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

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

* **How do you enable auditing in a Spring Boot application?**

You enable auditing by adding the @*EnableJpaAuditing* annotation in the main application class and configuring an AuditorAware bean to define who the current auditor (user) is. Additionally, you must use the necessary annotations (@*CreatedBy*, @*LastModifiedBy*, @*CreatedDate*, @*LastModifiedDate*) on your entities to track creation and modification metadata.

* **What annotations are used for auditing?**

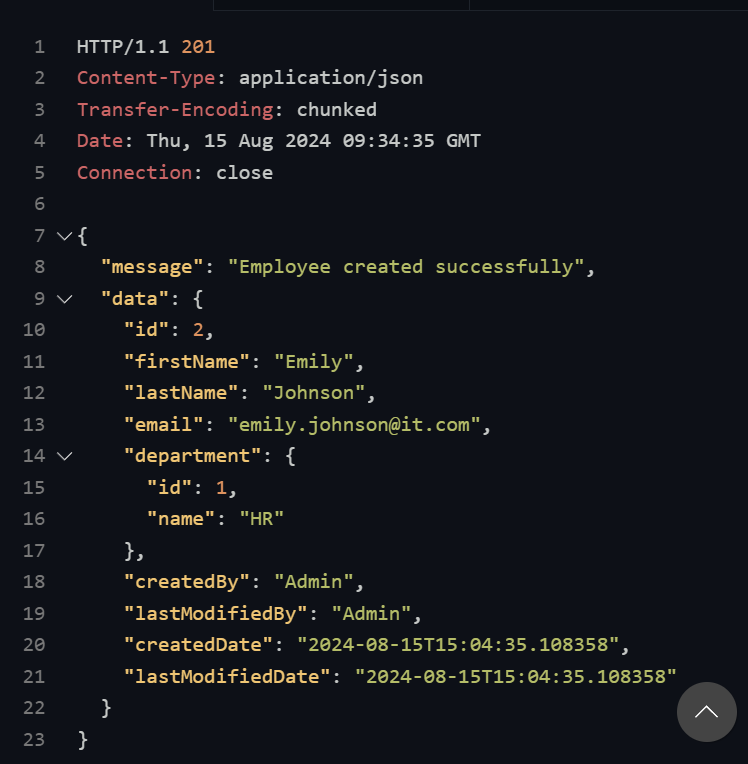
The annotations used for auditing are @*CreatedBy*, @*LastModifiedBy*, @*CreatedDate*, and @*LastModifiedDate*. These annotations help in capturing the user responsible for creating or modifying the entity and the respective timestamps.

* **How does Spring Boot manage auditing fields automatically?**

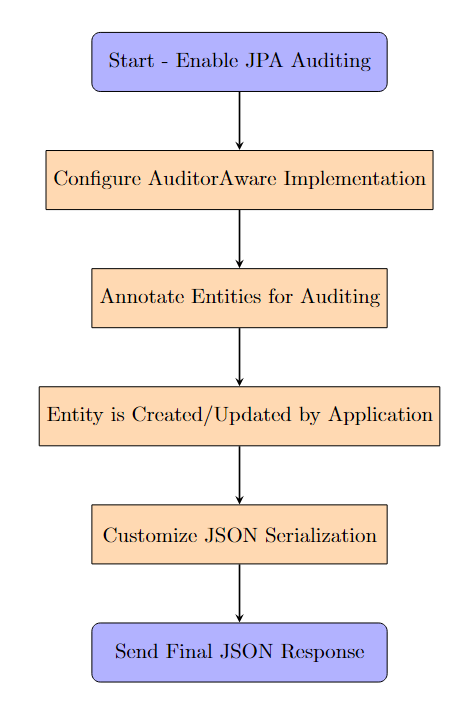
Spring Boot manages auditing fields automatically by using JPA entity listeners. When an entity is persisted or updated, the auditing fields are populated with the current user and the current timestamp, using the *AuditorAware* implementation to fetch the current user.

* **How do you prevent certain fields from being serialized in the JSON response?**

To prevent specific fields from being serialized in the JSON response, you can use the @*JsonIgnore* annotation from the Jackson library. This annotation ensures that the fields are excluded from the JSON output when the entity is converted to a JSON response.

* **The Output of the code:**

The JSON response indicates that an employee named Emily Johnson was successfully created. The response includes her basic information (*ID, first name, last name, email*) and the department details (*ID, name*). Additionally, it shows audit fields (*createdBy, lastModifiedBy, createdDate, lastModifiedDate*) for both the employee and department, with "Admin" as the user who created and last modified the record.

* **The Flowchart of the program:**

**Enable JPA Auditing**: Start by enabling JPA auditing in the main application using @*EnableJpaAuditing*.

**Configure AuditorAware**: Implement and configure *AuditorAware* to define the current user responsible for entity changes.

**Annotate Entities**: Use auditing annotations (@*CreatedBy*, @*LastModifiedBy*, @*CreatedDate*, @*LastModifiedDate*) on the *Employee* and *Department* entities to automatically track changes.

**Entity Lifecycle Management**: When an entity is created or updated, Spring Data JPA automatically fills in the auditing fields.

**Customize JSON Response**: Finally, control which fields are included in the JSON response using @*JsonIgnore* annotations.