Java MVC Frameworks

Filters and User Authentication

SoftUni Team

Technical Trainers

Software University

SoftUni Foundation

Software University

http://softuni.bg

Table of Contents

Have a Question?

SoftUni Foundation

sli.do

#java-web

Filters

Filters

SoftUni Foundation

- A filter is an object used to intercept the HTTP requests and responses of your application
- We can perform two operations at two instances:
  - Before sending the request to the controller
  - Before sending a response to the client

Filter Example(1)

SoftUni Foundation

GreetingFilter.java

```
@Component
public class GreetingFilter implements Filter {

    @Override
    public void doFilter(ServletRequest servletRequest, ServletResponse servletResponse,
FilterChain filterChain) throws IOException, ServletException {
        HttpServletRequest request = (HttpServletRequest) servletRequest;
        HttpServletResponse response = (HttpServletResponse) servletResponse;

        request.getSession().setAttribute("name", "Pesho");

        filterChain.doFilter(request, response);
    }
}
```

Filter Example(2)

HomeController.java

```
@Controller
public class HomeController {

    @GetMapping("/")
    public ModelAndView index(ModelAndView modelAndView, HttpSession session) {
        modelAndView.setViewName("index");
        modelAndView.addObject("name", session.getAttribute("name"));

        return modelAndView;
    }
}
```

Filter Example(3)

SoftUni Foundation

index.html

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>Filter Demo</title>
</head>
<body>
    <h1 th:text="'Hello, ' + ${name} + '!'"></h1>
</body>
</html>
```

Spring Security

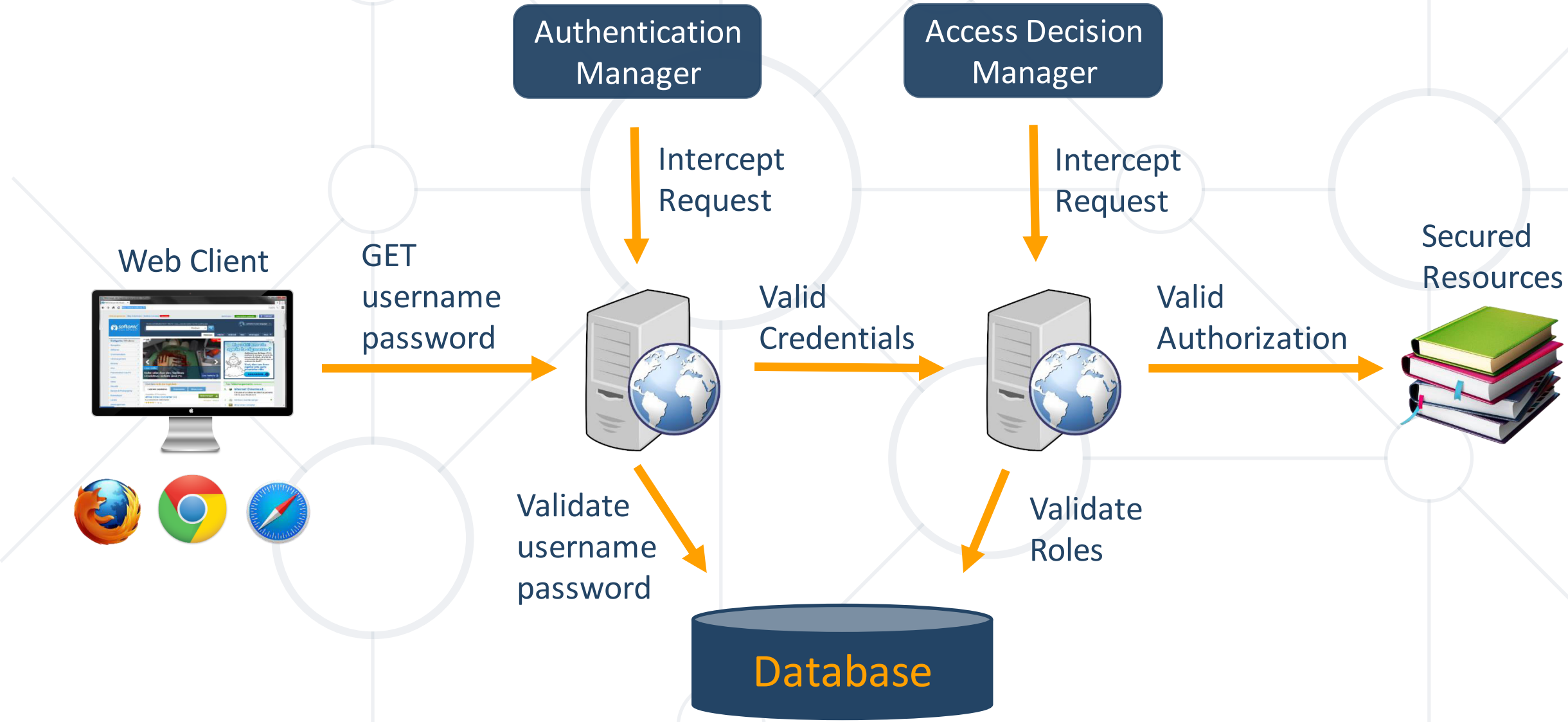
Spring Security

- Framework that focuses on providing both authentication and authorization

Authentication

Authorization

Spring Security Mechanism



Spring Security Maven

SoftUni Foundation

pom.xml

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

Spring Security Configuration (1)

- Extending the WebSecurityConfigurerAdapter class.

SecurityConfiguration.java

