

## 09 - Loading Data and Entity

### 👉 Entity Creation with JobPost:

- **@Entity:** Specifies that the JobPost class is an entity mapped to a database table.
- **@Id:** Identifies the primary key of the entity, which in this case is the postId field.
- **@Data:** Generates getters, setters, equals(), hashCode(), toString(), and other methods automatically.
- **@NoArgsConstructor** and **@AllArgsConstructor:** Provide a default constructor and an all-arguments constructor, simplifying object creation and injection.

```
@Data
@NoArgsConstructor
@AllArgsConstructor
@Entity
public class JobPost {

    @Id
    private int postId;
    private String postProfile;
    private String postDesc;
    private Integer reqExperience;
    private List<String> postTechStack;

}
```

### 👉 Database Configuration in application.properties:

This configuration file connects the application to a PostgreSQL database and manages database properties.

```
spring.datasource.url=jdbc:postgresql://localhost:5432/telusko
spring.datasource.username=postgres
spring.datasource.password=Lumia@540
spring.datasource.driver-class-name=org.postgresql.Driver
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
```

**Code Link :**

<https://github.com/navinreddy20/spring6-course/tree/c6690e4f2c70d8f530d70623f13d14ff0ffd7e7d/9%20Spring%20DataJPA/9.9%20Loading%20Data%20And%20Entity>