**Exercise 7: Employee Management System - Enabling Entity Auditing**

**Business Scenario:**

Implement auditing to track the creation and modification of employees and departments.

**Instructions:**

1. **Entity Auditing:**

Enable auditing in your application by configuring auditing properties.

Use annotations like @CreatedBy, @LastModifiedBy, @CreatedDate, and @LastModifiedDate.

package com.company.entity;

import jakarta.persistence.\*;

import lombok.\*;

import org.springframework.data.annotation.CreatedBy;

import org.springframework.data.annotation.LastModifiedBy;

import org.springframework.data.annotation.CreatedDate;

import org.springframework.data.annotation.LastModifiedDate;

import org.springframework.data.jpa.domain.support.AuditingEntityListener;

import java.time.LocalDateTime;

@NamedQueries({

@NamedQuery(

name = "Employee.findByEmail",

query = "SELECT e FROM Employee e WHERE e.email = :email"

),

@NamedQuery(

name = "Employee.findByDeptName",

query = "SELECT e FROM Employee e WHERE e.department.name = :deptName"

)

})

@Entity

@EntityListeners(AuditingEntityListener.class)

@Table(name = "employees")

@Data

@NoArgsConstructor

@AllArgsConstructor

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

private String email;

@ManyToOne

@JoinColumn(name="department\_id")

private Department department;

@CreatedDate

private LocalDateTime createdDate;

@LastModifiedDate

private LocalDateTime modifiedDate;

