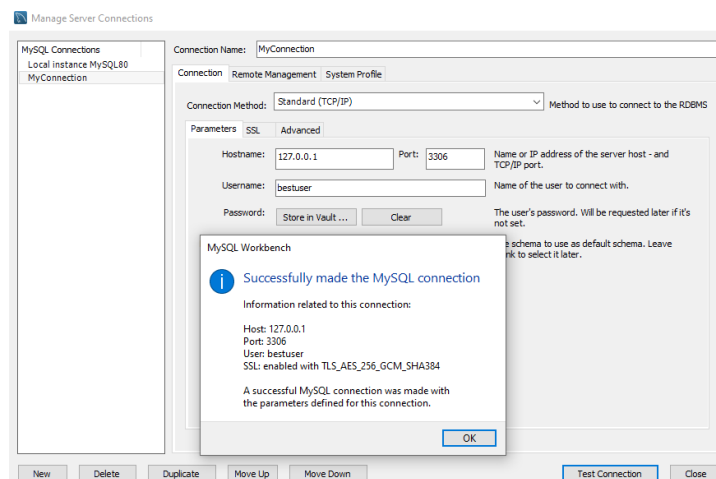


25. Spring Framework (Security)

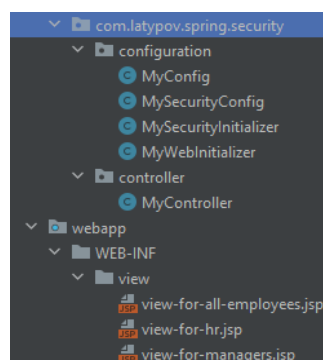
- Защита приложения встроенными средствами Spring
- Авторизация и аутентификация
- Разграничение доступа
- Шифрование паролей

Сценарий: сохранить данные входа пользователей в систему в БД в зашифрованном виде, разграничить доступ к страницам в зависимости от роли пользователя, реализовать процедуры авторизации и аутентификации встроенным функционалом Spring, на основе Tomcat-сервера.

Конфигурация подключения в среде MySQL:



Структура проекта:



Создание таблиц пользователей и ролей, добавление данных:

```
USE my_db;

CREATE TABLE users (
  username varchar(15),
  password varchar(100),
  enabled tinyint(1),
  PRIMARY KEY (username)
) ;

CREATE TABLE authorities (
  username varchar(15),
  authority varchar(25),
  FOREIGN KEY (username) references users(username)
) ;

INSERT INTO my_db.users (username, password, enabled)
VALUES
  ('linar', '{noop}linar', 1),
  ('elena', '{noop}elena', 1),
  ('ivan', '{noop}ivan', 1);

INSERT INTO my_db.authorities (username, authority)
VALUES
  ('linar', 'ROLE_EMPLOYEE'),
  ('elena', 'ROLE_HR'),
  ('ivan', 'ROLE_HR'),
  ('ivan', 'ROLE_MANAGER');
```

Pom.xml (добавление зависимостей):

```
<dependencies>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>5.2.12.RELEASE</version>
  </dependency>

  <dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>4.0.1</version>
    <scope>provided</scope>
  </dependency>

  <dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-config</artifactId>
    <version>5.4.1</version>
  </dependency>

  <dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-web</artifactId>
    <version>5.4.1</version>
  </dependency>

  <dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-taglibs</artifactId>
    <version>5.4.2</version>
  </dependency>

  <dependency>
```

```

<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
<version>8.0.22</version>
</dependency>

<dependency>
<groupId>com.mchange</groupId>
<artifactId>c3p0</artifactId>
<version>0.9.5.2</version>
</dependency>
</dependencies>

