

Java MVC Frameworks

Spring Security. Authentication.
Authorization.



SoftUni Team
Technical Trainers
Software University
<http://softuni.bg>

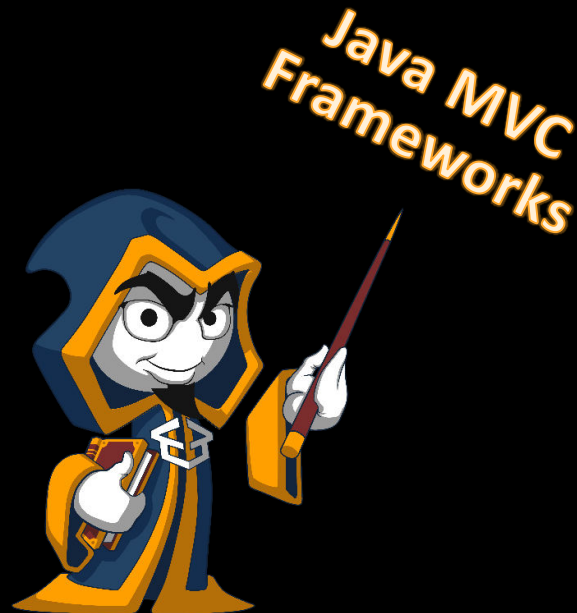


Table of Contents



sli.do

#java-web



What is Spring Security

Spring Security

- Framework that focuses on providing both **authentication** and **authorization**

