Java Interview Series #6

@Entity V/S @Data

UNDERSTAND HOW JPA MAPS YOUR CLASSES, AND HOW LOMBOK CUTS THE CLUTTER.

DIVYAM KUMAR

@Entity – JPA Annotation

- Purpose: Marks the class as a JPA entity, meaning it will be mapped to a table in your database.
- When to use: When the class represents a table in your database and will be managed by Hibernate or another JPA provider

Example:

```
import jakarta.persistence.Entity;
import jakarta.persistence.Id;

@Entity
public class Movie {
    @Id
    private Long id;
    private String name;
}
```

DIVYAM KUMAR

@Data — Lombok Annotation

Purpose: Automatically generates boilerplate code like:

- Getters & Setters
- toString()
- equals() and hashCode()
- Constructor for final fields

When to use: When you want to save time writing boilerplate code in your Java classes (like DTOs or Entities).

Example:

```
import lombok.Data;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;

@Data
@Entity
public class Movie {
    @Id
    private Long id;
    private String name;
}
```

DIVYAM KUMAR

🚀 When to Use Them Together

- Use *@Entity* to map your class to a DB table.
- Use *@Data* to avoid writing boilerplate (especially useful in entities and DTOs).

✅ **Use Case:** Define a User entity for a database table with minimal boilerplate

```
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import lombok.Data;
```

```
@Entity        // Maps this class to a DB table named "user"
@Data          // Generates getters, setters, toString, equals, and hashCode
public class User {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;

    private String email;
}
```

🔍 **Explanation**

@Entity: Tells JPA that this class should be mapped to a database table.

@Data: From Lombok, auto-generates:

- Getters and setters
- *toString()*
- *equals()* and *hashCode()*
- *RequiredArgsConstructor()* if no other constructor is defined

Let's Connect! 🚀

Thanks for joining me on this learning journey!

Ready to take your skills to the next level? 💪

Share your thoughts, questions, or tips below!

Let's keep the conversation alive! **Repost** to help others join the learning journey! 🌱

DIVYAM KUMAR