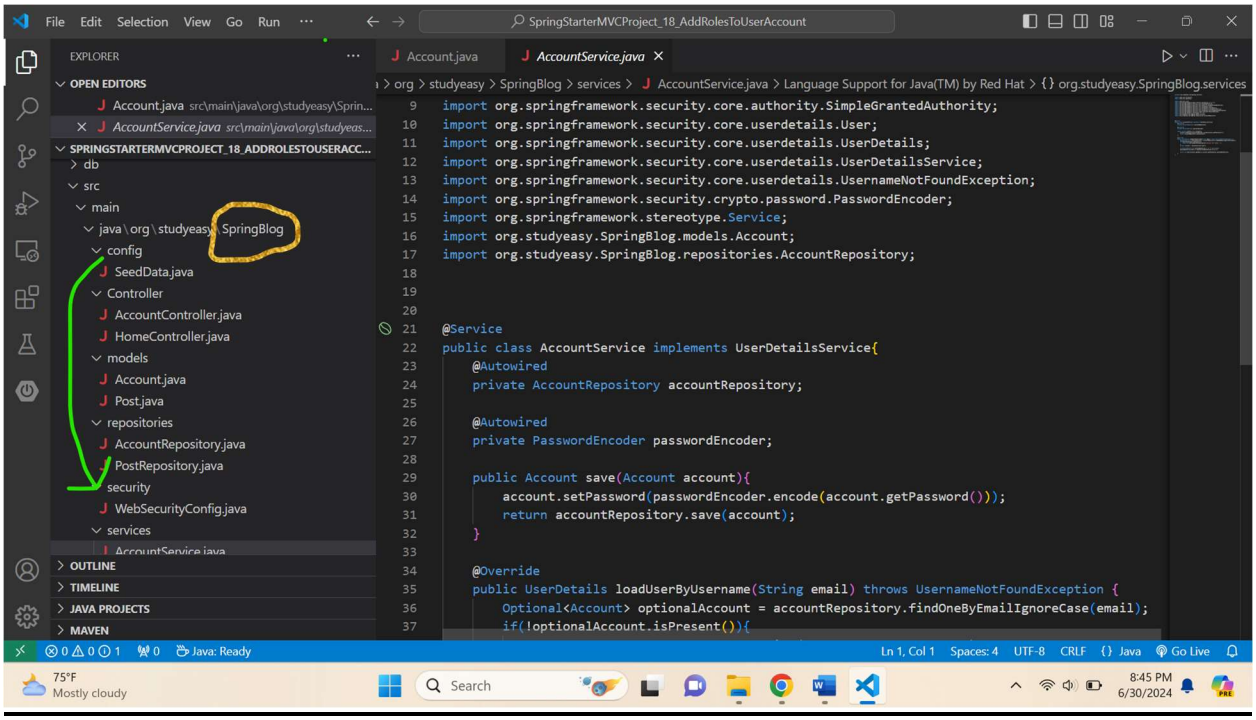
# SpringBoot – Add Roles To User Account

# Change SpringStarter to SpringBlog




**Move WebConfig.java from config folder to security folder.**

**Create new folder -> utils/constant -> Then new file ->**

**1. Roles.java**

**2. Authorities.java**

**Make it enum.**

## Roles.java

```
package org.studyeasy.SpringBlog.utils.constant;

public enum Roles {
    USER("ROLE_USER"),ADMIN("ROLE_ADMIN"),EDITOR("ROLE_EDITOR");
    private String role;
    private Roles(String role) {
        this.role = role;
```

```
    }
    public String getRole() {
        return role;
    }
}
```

## In AccountService.java

```
 public Account save(Account account){
        account.setPassword(passwordEncoder.encode(account.getPassword()));
        account.setRole(Roles.USER.getRole());
        return accountRepository.save(account);
    }
```

## Add this line ->

```
account.setRole(Roles.USER.getRole());
```

## Final Code of AccountService.java

```
package org.studyeasy.SpringBlog.services;

import java.util.ArrayList;
import java.util.Optional;

import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.GrantedAuthority;
import org.springframework.security.core.authority.SimpleGrantedAuthority;
import org.springframework.security.core.userdetails.User;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.core.userdetails.UsernameNotFoundException;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.stereotype.Service;
import org.studyeasy.SpringBlog.models.Account;
import org.studyeasy.SpringBlog.repositories.AccountRepository;
import org.studyeasy.SpringBlog.utils.constant.Roles;
```

```
@Service
public class AccountService implements UserDetailsService{
    @Autowired
    private AccountRepository accountRepository;

    @Autowired
    private PasswordEncoder passwordEncoder;

    public Account save(Account account){
        account.setPassword(passwordEncoder.encode(account.getPassword()));
        account.setRole(Roles.USER.getRole());
        return accountRepository.save(account);
    }

    @Override
    public UserDetails loadUserByUsername(String email) throws
UsernameNotFoundException {
        Optional<Account> optionalAccount =
accountRepository.findOneByEmailIgnoreCase(email);
        if(!optionalAccount.isPresent()){
            throw new UsernameNotFoundException("Account not found!..");
        }
        Account account = optionalAccount.get();

        List<GrantedAuthority> grantedAuthority = new ArrayList<>();
        grantedAuthority.add(new SimpleGrantedAuthority(account.getRole()));

        return new
User(account.getEmail(),account.getPassword(),grantedAuthority);
    }
}
```

