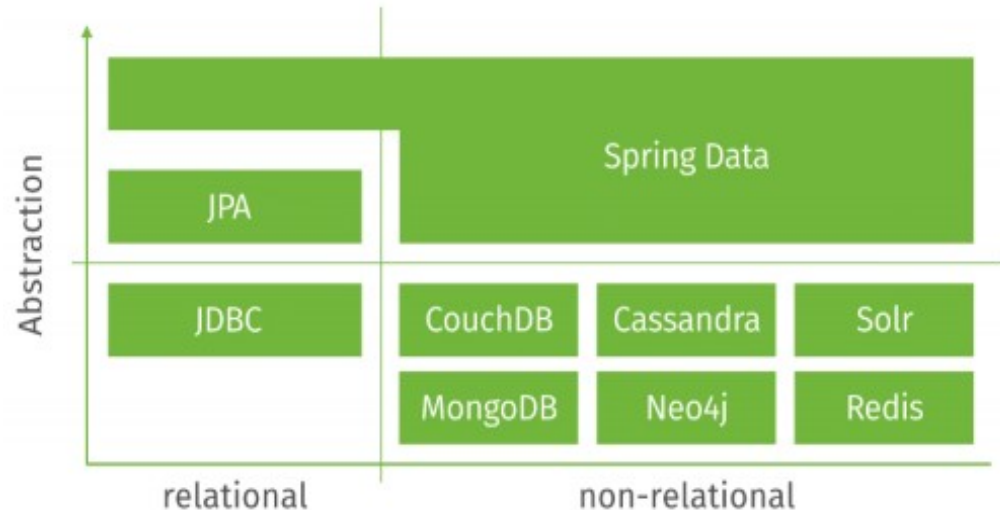
- JPA uses Object Relational Mapping
 - Spring Data extends this concept to NoSQL datastores
- All Spring Data modules provide an Object to datastore mapping
 - Because these data store structures can be so different, there is no common API
 - Each type of datastore has it's own set of annotations to map between objects and datastore structures.

Mapping Examples

JPA	MongoDB	Neo4J
<pre>@Entity @Table(name="USR") public class User { @Id private String id; @Column(name="fn") private String name; private Date lastLogin; ... }</pre>	<pre>@Document(collection="usr") public class User { @Id private String id; @Field("fn") private String name; private Date lastLogin; ... }</pre>	<pre>@NodeEntity public class User { @GraphId Long id; private String name; private Date lastLogin; ... }</pre>

Spring Data Templates

- Each type of DB has its own template that:
 - Connection Management
 - CRUD operations, Queries, Map/Reduce jobs
 - Exception Translation to `DataAccessExceptions`
- No template for JPA
 - It provides most of these already (is an abstraction)



MongoDB Template Example

```
@Configuration
public class MongoConfig extends AbstractMongoConfiguration {

    @Override
    protected String getDatabaseName() {
        return "test";
    }

    @Override
    public MongoClient mongoClient() {
        return new MongoClient("127.0.0.1", 27017);
    }

    @Override
    protected String getMappingBasePackage() {
        return "org.baeldung";
    }
}
```

```
<!-- Connection to MongoDB server -->
<mongo:mongo-client id="mongoClient" host="localhost" />
<mongo:db-factory id="mongoFactory" dbname="test" mongo-ref="mongoClient" />

<!-- MongoDB Template -->
<bean id="mongoTemplate" class="org.springframework.data.mongodb.core.MongoTemplate">
    <constructor-arg name="mongoDbFactory" ref="mongoDbFactory"/>
</bean>
```