```

configuration.MyConfig (класс конфигурации, создание бинов DataSource и ViewResolver):

```

@Configuration
@ComponentScan(basePackages = "com.latypov.spring.security")
@EnableWebMvc
public class MyConfig {

    @Bean
    public ViewResolver viewResolver() {
        InternalResourceViewResolver internalResourceViewResolver = new
InternalResourceViewResolver();
        internalResourceViewResolver.setPrefix("/WEB-INF/view/");
        internalResourceViewResolver.setSuffix(".jsp");
        return internalResourceViewResolver;
    }

    @Bean
    public DataSource dataSource() {
        ComboPooledDataSource dataSource = new ComboPooledDataSource();
        try {
            dataSource.setDriverClass("com.mysql.cj.jdbc.Driver");

dataSource.setJdbcUrl("jdbc:mysql://localhost:3306/my_db?useSSL=false&serverTime
zone=UTC");
            dataSource.setUser("bestuser");
            dataSource.setPassword("bestuser");
        } catch (PropertyVetoException e) {
            e.printStackTrace();
        }
        return dataSource;
    }
}

```

configuration.MyWebInitializer: (определение конфиг класса и маппингов для сервлета):

```

public class MyWebInitializer extends
AbstractAnnotationConfigDispatcherServletInitializer {

    @Override
    protected Class<?>[] getRootConfigClasses() {
        return null;
    }

    @Override
    protected Class<?>[] getServletConfigClasses() {
        return new Class[] {MyConfig.class};
    }
}

```

```

@Override
protected String[] getServletMappings() {
    return new String[]{"/*"};
}
}

```

configuration.MySecurityInitializer (инициализация Security блока):

```

public class MySecurityInitializer extends
AbstractSecurityWebApplicationInitializer {
}

```

configuration.MySecurityConfig (добавление аутентификации и разграничение доступа, аннотация @EnableWebSecurity):

```

@EnableWebSecurity
public class MySecurityConfig extends WebSecurityConfigurerAdapter {
    @Autowired
    DataSource dataSource;

    @Override
    protected void configure(AuthenticationManagerBuilder auth) throws Exception
    {
        auth.jdbcAuthentication().dataSource(dataSource);
    }

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.authorizeRequests()
            .antMatchers("/").hasAnyRole("EMPLOYEE", "HR", "MANAGER")
            .antMatchers("/hr_info").hasRole("HR")
            .antMatchers("/manager_info").hasRole("MANAGER")
            .and().formLogin().permitAll();
    }
}

```

Устаревший вариант с заданием паролей вручную:

```

UserBuilder userBuilder = User.withDefaultPasswordEncoder();
auth.inMemoryAuthentication()

.withUser(userBuilder.username("linar").password("linar").roles("EMPLOYEE"))
    .withUser(userBuilder.username("elena").password("elena").roles("HR"))

.withUser(userBuilder.username("ivan").password("ivan").roles("MANAGER", "HR"));

```

controller.MyController (определение маппингов):

```

@Controller
public class MyController {
    @GetMapping("/")
    public String getInfoForAllEmps() {
        return "view-for-all-employees";
    }

    @GetMapping("/hr_info")
    public String getInfoOnlyForHR() {
        return "view-for-hr";
    }
}

```

```

    }

    @GetMapping("/manager_info")
    public String getInfoOnlyForManagers() {
        return "view-for-managers";
    }
}

```

views.view-for-all-employees.html (страница для всех):

```

<html>
<head>
    <title></title>
</head>
<body>
<h3>
    Information for all Employees
</h3>
<br><br>
<security:authorize access="hasRole('HR')">
<input type="button" value="Salary" onclick="window.location.href = 'hr_info'">
Only for HR staff
</security:authorize>
<br><br>

<security:authorize access="hasRole('MANAGER')">
<input type="button" value="Performance" onclick="window.location.href =
'manager_info'">
Only for Managers
</security:authorize>

</body>
</html>

```

views.view-for-hr.html (страница для HR-сотрудников):

```

<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
    <title>Title</title>
</head>
<body>
<h3>Here you can see all salaries</h3>

</body>
</html>