## Final Code of Account.java

```
package org.studyeasy.SpringBlog.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;
```

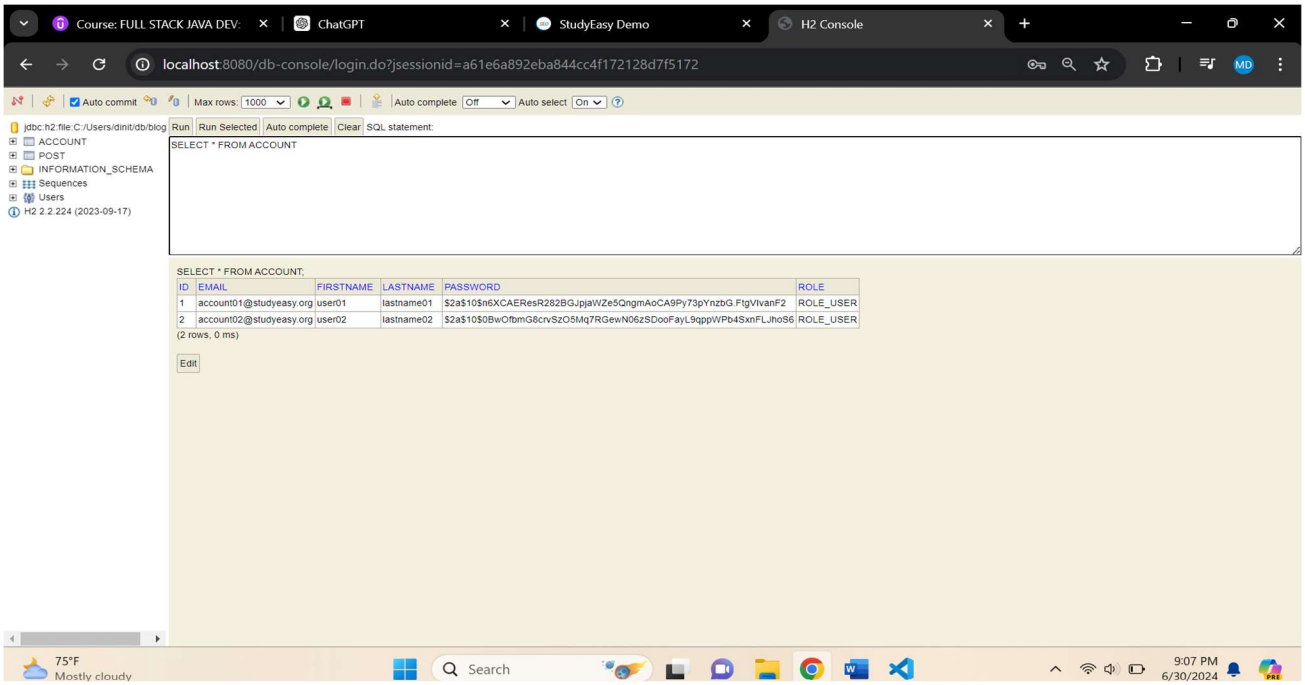
```java
import lombok.NoArgsConstructor;
import lombok.Setter;

@Entity
@Getter
@Setter
@NoArgsConstructor
public class Account {
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE)
    private Long id;

    private String email;

    private String password;

    private String firstname;

    private String lastname;

    private String role;

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

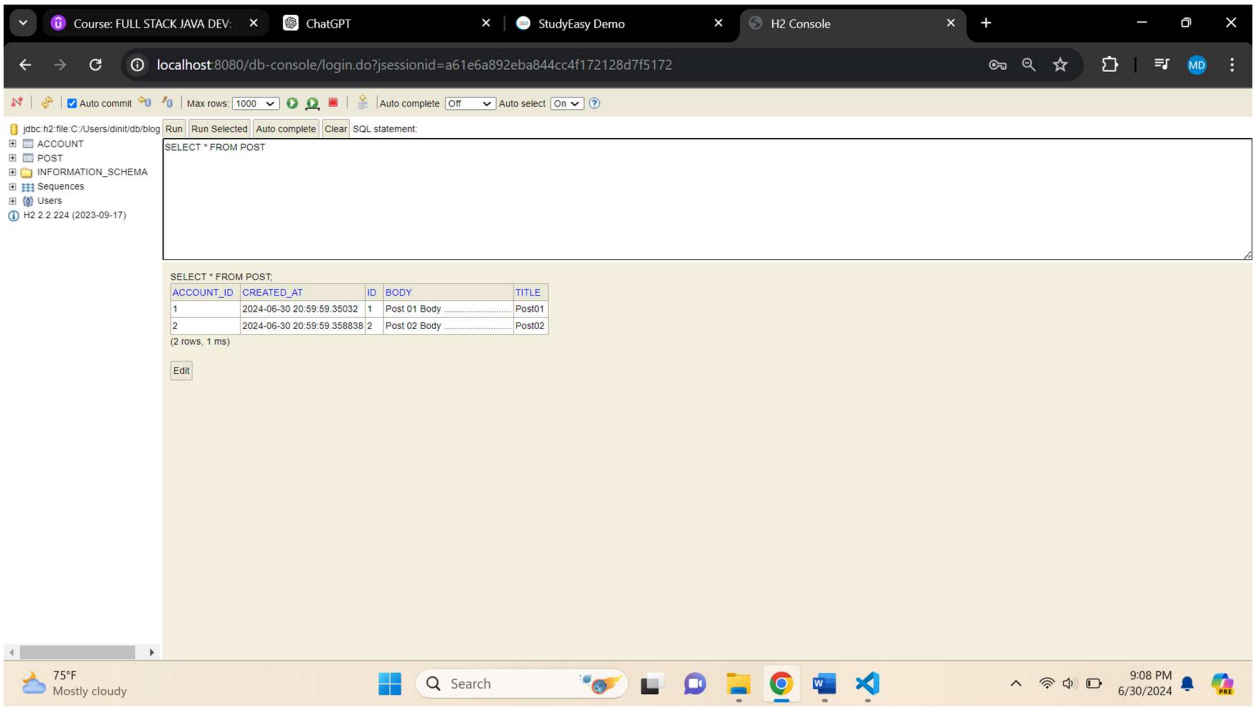
## Modify Seed Data.java -> Add below line

