

BOGO 1:

Spring Boot Security: Art Collectors Registration and Login Application

Ms. Lewis

July 4, 2024

PART I.

TODOS:

Create a spring boot security application that registers an art-collector. The application implements role based security and validates each registration field with Java bean validation.

Main Functionalities:

Register an art-collector and store data in a MySQL database.

Login Authentication - validate art-collector login credentials with database email and password.

Authorize request with Spring Security's filter chain.

After successful authentication and authorization display a list of art-collectors.

Implementation:

Create Spring Boot Security application.

Project Name: Art Collectors Registration

Add Dependencies

SPRING SECURITY

SPRING Web

SPRING DEV TOOLS

MYSQL JDBC→ MYSQL Driver

SPRING DATA JPA

LOMBOOK
THYMELEAF
SPRING VALIDATION → I/O

Create Four Simple Thymeleaf pages.

index.html - main landing page

Create Thymeleaf Template for Home Page index.html - the main landing page with title of page and links to login and register endpoints.

art-collector-registration.html - art-collector registration form. To register

enter: username
email
password
country

login.html - art-collector uses credentials to login in

username
password

registered-art-collectors.html - a list of registered art-collector

username
country

Configure the application to connect, save and query the MySQL
database Create a database: artcollectorsdb

Create JPA Entities classes - ArtCollector and Role

Set the relationship as a ManyToMany.

Add Entity classes to "entity" package

Each Entity class will have a corresponding table in the artcollectorsdb database For the ArtCollector entity name its table: "artcollectors"
For the Roles entity name its table: "roles"

ArtCollector Class

Long id

String username

String email

String password

List of Roles

getters/setters

constructors

toString

Role Class

Long id

String name

List of Art-Collectors

getters/setters

constructors

toString

Create ArtCollectorDto class- dto" package

Validate each field using Spring validation, as "NotEmpty" and provide a message for email and password

Create method: isAdminRegistration() and implement check if email domain indicates an admin role.

getters/setters

Create ArtCollectorRepository and RoleRepository interfaces - in the "repository" package, include @Repository

both extend JpaRepository

Add one method declaration to the ArtCollectorRepository interface that finds the collector by email

Add one method declaration to the RoleRepository interface that finds the role by name.

Create ArtCollectorService interface in the service package, include

@Component Add method declarations

```
void saveArtCollector(ArtCollector artCollector);  
ArtCollector findArtCollectorByEmail(String email);  
List<ArtCollector> findAllArtCollectors();
```

Create ArtCollectorServiceImpl class annotate as **@Service**

Inject both repository classes and a BCryptPasswordEncoder or PasswordEncoder object using constructor Based injection

Implement ArtCollectorService and override its methods and provide implementation

When saving user, invoke the isAdminRegistration() method to assign a role to new user. Create method that converts a User object to a UserDto object, use example in demo

Create AuthorizeController class use @Controller to manage request.

Create a method to handle home endpoint requests and returns the index page. @GetMapping

Endpoint: index

Returns index.html

Create a handler method to handle the art-collectors registration form request @GetMapping

Endpoint: register

Returns art-collector-registration.html

Create a handler method to handle art-collectors registration form submit request, (saves art-collector to database)

@PostMapping

EndPoint: register/save

Returns "redirect:/register?success";

Create a handler method to handle a list of art collectors.

@GetMapping

Endpoint: artcollectors

Returns registered-art-collectots.html

Create a handler method to handle logging in.

@GetMapping

Endpoint: login

Returns login.html

Configure Spring Security

Create a SpringSecurity class and annotate it with @Configuration, @EnableWebSecurity in the package "config"

Add the passwordEncoder method to return a BCryptPasswordEncoder and inject it using @Bean

Add the filterChain method that authorizes requests and returns a SecurityFilterChain object.
Inject filterChain method as bean using @Bean
<p>Request authorization filter</p> <p>authorizeHttpRequests with authorize permitAll to the “register” and all pages that begin with register endpoint permitAll to the “index” endpoint Access to the artcollectors endpoint , If request has role as “USER” or ADMIN</p> <p>Custom form login filter</p> <p>Add customer formLogin loginPage access to “login” endpoint loginProcessingUrl to “login” endpoint defaultSuccessUrl to “artcollecotrs” endpoint Finally permitAll</p> <p>Logout filter</p> <p>logoutRequestMatcher configure new AntPathRequestMatcher to “logout permitAll</p> <p>build SecurityFilterChain object and return</p>
<p>Create CustomUserDetailsService class and implement UserDetailsService</p> <p>Inject the ArtCollectorRepository using constructor based injection Override the method loadUserByUsername and provide implementation to Get art collector by email Get art collector password Get role and apply SimpleGrantedAuthority to role - Map role to authority (USER, ADMIN) Return UserDetails</p>

PART II.

TODOS:

Run program and register a user using the gmail domain (example: user1@gmail.com). Register another user as an admin using the admin domain (example: admin1@admin.com)

Each team member presents one section.

Section 1

Register an art-collector and store data in a MySQL database

- Explain database configuration and entity implementation
- Explain the process to register and validate an art collector.
- Explain and walk through simple Thymeleaf page
- Explain and walk through method with endpoint that handles art collectors' registration request
- Explain and walk through method that saves an art collector to database
- Explain and walk through how password is encrypted

Section 2

Login Authentication - validate art-collector login credentials with database email and password.

- Explain and walk through simple Thymeleaf page
- Explain the process to register and validate an art collector.
- Explain and walk through method(s) that handles validating an art collectors login and returns a valid UserDetails object
- Containing:
 - email
 - password
 - role
 - or an invalid credentials message.

Section 3

Authorize request with Spring Security

- Explain and walk through method(s) that handles authorization
- Explain and walk through each section of method(s)

Section 4

After successful authentication and authorization display a list of art-collectors

- Explain and walk through simple Thymeleaf page

- Explain and walk through method(s) that handle returning a list of art-collectors after successful login.