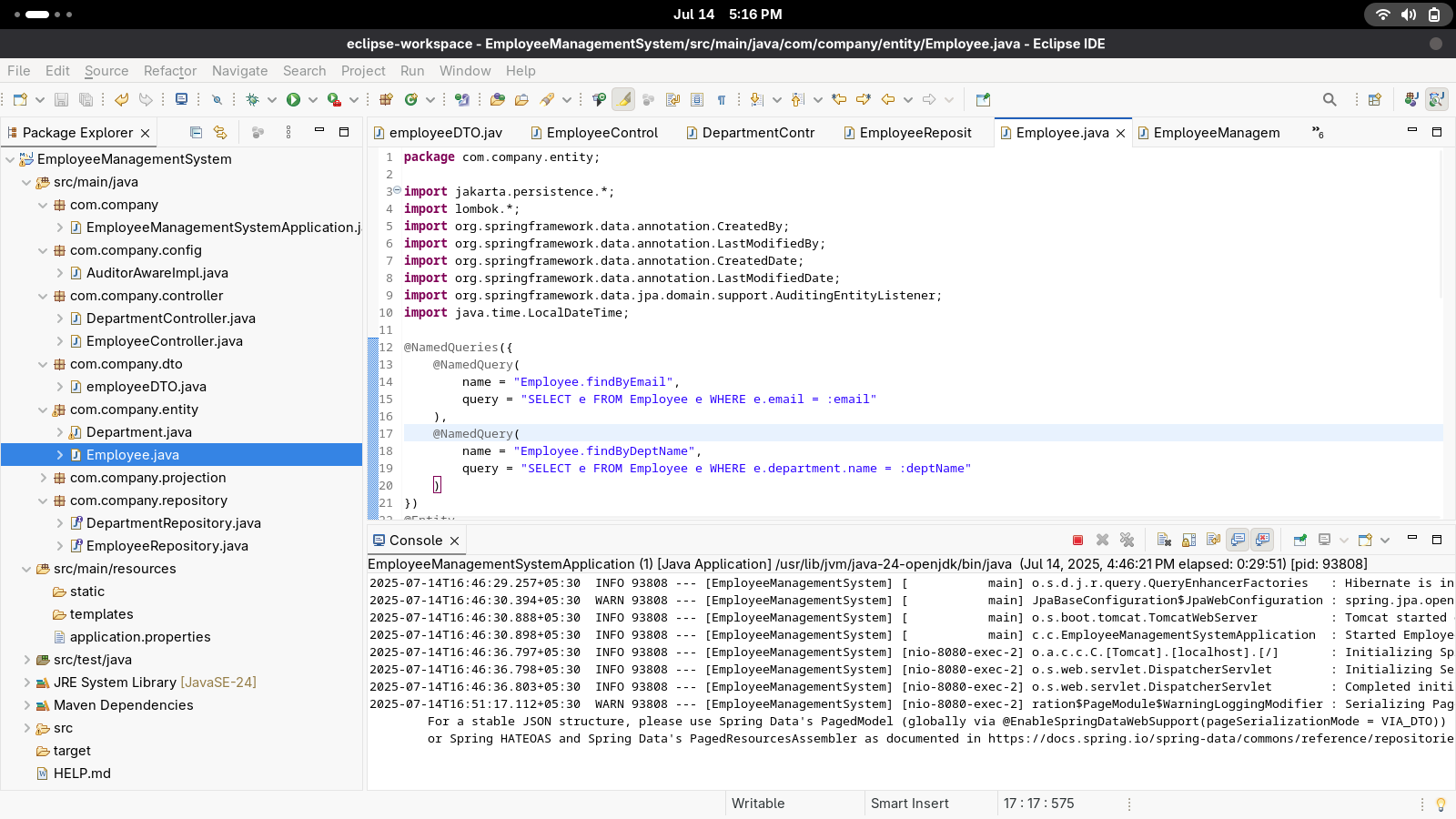
@CreatedBy

private String createdBy;

@LastModifiedBy

private String modifiedBy;

}



package com.company.entity;

import jakarta.persistence.\*;

import lombok.\*;

import org.springframework.data.annotation.CreatedBy;

import org.springframework.data.annotation.LastModifiedBy;

import org.springframework.data.annotation.CreatedDate;

import org.springframework.data.annotation.LastModifiedDate;

import org.springframework.data.jpa.domain.support.AuditingEntityListener;

import com.fasterxml.jackson.annotation.JsonIgnore;

import java.time.LocalDateTime;

import java.util.List;

@Entity

@EntityListeners(AuditingEntityListener.class)

@Table(name = "departments")

@Data

@NoArgsConstructor

@AllArgsConstructor

public class Department {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private long id;

private String name;

@OneToMany(mappedBy = "department", cascade = CascadeType.ALL)

@JsonIgnore

private List<Employee> employees;

@LastModifiedDate

private LocalDateTime modifiedDate;