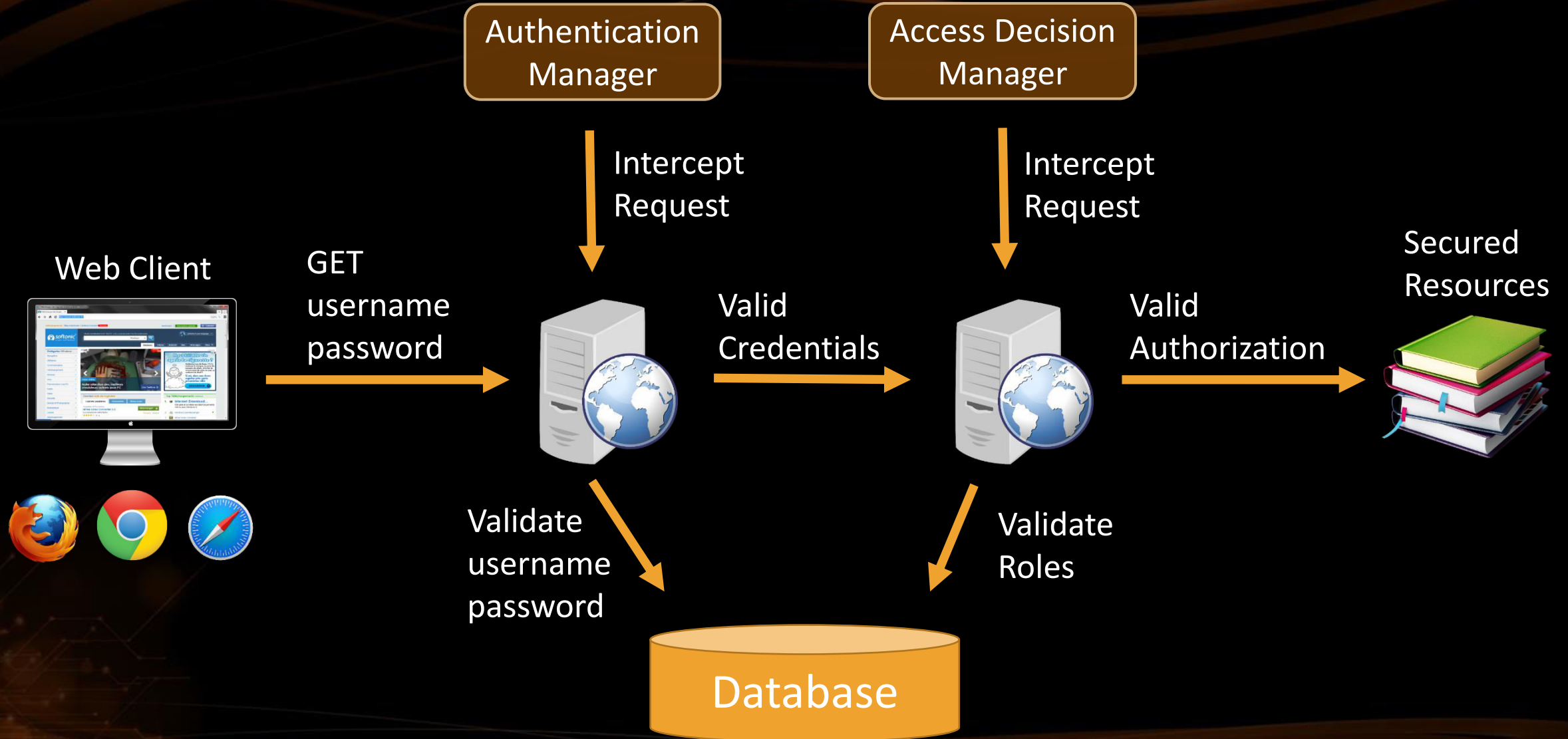


Authentication



Authorization

Spring Security Mechanism



Spring Security Maven

`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)

- Overriding `configure()` method.

SecurityConfiguration.java

@Override

```
protected void configure(HttpSecurity http) throws Exception {  
    http
```

```
        .authorizeRequests()
```

Authorize Requests

```
        .antMatchers("/", "/register").permitAll()
```

Permit Routes

```
        .anyRequest().authenticated()
```

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;
```

Encrypt Password

@Override

```
public void register(RegisterModel registerModel) {
```

```
    bCryptPasswordEncoder.encode(password);
```

```
}
```

```
}
```

Registration - Configuration

- Disabling **CSRF** protection temporarily.

SecurityConfiguration.java

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

Disable CSRF

Login Mechanism

Web Client



GET localhost:8080
Session Cookie
GET localhost:8080
Session Cookie



Create
Session

Validate
Session



Login - Configuration

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;
```

User Service
Interface

@Override

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

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

SecurityConfiguration.java

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

Logout. No
Controller is
required

Remember Me

SecurityConfiguration.java

```
.and()  
    .rememberMe()  
    .rememberMeParameter("remember")  
    .key("remember Me Encryption Key")  
    .rememberMeCookieName("rememberMeCookieName")  
    .tokenValiditySeconds(10000)
```

