**Demonstrate implementation of Query Methods feature of Spring Data JPA**

Spring Data JPA allows us to create custom queries using **method names**, without writing SQL or JPQL. These method names follow specific patterns that are automatically translated to SQL at runtime.

**Implementation Example :**

**Product.java**

package com.example.demo.entity;

import jakarta.persistence.\*;

import java.time.LocalDate;

@Entity

public class Product {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

private double price;

private LocalDate createdDate;

// Getters and Setters

}

**ProductRepository .java**

package com.example.demo.repository;

import com.example.demo.entity.Product;

import org.springframework.data.jpa.repository.JpaRepository;

import java.time.LocalDate;

import java.util.List;

public interface ProductRepository extends JpaRepository<Product, Long> {

List<Product> findByNameContaining(String keyword);

List<Product> findByNameStartingWith(String prefix);

List<Product> findByPriceGreaterThan(double price);

List<Product> findByPriceLessThan(double price);

List<Product> findByCreatedDateBetween(LocalDate start, LocalDate end);

List<Product> findTop3ByOrderByPriceDesc();

List<Product> findAllByOrderByNameAsc();

}

**ProductController .java**

package com.example.demo.controller;

import com.example.demo.entity.Product;

import com.example.demo.repository.ProductRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.\*;

import java.time.LocalDate;

import java.util.List;

@RestController

@RequestMapping("/products")

public class ProductController {

@Autowired

private ProductRepository repo;

@GetMapping("/search")

public List<Product> search(@RequestParam String keyword) {

return repo.findByNameContaining(keyword);

}

@GetMapping("/starts")

public List<Product> starts(@RequestParam String prefix) {

return repo.findByNameStartingWith(prefix);

}

@GetMapping("/gt")

public List<Product> greater(@RequestParam double price) {

return repo.findByPriceGreaterThan(price);

}

@GetMapping("/lt")

public List<Product> less(@RequestParam double price) {

return repo.findByPriceLessThan(price);

}

@GetMapping("/between")

public List<Product> between(@RequestParam String start, @RequestParam String end) {

return repo.findByCreatedDateBetween(LocalDate.parse(start), LocalDate.parse(end));

}

@GetMapping("/top3")

public List<Product> top3() {

return repo.findTop3ByOrderByPriceDesc();

}

@GetMapping("/sorted")

public List<Product> sorted() {

return repo.findAllByOrderByNameAsc();

}

}

**OUTPUT:**

[

{

"id": 1,

"name": "iPhone",

"price": 95000,

"createdDate": "2024-03-10"

}

]