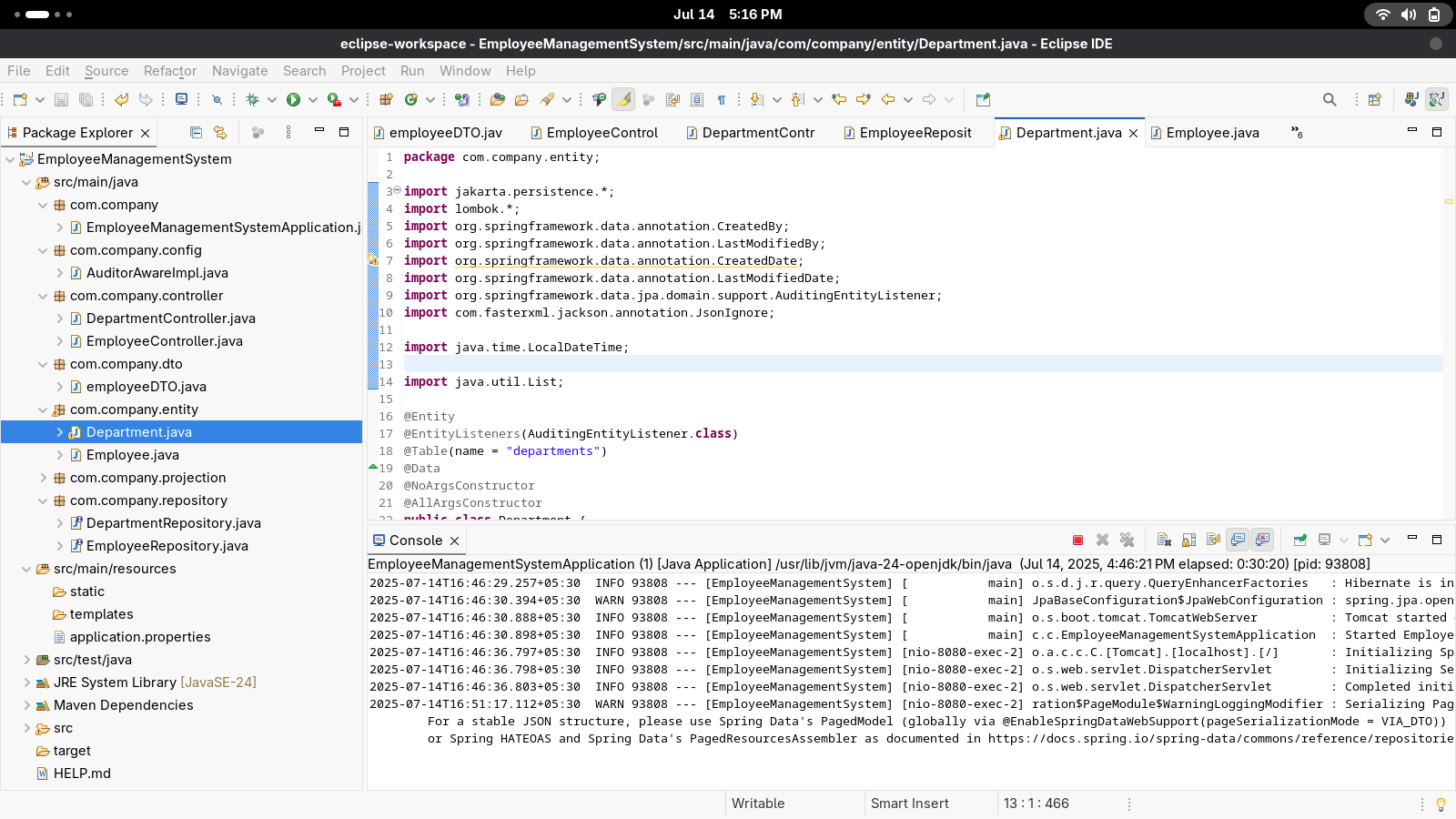
@CreatedBy

private String createdBy;

@LastModifiedBy

private String modifiedBy;

}



package com.company.config;

import org.springframework.data.domain.AuditorAware;

import org.springframework.stereotype.Component;

import java.util.Optional;