```
@Configuration
@EnableWebSecurity
public class SecurityConfiguration extends WebSecurityConfigurerAdapter {
//Configuration goes here
}
```

Enable Security

Spring Security Configuration (2)

SoftUni Foundation

- Overriding configure() method.

SecurityConfiguration.java

```
@Override
protected void configure(HttpSecurity http) throws Exception {
http
    .authorizeRequests()
    .antMatchers("/", "/register").permitAll()
    .anyRequest().authenticated()
}
```

Authorize Requests

Permit Routes

Require Authentication

Registration - User

- Implementing the UserDetails interface.

User.java

```
@Entity
public class User implements UserDetails {
    private String username;
    private String password;
    private boolean isAccountNonExpired;
    private boolean isAccountNonLocked;
    private boolean isCredentialsNonExpired;
    private boolean isEnabled;
    private Set<Role> authorities;
}
```

Registration - Roles

- Implementing the GrantedAuthority interface.

Role.java

```
public class Role implements GrantedAuthority {
    private String authority;
}
```

Role Interface

Registration - UserService

- Implementing the UserDetailsService interface.

UserServiceImpl.java

```
@Service
public class UserServiceImpl implements UserDetailsService {
    @Autowired
    private BCryptPasswordEncoder bCryptPasswordEncoder;

    @Override
    public void register(RegisterModel registerModel) {
        bCryptPasswordEncoder.encode(password));
    }
}
```

Encrypt Password

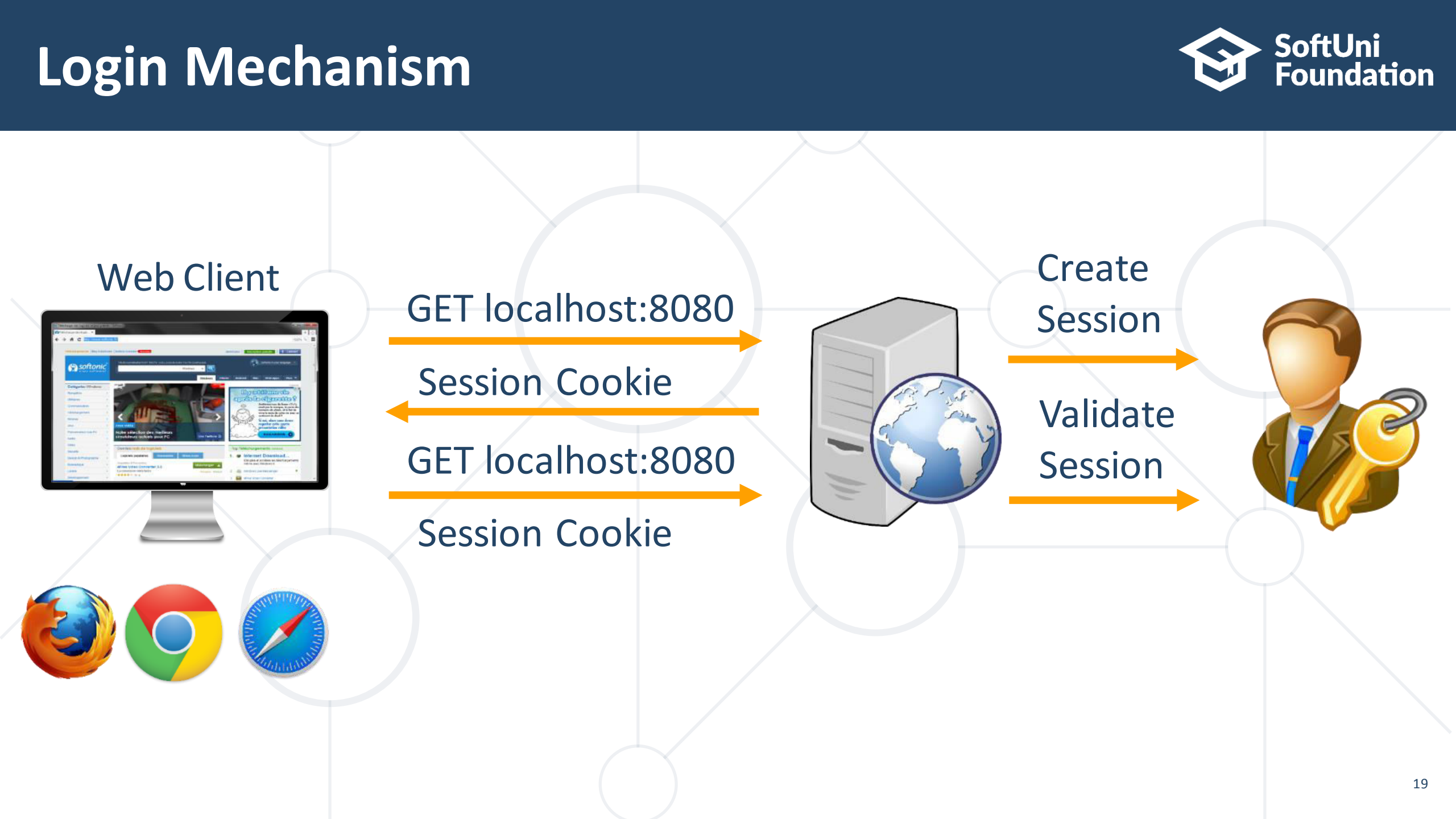
Registration - Configuration

- Disabling CSRF protection temporarily.

SecurityConfiguration.java