login.html

```
<input name="remember" type="checkbox" />
```

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

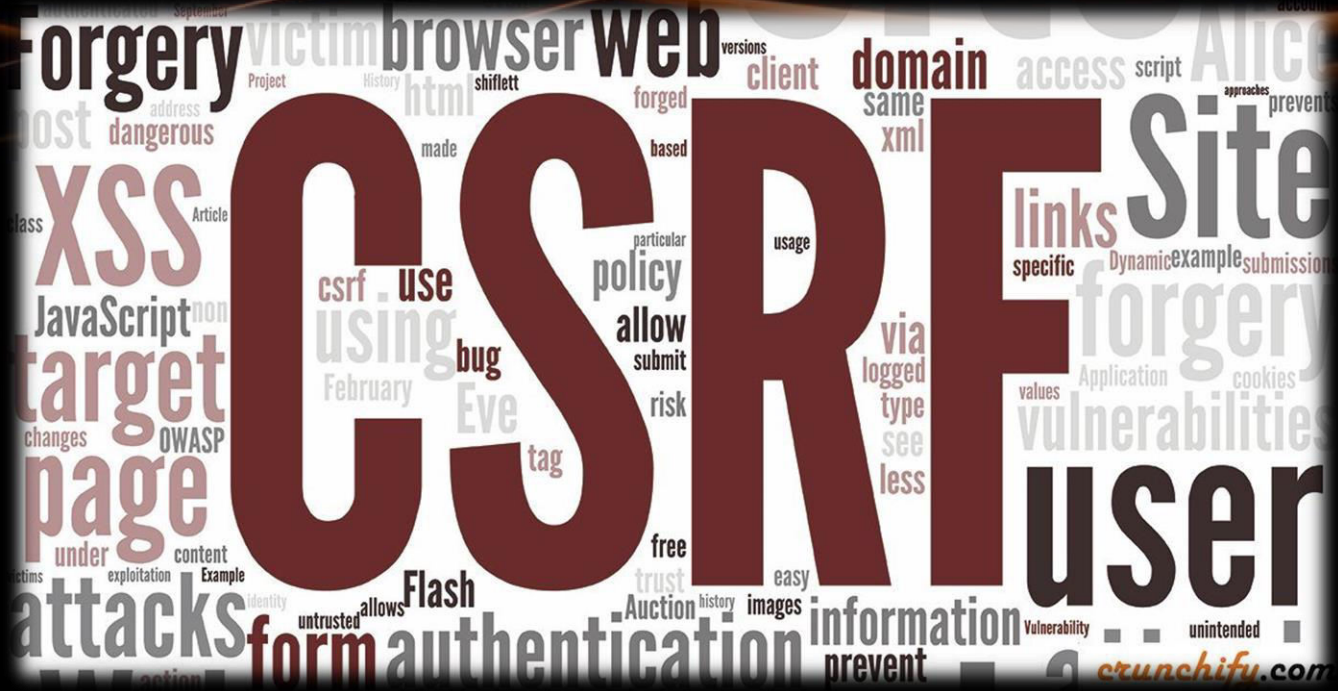
No Access Handling

SecurityConfiguration.java

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

AccessController.java

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



CSRF

Cross-Site Request Forgery

Spring CSFR Protection

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}" />
```



What is Thymeleaf Security

- Functionality to **display** data based on **authentication** rules

pom.xml

```
<dependency>  
    <groupId>org.thymeleaf.extras</groupId>  
    <artifactId>thymeleaf-extras-springsecurity4</artifactId>  
</dependency>
```

pom.xml

```
<!DOCTYPE html>
<html lang="en"
      xmlns:th="http://www.thymeleaf.org"
      xmlns:sec="http://www.thymeleaf.org/extras/spring-security">
<body>
<div sec:authentication="name">
    The value of the "name" property of the authentication object
    should appear here.
</div>
</body>
</html>
```

Show the
username

pom.xml

```
<!DOCTYPE html>
<html lang="en"
      xmlns:th="http://www.thymeleaf.org"
      xmlns:sec="http://www.thymeleaf.org/extras/spring-security">
<body>
<div sec:authorize="hasRole('ADMIN')">
    This content is only shown to administrators.
</div>
</body>
</html>
```

Show if you are
admin

Summary

- **Spring Security** – framework that focuses on providing both **authentication** and **authorization**
- **Thymeleaf Security** – functionality to **display** data based on **authentication** rules



Java MVC Frameworks – Spring Security



Questions?



SoftUni Diamond Partners



INDEAVR
Serving the high achievers

 **INFRAGISTICS®**

 **SoftwareGroup**
doing it right


XSsoftware

NETPEAK
SEO and PPC for Business

**SUPER
HOSTING
®.BG**

SoftUni Diamond Partners



LIEBHERR



License

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



Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
 - softuni.bg
- Software University Foundation
 - <http://softuni.foundation/>
- Software University @ Facebook
 - facebook.com/SoftwareUniversity
- Software University Forums
 - forum.softuni.bg



**Software
University**



**SoftUni
Foundation**

