Spring Boot

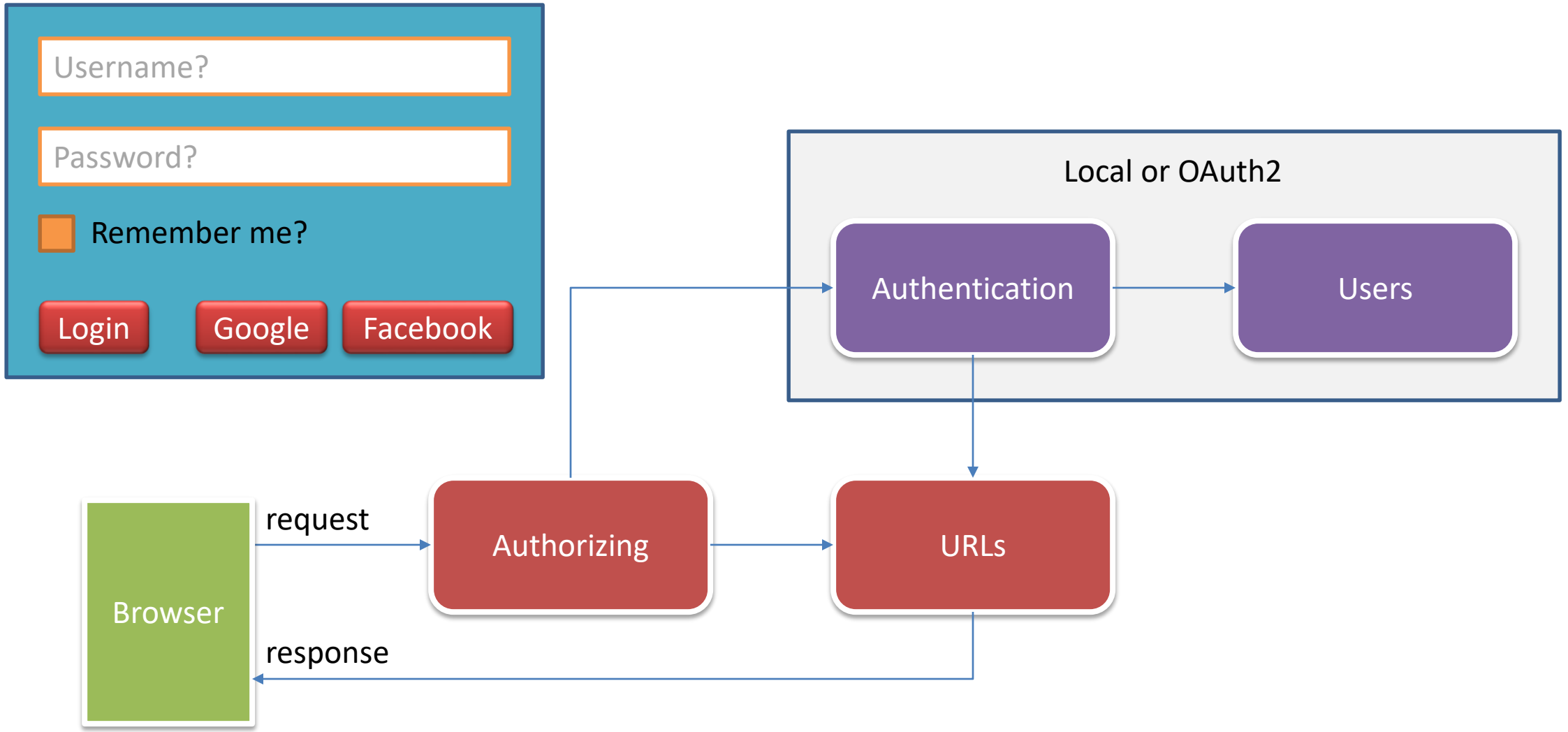
http://www.lycato.com

Basic Spring Security

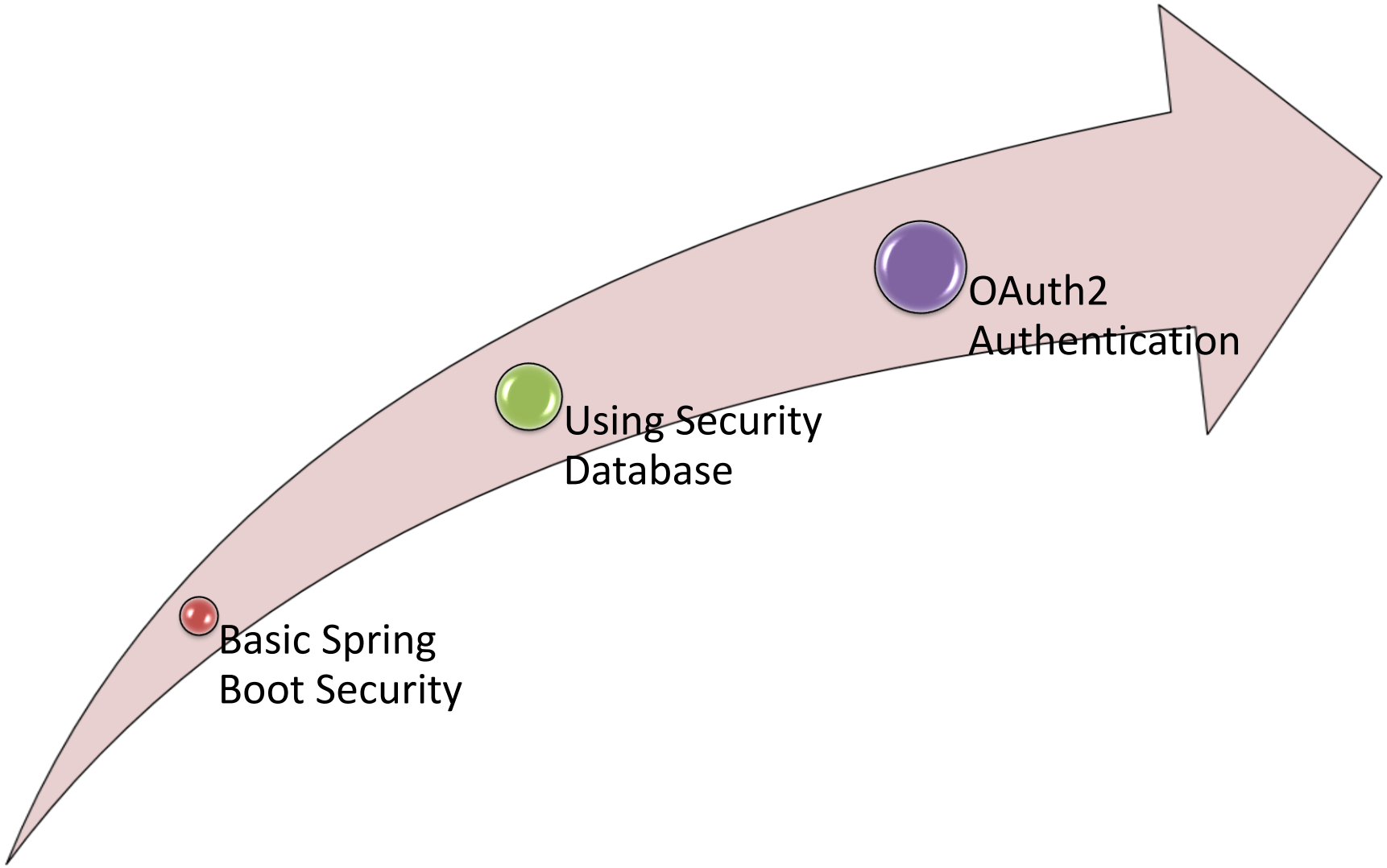
Nguyễn Nghiệm

Microsoft SQL Server Visual Studio Code eclipse

Authentication & Authorization Introduction



Spring Boot Security Roadmap



Agenda

- Security Configuration
  - AuthenticationManagerBuilder
  - HttpSecurity
  - WebSecurity
  - PasswordEncoder
- Login UI
  - HttpBasic
  - HTML Login Form
- Custom Security
  - @PreAuthorize
  - Programmatically
  - Thymeleaf/ Thymeleaf Extras