@Component

public class AuditorAwareImpl implements AuditorAware<String> {

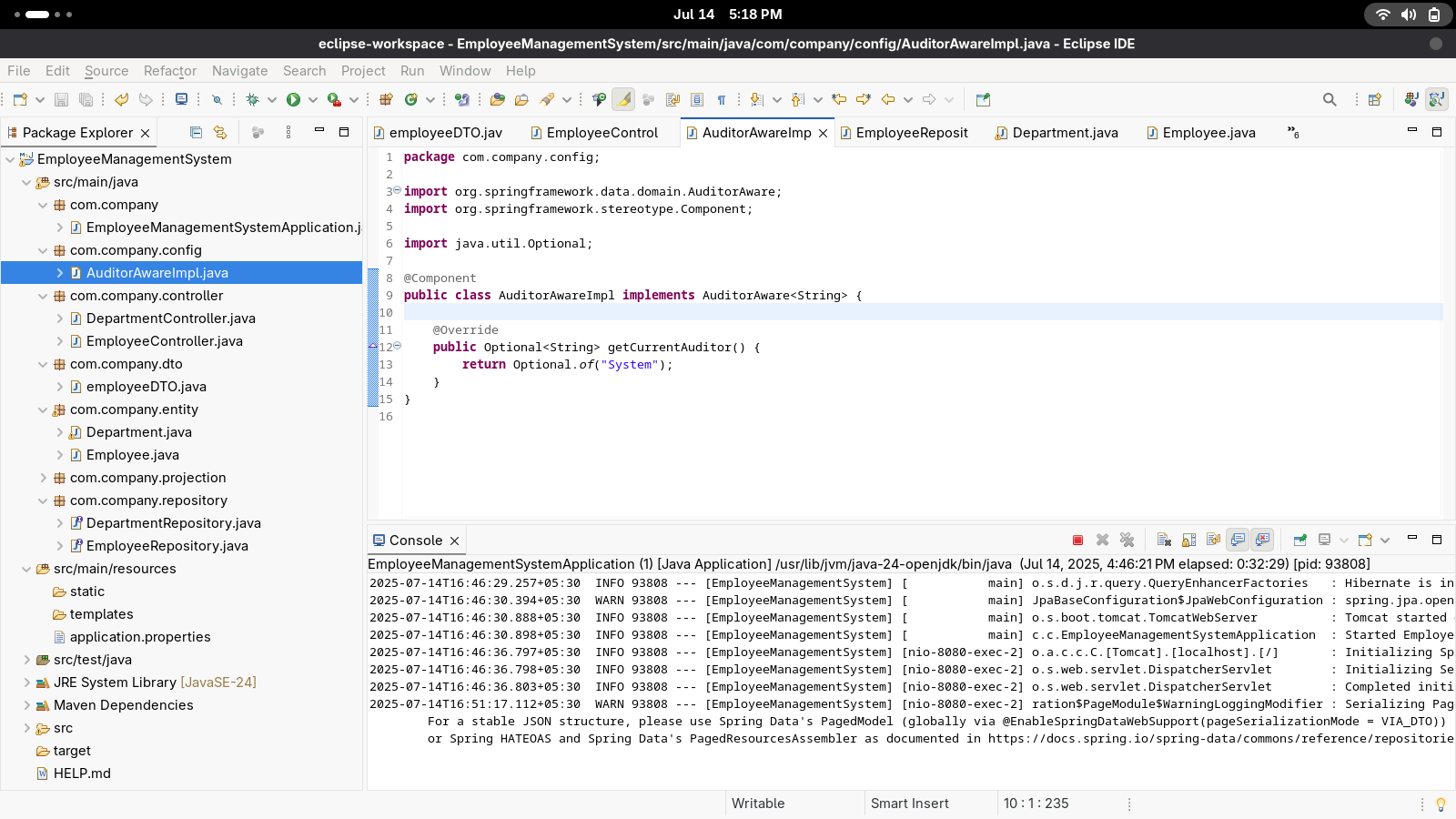
@Override

public Optional<String> getCurrentAuditor() {

return Optional.of("System");

}

}



{

"content": [

{

"id": 1,

"name": "Alice Johnson",

"email": "alice@example.com",

"department": {

"id": 1,

"name": "IT",

"modifiedDate": "2025-07-14T16:46:37.181301",

"createdBy": "System",

"modifiedBy": "System"

},

"createdDate": "2025-07-14T16:48:52.196217",

"modifiedDate": "2025-07-14T16:48:52.196217",

"createdBy": "System",

"modifiedBy": "System"

},

{

"id": 2,

"name": "Bob Smith",

"email": "bob@example.com",

"department": {

"id": 1,

"name": "IT",

"modifiedDate": "2025-07-14T16:46:37.181301",

"createdBy": "System",

"modifiedBy": "System"

},

"createdDate": "2025-07-14T16:50:56.254986",

"modifiedDate": "2025-07-14T16:50:56.254986",

"createdBy": "System",

"modifiedBy": "System"

}

],

"pageable": {

"pageNumber": 0,

"pageSize": 2,

"sort": {

"sorted": true,

"empty": false,

"unsorted": false

},

"offset": 0,

"paged": true,

"unpaged": false

},

"totalPages": 1,

"totalElements": 2,

"last": true,

"size": 2,

"number": 0,

"numberOfElements": 2,

"sort": {

"sorted": true,

"empty": false,

"unsorted": false

},

"first": true,

"empty": false

}