```
@Override
protected void configure(HttpSecurity http) throws Exception {
    http
        .and()
        .csrf().disable();
}
```

Login Mechanism



Login - Configuration

SoftUni Foundation

SecurityConfiguration.java

```
.and()
  .formLogin().loginPage("/login").permitAll()
  .usernameParameter("username")
  .passwordParameter("password")
```

login.html

```
<input type="text" name="username"/>
<input type="text" name="password"/>
```

Login - UserService

UserServiceImpl.java

```
@Service
public class UserServiceImpl implements UserDetailsService {
    @Autowired
    private BCryptPasswordEncoder bCryptPasswordEncoder;

    @Override
    public UserDetails loadUserByUsername(String username) throws
UsernameNotFoundException  {
    }
}
```

User Service Interface

Login - Controller

LoginController.java

```
@Controller
public class LoginController {
    @GetMapping("/login")
    public String getLoginPage(@RequestParam(required = false) String error, Model model) {
        if(error != null){
            model.addAttribute("error", "Error");
        }

        return "login";
    }
}
```

Error Handling

Logout

SoftUni Foundation

SecurityConfiguration.java

```
.and()
.logout().logoutSuccessUrl("/login?logout").permitAll()
```

Logout. No Controller is required

Principal

- This is the currently logged user

UserController.java

```
@GetMapping("/user")
public String getUser(Principal principal){
    System.out.println(principal.getName());
    return "user";
}
```

Print Logged-In username

Pre / Post Authorize

- Grant Access to specific methods

SecurityConfiguration.java

```
@EnableGlobalMethodSecurity(prePostEnabled = true)
public class SecurityConfiguration extends WebSecurityConfigurerAdapter {
}
```

Enables **PreAuthorize**

UserService.java

```
public interface UserService extends UserDetailsService {
    @PreAuthorize("hasRole('ADMIN')")
    void delete();
}
```

Requires Admin Role to execute

No Access Handling

SoftUni Foundation

SecurityConfiguration.java

```
.and()
.exceptionHandling().accessDeniedPage("/unauthorized");
```

AccessController.java

```
@GetMapping("/unauthorized")
public String unauthorized(){
    return "unauthorized";
}
```

CSRF

Cross-Site Request Forgery

Spring CSFR Protection

SoftUni Foundation

AccessController.java

```
.csrf()
   .csrfTokenRepository(csrfTokenRepository())

private CsrfTokenRepository csrfTokenRepository() {
  HttpSessionCsrfTokenRepository repository =
        new HttpSessionCsrfTokenRepository();
  repository.setSessionAttributeName("_csrf");
  return repository;
}
```

form.html

```
<input type="hidden" th:name="${_csrf.parameterName}" th:value="${_csrf.token}" />
```

Summary

- Spring Security – framework that focuses on providing both authentication and authorization
- Filters & Interceptors
  - Intercepting Requests
  - Modifying Request Chain Behaviour

Questions?

SoftUni

Software University

SoftUni Svetlina

SoftUni Creative

SoftUni Digital

SoftUni Foundation

SoftUni Kids

SoftUni Diamond Partners

SoftUni Foundation

XSsoftware

SoftUni Organizational Partners

SoftUni Foundation

Trainings @ Software University (SoftUni)

SoftUni Foundation

- Software University – High-Quality Education and Employment Opportunities
  - softuni.bg
- Software University Foundation
  - http://softuni.foundation/
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg

License

- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license