WebSecurity Configuration

Spring Security Dependence

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

<dependency>
    <groupId>org.thymeleaf.extras</groupId>
    <artifactId>thymeleaf-extras-springsecurity5</artifactId>
</dependency>
```

```java
@Configuration @EnableWebSecurity
public class SecurityConfig extends WebSecurityConfigurerAdapter {
    @Autowired
    PasswordEncoder pe;
    @Override // Quản lý user
    protected void configure(AuthenticationManagerBuilder auth) throws Exception {...}
    @Override // Cấu hình cấp quyền truy cập
    protected void configure(HttpSecurity http) throws Exception {...}
    @Override // Cấu hình bỏ qua các tài nguyên tĩnh
    public void configure(WebSecurity web) throws Exception {...}
    @Bean // Hỗ trợ cơ chế mã hóa và so sánh mật khẩu
    public BCryptPasswordEncoder getPasswordEncoder() {
        return new BCryptPasswordEncoder();
    }
}
```

WebSecurityConfigurer

Security Configuration

```
@Override // username=user, password=123, role=ADMIN
protected void configure(AuthenticationManagerBuilder auth) throws Exception {
    auth.inMemoryAuthentication()
        .withUser("user").password(pe.encode("123")).roles("ADMIN");
}
@Override
protected void configure(HttpSecurity http) throws Exception {
    http.csrf().disable().cors().disable();
    http.authorizeRequests()
        .anyRequest().authenticated(); // Tất cả mọi request đều phải đăng nhập
    http.httpBasic(); // Sử dụng hộp thoại đăng nhập
}
@Override // Không kiểm soát quyền truy cập các url bắt đầu bởi /static/ và /resources/
public void configure(WebSecurity web) throws Exception {
    web.ignoring().antMatchers("/static/**", "/resources/**");
}
```

Basic Authorizing

1. AuthenticationManagerBuilder

```
auth.inMemoryAuthentication() // quản lý nguồn users
    .withUser("user1").password(pe.encode("123")).roles("USER")
    .withUser("user2").password(pe.encode("123")).roles("ADMIN");
```

2. HttpSecurity

```
http.authorizeRequests() // phân quyền sử dụng
    .antMatchers("/admin/**").hasAnyRole("ADMIN", "USER")
    .antMatchers("/admin/report").hasRole("ADMIN")
    .antMatchers("/order/**",
        "/account/edit-profile", "/account/change-password").authenticated()
    .anyRequest().permitAll();
http.exceptionHandling()
    .accessDeniedPage("/security/access/denied"); // Xử lý truy cập trái phép
```

configure(HttpSecurity http)

```
http.exceptionHandling()
        .accessDeniedPage("/security/access/denied");
http.formLogin() // <= http.httpBasic()
        .loginProcessingUrl("/spring/login")
        .loginPage("/security/login/form")
        .defaultSuccessUrl("/security/login/success")
        .failureUrl("/security/login/failure");
http.logout()
        .logoutUrl("/spring/logout")
        .logoutSuccessUrl("/security/logout/success")
        .addLogoutHandler(new SecurityContextLogoutHandler());
http.rememberMe()
        .tokenValiditySeconds(5 * 24 * 60 * 60);
```

login.html

```
<form action="/spring/login" method="post">
        <input name="username">
        <input name="password">
        <input name="remember-me" type="checkbox">
        <button>Login</button>
</form>
```

```
<a href="/spring/logout">Sign Out</a>
```

SecurityController

```
@/security/login/form
@/security/login/success
@/security/login/failure
@/security/logout/success
@/security/access/denied
```

Custom Security

USING @PREAUTHORIZE()

```
@Configuration
@EnableWebSecurity
@EnableGlobalMethodSecurity(prePostEnabled = true)
public class SecurityConfig extends WebSecurityConfigurerAdapter {...}
```

