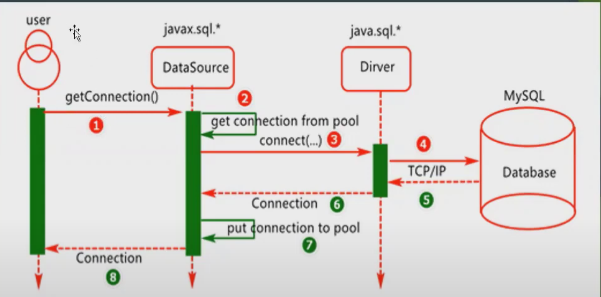
* Persistence layers: layer make it easier for program to persist its state ,By using database management system.

1. DB 🡪 JDBC/SQL
2. Persistence/DAO Tier 🡪 Entity
3. Business logic 🡪DTO
4. Presentation Tier

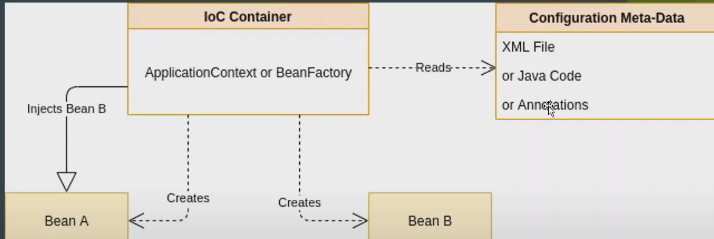
* **Spring JDBC**: framework that simplifies working with databases using JDBC. includes:
  + **DataSource**: Manages connections to the database.
  + **JdbcTemplate**: Provides an easier interface for executing queries.
  + **NamedParameterJdbcTemplate**: Supports named parameters in queries.

C**onfiguration in “application.properties”:**

* Contains settings for database connection such as URL, username, and password.
* **Connection pooling: mechanism** can be enabled to improve performance by reusing connections, allocating new connections, managing available connections, and closing connections.



* Configuration types:
* XML
* Java based
* Annotations



Annotations: metadata provides data (supplemental information) about the program

1. @Autowired: dependency injection 🡪used on setter methods, variables, and constructors. Spring manages & creates the necessary objects.
2. @component: class-level annotation indicate a class is a Spring class(as a Bean)) to be discovered & managed by Spring during the classpath.
3. @componentScan: specify the packages that Spring should scan for components to be managed as Beans.
4. @Configuration: indicate that a class contains one or more definitions to be managed by Spring Controller.
5. @Bean: used inside a class with @configration to indicate. that a method returns an object that should be managed as a Bean by spring.
6. @Service: define a class as a service managed by Spring.
7. @Repository: define a class as a Repository managed by Spring.

ORM: