SESSION 21 PROGRAMS(SPRING DATA JPA)

Spring Data JPA provides the Data Access Layer using **Java Persistence API** and **ORM** implementations like **Hibernate**.

It is used for accessing data for relational databases.

JPA (Java Persistent API) is the sun specification for persisting objects in the enterprise application.

The implementation of JPA specification are provided by many vendors such as:

- Hibernate or Spring Data JPA
- Toplink
- iBatis
- OpenJPA etc.

Advantage of Spring JpaTemplate

You don't need to write the before and after code for persisting, updating, deleting or searching object.

Programs

application.properties file configuration

```
server.port=9090
spring.mvc.view.prefix=/webpages/
spring.mvc.view.suffix=.jsp
spring.h2.console.enabled=true
spring.datasource.platform=h2
spring.datasource.url=jdbc:h2:mem:test;DB_CLOSE_DELAY=-
1;DB_CLOSE_ON_EXIT=FALSE
```

Using CRUDRepository

```java
import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface CustomerInt extends CrudRepository<Customer, Integer>
{

}
```

CustomerController

```java
import java.util.Iterator;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.SessionAttributes;
import org.springframework.web.servlet.ModelAndView;

@Controller
@SessionAttributes({"cid","cname","cemail"})
 public class CustomerController
 {
	@Autowired
	CustomerInt impl;

	@GetMapping("customerlogin")
	public String custLogin()
	{
		return "customerlogin";
	}
	@PostMapping("customerview")
	public ModelAndView custDetails(@RequestParam("cid") String cid,
@RequestParam("cname") String cname, @RequestParam("cemail") String cemail ,
@ModelAttribute Customer cust)
	{
		ModelAndView mv = new ModelAndView();
		mv.addObject("cid", cid);
		mv.addObject("cname", cname);
		mv.addObject("cemail", cemail);
		mv.setViewName("CustomerView");

		impl.save(cust);
		//impl.delete(cust);

		//Fetching Table records as Customer Entities
		System.out.println("No of Entities Available is :"+ impl.count());
		Iterable<Customer> itrbl = impl.findAll();
		Iterator<Customer> itr = itrbl.iterator();
		while(itr.hasNext()) {
			Customer cust1 = itr.next();
			System.out.println("Customer Name is : "+cust1.getCname());
		}

		return mv;

	}

	@GetMapping("customersessionview")
	public String custSession()
	{

		return "CustomerSessionview";
	}

}
```

SESSION 21 ASSIGNMENT

Write a Program to create Employee Controller class , employee jsp , employeeview jsp & employeesessionview jsp , interface EmployeeInt extends CrudRepository<Employee, Integer>

- Write 3 methods in Employee Controller class.
- 1st method mapped to url="employee"
- 2nd method mapped to url="employeeview"
- 3rd method mapped to url="employeesessionview"
- Employee jsp should have fields like Name,Address,Mobile No & submit button.
- Display Request data in employeeview jsp . href to employeesessionview
- Display Session data in employeesessionview jsp .
- In EmployeeController Class use @RequestParam , @ModelAttribute & print the corresponding values in console .

Create @Autowired
EmployeeInt impl; at Controller Class Level

Save , Update , Delete & Fetch records using impl object