

Spring Boot - Adding Accounts Model in Spring Boot

Add Model -> Account.java.

In order to create database table, we have to mark as entity -> @Entity

The Account.java has One to Many relationship. Need to create a relationship. Use below entity.

```
@OneToMany(mappedBy = "account")
```

```
@ManyToOne
```

Creates new column:-

```
@JoinColumn(name = "account_id", referencedColumnName = "id", nullable = false)
```

Account.java

```
@OneToMany(mappedBy = "account")  
private List<Post> posts;
```

Relationship
↓

Post.java

```
@ManyToOne  
@JoinColumn(name = "account_id", referencedColumnName = "id", nullable = false)  
private Account account;
```

Account.java

```
package org.studyeasy.SpringStarter.models;  
  
import java.util.List;  
import jakarta.persistence.Entity;  
import jakarta.persistence.GeneratedValue;  
import jakarta.persistence.GenerationType;  
import jakarta.persistence.Id;  
import jakarta.persistence.OneToMany;  
import lombok.Getter;  
import lombok.NoArgsConstructor;  
import lombok.Setter;  
  
@Entity  
@Getter  
@Setter  
@NoArgsConstructor  
public class Account {  
  
    @Id  
    @GeneratedValue(strategy = GenerationType.SEQUENCE)  
    private Long id;  
  
    private String email;  
  
    private String password;  
  
    private String firstname;  
  
    @OneToMany(mappedBy = "account")  
    private List<Post> posts;  
  
}
```

Post.java

```
package org.studyeasy.SpringStarter.models;

import java.time.LocalDateTime;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.JoinColumn;
import jakarta.persistence.ManyToOne;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;

@Entity
@Getter
@Setter
@NoArgsConstructor
public class Post {

    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE)

    private Long id;

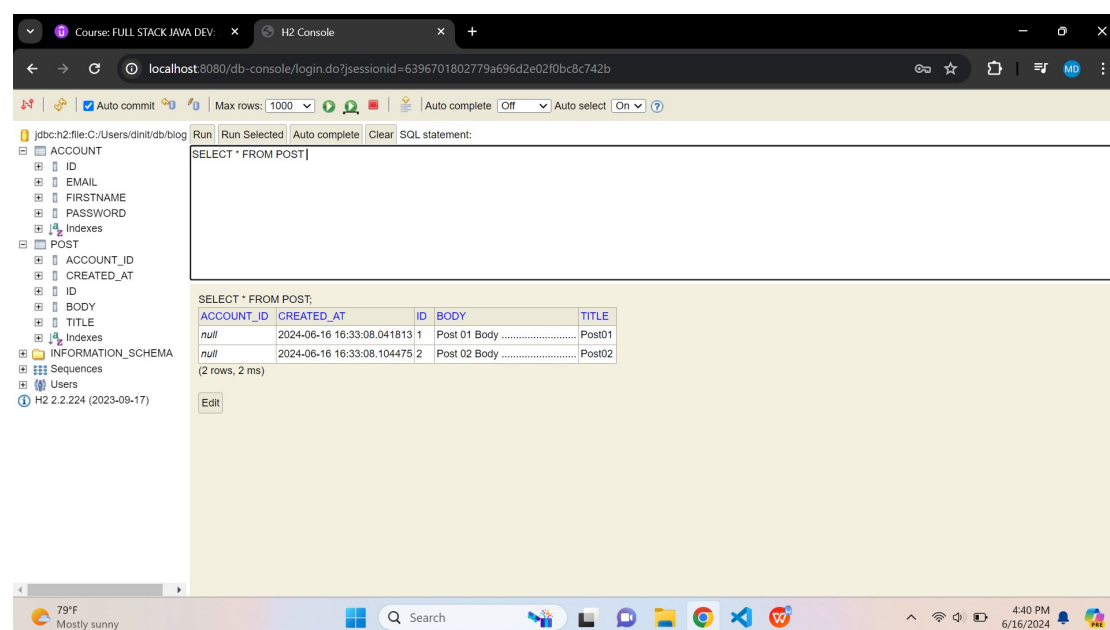
    private String title;

    @Column(columnDefinition = "TEXT")
    private String body;

    private LocalDateTime createdAt;

    @ManyToOne
    @JoinColumn(name = "account_id", referencedColumnName = "id", nullable = true)
    private Account account;
}
```

Output:-



The screenshot shows the H2 Console interface with a database connection to 'jdbc:h2:file:C:/Users/dinit/db/blog'. The left sidebar displays the database schema, including tables like ACCOUNT, POST, INFORMATION_SCHEMA, Sequences, and Users. The main area shows the SQL statement 'SELECT * FROM POST;' and its results. The results are displayed in a table with columns: ACCOUNT_ID, CREATED_AT, ID, BODY, and TITLE. Two rows of data are shown, representing two posts. The first row has a null account_id, a timestamp of 2024-06-16 16:33:08.041813, an id of 1, and a title of 'Post01'. The second row has a null account_id, a timestamp of 2024-06-16 16:33:08.104475, an id of 2, and a title of 'Post02'. The results are summarized as '(2 rows, 2 ms)'.

ACCOUNT_ID	CREATED_AT	ID	BODY	TITLE
null	2024-06-16 16:33:08.041813	1	Post 01 Body	Post01
null	2024-06-16 16:33:08.104475	2	Post 02 Body	Post02

(2 rows, 2 ms)