```

views.view-for-managers.html (страница для менеджеров):

```

<html>
<head>
    <title>Title</title>
</head>
<body>
<h3>Here you can see performance of employees</h3>

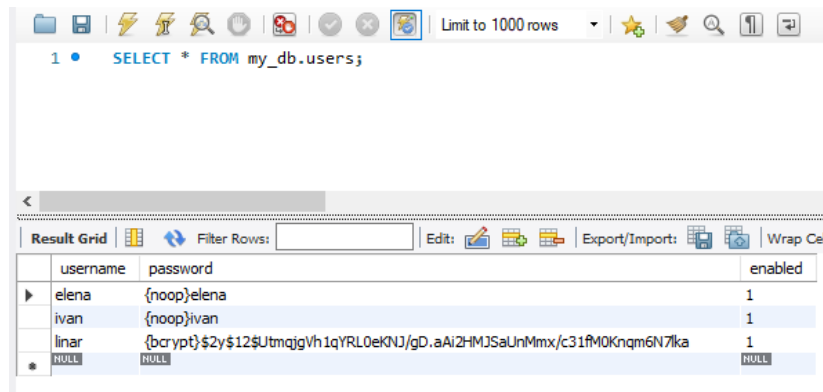
</body>
</html>

```

Задание нового пароля в зашифрованном виде для одного из пользователей (bcrypt-шифрование):

```
update my_db.users set password =  
'{bcrypt}$2y$12$.Qx6KpHWWIeEtiqjNZa5TOSVCzIZ49pPdbXFvW48FxnPXnfj3qmFO'  
where username = 'linar';
```

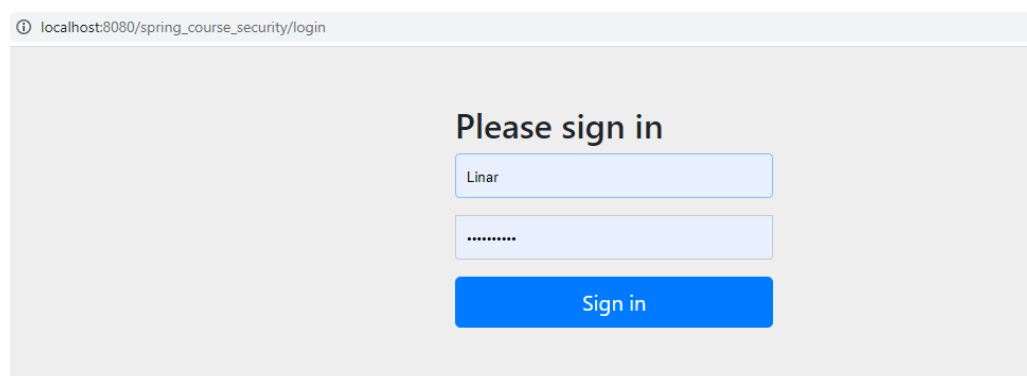
Результат:



username	password	enabled
elena	{noop}elena	1
ivan	{noop}ivan	1
linar	{bcrypt}\$2y\$12\$.Qx6KpHWWIeEtiqjNZa5TOSVCzIZ49pPdbXFvW48FxnPXnfj3qmFO	1
NULL	NULL	NULL

Запуск приложения

Окно входа в приложение:



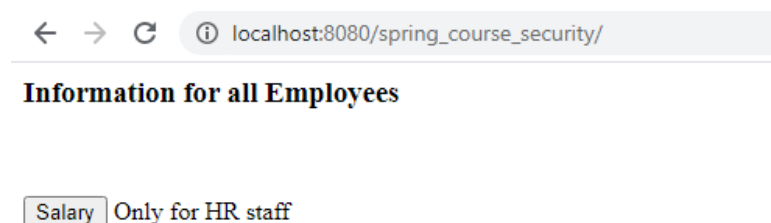
Please sign in

Linar

.....

Sign in

Авторизация под HR-пользователем (ресурсы видимые только для HR):



← → ↻ ⓘ localhost:8080/spring_course_security/

Information for all Employees

Salary Only for HR staff

Авторизация под менеджером (ресурсы видимые только для менеджеров):

Information for all Employees

Salary Only for HR staff

Performance Only for Managers

Авторизация под обычным пользователем (пустая страница):

Information for all Employees