```java
account01.setLastname("lastname01");
```

```java
account02.setLastname("lastname02");
```

# Output:-


Course: FULL STACK JAVA DEV:
ChatGPT
StudyEasy Demo
H2 Console
localhost:8080/db-console/login.do?jsessionid=a61e6a892eba844cc4f172128d7f5172
Auto commit
Max rows: 1000
Auto complete Off
Auto select On
jdbc:h2:file:C:/Users/dinit/db/blog
Run
Run Selected
Auto complete
Clear
SQL statement:
ACCOUNT
POST
INFORMATION_SCHEMA
Sequences
Users
H2 2.2.224 (2023-09-17)
SELECT * FROM ACCOUNT
SELECT * FROM ACCOUNT;
ID EMAIL FIRSTNAME LASTNAME PASSWORD ROLE
1 account01@studyeasy.org user01 lastname01 $2a$10$n6XCAEResR282BGJpjaWZe5QngmAoCA9Py73pYnzbG.FtgVlvanF2 ROLE_USER
2 account02@studyeasy.org user02 lastname02 $2a$10$0BwOfbmG8crvSzO5Mq7RGewN06zSDooFayL9qppWPb4SxnFLJhoS6 ROLE_USER
(2 rows, 0 ms)
Edit
75°F
Mostly cloudy
Search
9:07 PM
6/30/2024


Course: FULL STACK JAVA DEV:
ChatGPT
StudyEasy Demo
H2 Console
localhost:8080/db-console/login.do?jsessionid=a61e6a892eba844cc4f172128d7f5172
Auto commit
Max rows: 1000
Auto complete Off
Auto select On
jdbc:h2:file:C:/Users/dinit/db/blog
Run
Run Selected
Auto complete
Clear
SQL statement:
ACCOUNT
POST
INFORMATION_SCHEMA
Sequences
Users
H2 2.2.224 (2023-09-17)
SELECT * FROM POST
SELECT * FROM POST;
ACCOUNT_ID CREATED_AT ID BODY TITLE
1 2024-06-30 20:59:59.35032 1 Post 01 Body ..... Post01
2 2024-06-30 20:59:59.358838 2 Post 02 Body ..... Post02
(2 rows, 1 ms)
Edit
75°F
Mostly cloudy
Search
9:08 PM
6/30/2024