Day 29 Spring Data JPA in Spring Boot Application.

What is Spring Data JPA.

Adv of Spring Data JPA

Auto Configuration.

DB customization.

Rate Limiting, Paging & Sorting.

Day 28 JPA with Hibernate.

Hibernate is a popular ORM (Object Relational Mapping) – This framework is mainly used to interact with any RDBMS easily.

It uses HQL which is RDBMS independent.

It automatically creates Entities from the Bean Class.

It also helps to configure DB caching, Rate Limiting, Validation & Connection Pooling.

Hibernate can be added to stand-alone and web applications.

With the help of Maven we can add all the dependencies of Hibernate.

Two types of Mapping Configuration for Hibernate.

1. XML based Mapping (employee.hbm.xml – hbm – Hibernate Mapping)
2. Annotation based Mapping ( Add Annotations directly on the Entity Bean class and columns)

JPA Annotations (Jakarta.persistence)

1. Entity
2. Table
3. Column
4. Query
5. NativeQuery
6. NamedQuery
7. Id
8. GeneratedValue
9. OneToOne
10. OneToMany
11. ManyToMany
12. JoinColumn
13. JoinTable

Spring Data JPA with Hibernate.

API Documentation - <https://docs.spring.io/spring-data/commons/docs/current/api/>

findAll () – Simple Select Query ( select \* from tbl\_passenger – Native Query – HQL 🡪 select p from Passenger p)

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Query | Return | Input |
| findAll() – [ReadAll] | Simple Select Query | List<Entity> | N/A |
| findById() – [ReadSpecific] | Select Query with where clause | Optional<Entity> | ID |
| save() – [Insert] | Insert Query | int (No of rows affected) | N/A |
| save() – [update] | Update Query | Int (No of rows affected) | ID |
| deleteById() - | Delete Query | Int (No of rows affected) | ID |

Finding Record by Non-Key properties

findByEmail(String email) -- Optional<Entity>

findByName(String name) -- List<Entity>

findByMobile(long mobile) -- Optional<Entity>