

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №6

Spring Security

тема

Преподаватель

дата, подпись

А.С. Черниговский

инициалы, фамилия

Студент

КИ18-16Б

номер группы, зачётной книжки

дата, подпись

К.И. Оглезнев

инициалы, фамилия

Красноярск 2021

1 Цель работы

Познакомиться с настройкой безопасности в Spring.

2 Задачи работы

Взять практическое задание №5 и добавить следующий функционал:

- 1) Добавить простейшую страницу регистрации. Пользователь вводит свои логин и пароль, и данная информация вносится в базу данных, пользователю присваивается роль пользователя (User) приложения.
- 2) Добавить простейшую форму аутентификации. Форма должна быть создана студентом, а не автоматически сгенерированной Spring.
- 3) В приложении должен быть предусмотрен пользователь — администратор, с, отличной от роли User, ролью администратора (Admin).
- 4) Разграничить уровни доступа к страницам приложения. Пользователь (User) имеет доступ только к страницам просмотра всех записей и запросов. Администратор (Admin) имеет возможность добавлять, редактировать и удалять записи.
- 5) Информация о пользователях и их ролях должна храниться в базе данных. Способ хранения — на усмотрение студента.
- 6) Предусмотреть возможность пользователю выйти из приложения (logout).
- 7) Продемонстрировать умение настраивать безопасность на уровне представлений. Для этого реализуйте приветствие пользователя после его входа и отображение элемента на основе его роли.

3 Ход работы

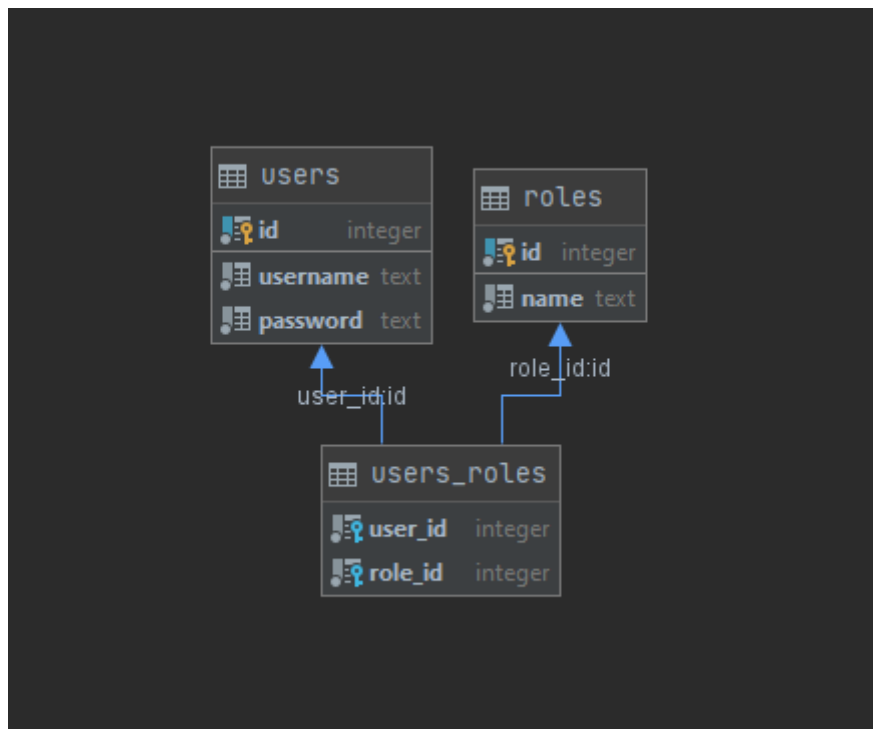


Рисунок 1

```

@Entity
@Table(name = "users")
public class User implements UserDetails {
    @ManyToMany(fetch = FetchType.EAGER)
    @JoinTable(name = "Users_Roles",
        joinColumns = {@JoinColumn(name = "user_id")},
        inverseJoinColumns = {@JoinColumn(name = "role_id")})
    private Collection<Role> roles;

    @Id
    @GeneratedValue
    private int id;

    @Size(min=5, message = "Не меньше 5 знаков")
    @Column(name = "username", nullable = false)
    private String username;

    @Size(min=5, message = "Не меньше 5 знаков")
    @Column(name = "password", nullable = false)
    private String password;

    public User() {
    }
}

```

Рисунок 2

```

public interface UserRepository extends JpaRepository<User, Integer> {
    User findByUsername(String username);
}

```

Рисунок 3

```
@Service("UserService")
public class UserService implements UserDetailsService {
    @PersistenceContext
    private EntityManager em;
    @Autowired
    UserRepository userRepository;
    @Autowired
    RoleRepository roleRepository;

    @Override
    public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
        User user = userRepository.findByUsername(username);

        if (user == null) {
            throw new UsernameNotFoundException("User not found");
        }

        return user;
    }

    public User findUserById(int userId) {
        Optional<User> userFromDb = userRepository.findById(userId);
        return userFromDb.orElse(new User());
    }

    public List<User> allUsers() { return userRepository.findAll(); }
```

