**SpringBoot\_Add Authorities To UserAccount**

**Authority.java**

package org.studyeasy.SpringBlog.models;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import lombok.Getter;

import lombok.NoArgsConstructor;

import lombok.Setter;

@Entity

@Getter

@Setter

@NoArgsConstructor

public class Authority {

    @Id

    private Long id;

    private String name;

}

**AuthorityRepository.java**

package org.studyeasy.SpringBlog.repositories;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

import org.studyeasy.SpringBlog.models.Authority;

@Repository

public interface AuthorityRepository extends JpaRepository<Authority, Long>{

}

**AuthorityService.java**

package org.studyeasy.SpringBlog.services;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.studyeasy.SpringBlog.models.Authority;

import org.studyeasy.SpringBlog.repositories.AuthorityRepository;

@Service

public class AuthorityService {

    @Autowired

    private AuthorityRepository authorityRepository;

    public Authority save(Authority authority){

        return authorityRepository.save(authority);

    }

}

**Authorities.java -> Enum**

package org.studyeasy.SpringBlog.utils.constant;

public enum Authorities {

    RESET\_ANY\_USER\_PASSWORD(1l, "RESET\_ANY\_USER\_PASSWORD"),

    ACCESS\_ADMIN\_PANEL(2l, "ACCESS\_ADMIN\_PANEL");

    private Long authorityId;

    private String authorityString;

    private Authorities(Long authorityId, String authorityString) {

        this.authorityId = authorityId;

        this.authorityString = authorityString;

    }

    public Long getAuthorityId() {

        return authorityId;

    }

    public String getAuthorityString() {

        return authorityString;

}

}

**Updated SeedData.java Code**

package org.studyeasy.SpringBlog.config;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.CommandLineRunner;

import org.springframework.stereotype.Component;

import org.studyeasy.SpringBlog.models.Account;

import org.studyeasy.SpringBlog.models.Authority;

import org.studyeasy.SpringBlog.models.Post;

import org.studyeasy.SpringBlog.services.AccountService;

import org.studyeasy.SpringBlog.services.AuthorityService;

import org.studyeasy.SpringBlog.services.PostService;

import org.studyeasy.SpringBlog.utils.constant.Authorities;

@Component

public class SeedData implements CommandLineRunner {

    @Autowired

    private PostService postService;

    @Autowired

    private AccountService accountService;



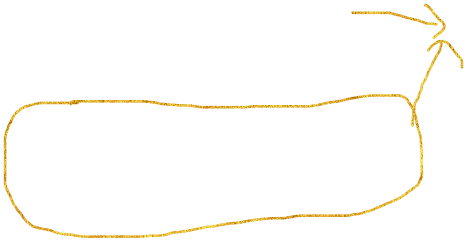
    @Autowired

    private AuthorityService authorityService;



    @Override

    public void run(String... args) throws Exception {

        for (Authorities auth : Authorities.values()) {

            Authority authority = new Authority();

            authority.setId(auth.getAuthorityId());

            authority.setName(auth.getAuthorityString());

            authorityService.save(authority);

        }

        // Account

        Account account01 = new Account();

        Account account02 = new Account();

        account01.setEmail("account01@studyeasy.org");

        account01.setPassword("password");

        account01.setFirstname("user01");

        account01.setLastname("lastname01");

        account02.setEmail("account02@studyeasy.org");

        account02.setPassword("password");

        account02.setFirstname("user02");

        account02.setLastname("lastname02");

        accountService.save(account01);

        accountService.save(account02);

        // Post

        List<Post> posts = postService.getAll();

        if (posts.size() == 0) {

            Post post01 = new Post();

            post01.setTitle("Post01");

            post01.setBody("Post 01 Body ..........................");

            // added

            post01.setAccount(account01);

            postService.save(post01);

            Post post02 = new Post();

            post02.setTitle("Post02");

            post02.setBody("Post 02 Body ..........................");

            // added

            post02.setAccount(account02);

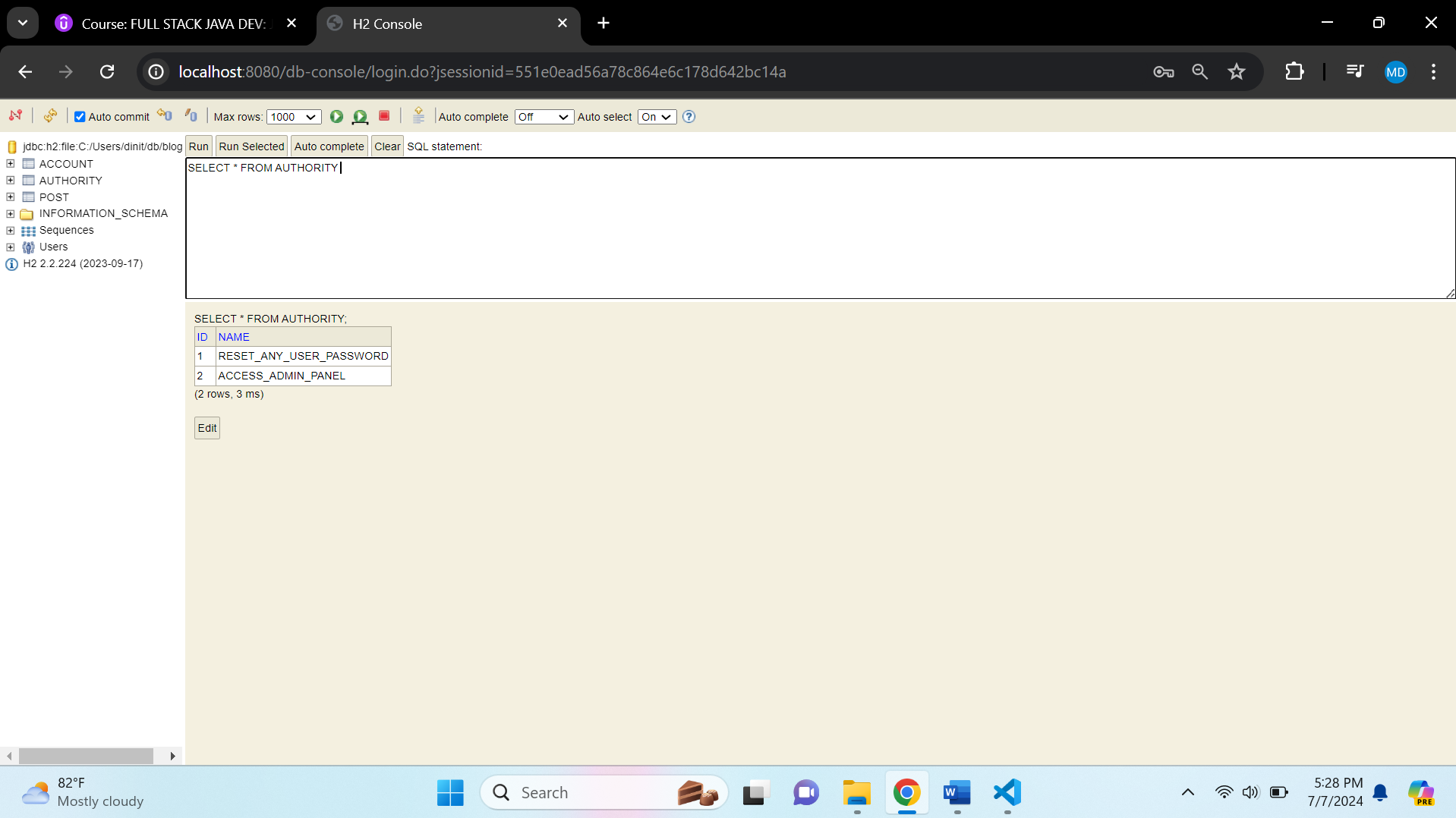
            postService.save(post02);

        }

    }

}

**Output:-**

****

**Rename Authorities.java to Privilages.java**

**Privilages.java**

package org.studyeasy.SpringBlog.utils.constant;

public enum Privilages {

    RESET\_ANY\_USER\_PASSWORD(1l, "RESET\_ANY\_USER\_PASSWORD"),

    ACCESS\_ADMIN\_PANEL(2l, "ACCESS\_ADMIN\_PANEL");

    private Long id;

    private String privillage;

    private Privilages(Long id, String privillage) {

        this.id = id;

        this.privillage = privillage;

    }

    public Long getId() {

        return id;

    }

    public String getPrivillage() {

        return privillage;

    }

}

**Add this piece of code in Account.java**

  @ManyToMany

    @JoinTable(

        name = "account\_authority",

        joinColumns = {@JoinColumn(name="account\_id", referencedColumnName = "id")},

        inverseJoinColumns = {@JoinColumn(name="authority\_id", referencedColumnName = "id")})

    private Set<Authority> authorities = new HashSet<>();

**Updated Account.java**

package org.studyeasy.SpringBlog.models;

import java.util.HashSet;

import java.util.List;

import java.util.Set;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.JoinColumn;

import jakarta.persistence.JoinTable;

import jakarta.persistence.ManyToMany;

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;

    private String lastname;

    private String role;

    @OneToMany(mappedBy = "account")

    private List<Post> posts;

    @ManyToMany

    @JoinTable(

        name = "account\_authority",

        joinColumns = {@JoinColumn(name="account\_id", referencedColumnName = "id")},

        inverseJoinColumns = {@JoinColumn(name="authority\_id", referencedColumnName = "id")})

    private Set<Authority> authorities = new HashSet<>();

}