Рисунок 4

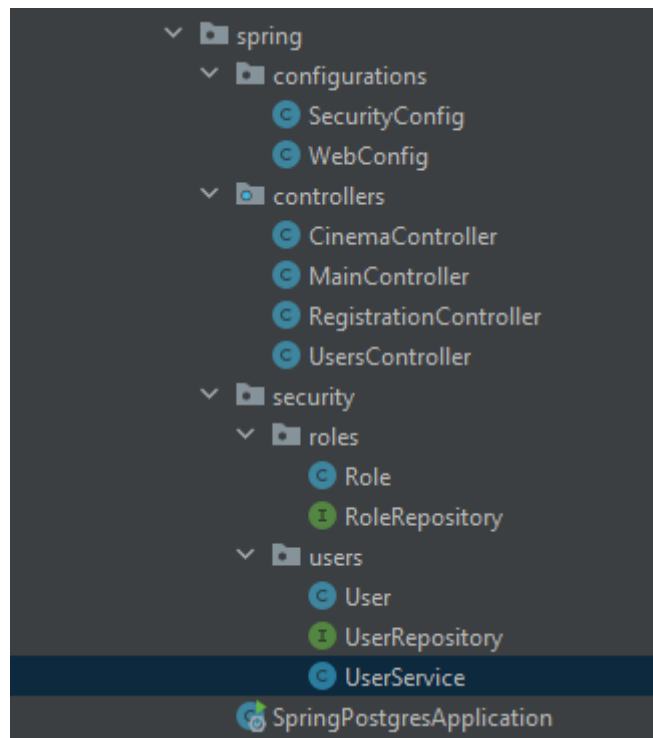


Рисунок 5

```
@Controller
public class RegistrationController {

    @Autowired
    private UserService userService;

    @GetMapping("/registration")
    public ModelAndView registration() {
        ModelAndView mav = new ModelAndView("registration");
        mav.addObject("userForm", new User());

        return mav;
    }

    @PostMapping("/registration")
    public ModelAndView addUser(@ModelAttribute("userForm") @Valid User userForm, BindingResult bindingResult) {
        ModelAndView mav = new ModelAndView("registration");
        if (bindingResult.hasErrors()) {
            return mav;
        }
        if (!userService.saveUser(userForm)) {
            mav.addObject("usernameError", "Пользователь с таким именем уже существует");
            return mav;
        }
        mav.setViewName("redirect:/");
        return mav;
    }
}
```

Рисунок 6

```

@Controller
public class UsersController {
    @Autowired
    private UserService userService;

    @GetMapping("/users")
    public ModelAndView userList() {
        ModelAndView mav = new ModelAndView( viewName: "users");
        mav.addObject( attributeName: "allUsers", userService.allUsers());
        return mav;
    }

    @PostMapping("/users")
    public String deleteUser(@RequestParam int userId,
                             @RequestParam String action) {
        if (action.equals("delete")) {
            userService.deleteUser(userId);
        }
        return "redirect:/users";
    }
}

```

Рисунок 7

```

spring.datasource.url=jdbc:postgresql://localhost:5432/Users
spring.datasource.username=postgres
spring.datasource.password=123
spring.jpa.generate-ddl=false
spring.jpa.hibernate.ddl-auto=update

spring.mvc.view.prefix = /pages/
spring.mvc.view.suffix = .jsp

spring.jpa.show-sql=true
spring.jpa.properties.hibernate.temp.use_jdbc_metadata_defaults=false

```

Рисунок 8

Войти

Зарегистрироваться

Рисунок 9

Вход в систему

Имя

Пароль

Войти

[Зарегистрироваться](#)

Рисунок 10

Привет, 1

Пользователи (только админ)

Выйти

Рисунок 11

ID	UserName	Password	Roles	
1	1	1	ROLE_ADMIN;	<button>Удалить</button>
2	12	12	ROLE_USER;	<button>Удалить</button>
3	13	13	ROLE_USER;	<button>Удалить</button>

Рисунок 12

Привет, 12

Выйти

Кинотеатры

ID	Название	Адрес	Телефон	
TEST	TEST	TEST	TEST	
A19	НАУТИЛУС	Абакан, Тараса Шевченко, 74	+7	
B24	ЛУЧ	Лесосибирск, 6-й микрорайон, 1	+7 XXX XXX-XX-XX	
A24	ЛУЧ	Красноярск, Карла Маркса 149	+7 XXX XXX-XX-XX	
C24	ЛУЧ	Красноярск, Что-то, 1	+7 XXX XXX-XX-XX	
QWE	C24	C24	C24	

Рисунок 13

Привет, 1

Пользователи (только админ)

Выйти

Кинотеатры

ID	Название	Адресс	Телефон		
TEST	TEST	TEST	TEST	Изменить	Удалить
A19	НАУТИЛУС	Абакан, Тараса Шевченко, 74	+7	Изменить	Удалить
B24	ЛУЧ	Лесосибирск, 6-й микрорайон, 1	+7 XXX XXX-XX-XX	Изменить	Удалить
A24	ЛУЧ	Красноярск, Карла Маркса 149	+7 XXX XXX-XX-XX	Изменить	Удалить
C24	ЛУЧ	Красноярск, Что-то, 1	+7 XXX XXX-XX-XX	Изменить	Удалить
QWE	C24	C24	C24	Изменить	Удалить
				Добавить новый	

Рисунок 14

4 Вывод

В ходе данной лабораторной работы были изучены основы работы с настройкой безопасности в Spring.