```
@PreAuthorize("isAuthenticated()")
@RequestMapping("url")
```

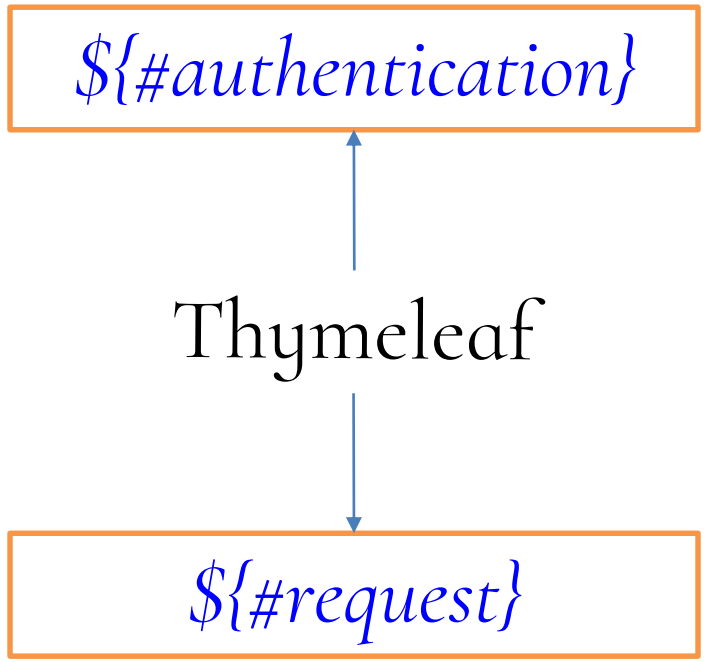
```
@PreAuthorize("hasAnyRole('ADMIN', 'USER')")
@RequestMapping("url ")
```

```
@PreAuthorize("hasRole('ADMIN')")
@RequestMapping("url")
```

Security Programmatically

```
@RequestMapping("/url")
public String method(Authentication auth) {
    String username = auth.getName();
    UserDetails user = (UserDetails) auth.getPrincipal();
    ...
}
```

```
@RequestMapping("/url")
public String method(HttpServletRequest request) {
    String username = request.getRemoteUser();
    if(request.isUserInRole("ADMIN")) { // do something...
    }
    UserDetails user = (UserDetails) request.getUserPrincipal();
    ...
}
```



Authentication & HttpServletRequest API

- **Authentication**
  - getName(): String
  - getAuthorities(): List<GrantedAuthority>
  - isAthenticated(): boolean
  - getPrincipal(): UserDetails
- **HttpServletRequest**
  - getRemoteUser(): String
  - isUserInRole(String): Boolean
  - getUserPrincipal(): UserDetails

- th:object=**${#authentication}**
  - *{name}
  - *{authorities}
  - *{authenticated}
  - *{principal}
- th:object=**${#request}**
  - *{remoteUser}
  - *(isUserInRole(String)}
  - *{userPrincipal}

```
<div th:object="${#authentication}“ th:if=“*{authenticated}“>
    <div th:text=“*{name}"></div>
    <ul th:each="authority: *{authorities}">
        <li th:text="${authority}"></li>
    </ul>
</div>
```

${#authentication}

```
<div th:object="${#request}“ th:if=“*{remoteUser != null}“>
    <div th:text=“*{remoteUser}"></div>
    <ul th:each="authority: *{userPrincipal.authorities}">
        <li th:text="${authority}"></li>
    </ul>
</div>
```

${#request}

THYMELEAF EXTRAS SPRING SECURITY

```
<html
    xmlns:th="http://www.thymeleaf.org"
    xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity5">

    <div sec:authentication="name"></div>
    <div sec:authorize="isAuthenticated()"></div>
    <div sec:authorize="hasRole('ADMIN')"></div>
    <div sec:authorize="hasAnyRole('ADMIN', 'USER')"></div>
</html>
```

Summary

- ☑ Security Configuration
  - ☑ AuthenticationManagerBuilder
  - ☑ HttpSecurity
  - ☑ WebSecurity
  - ☑ PasswordEncoder
- ☑ Login UI
  - ☑ HttpBasic
  - ☑ HTML Login Form
- ☑ Other Protected Ways
  - ☑ @PreAuthorize
  - ☑ Programmatically
  - ☑ Thymeleaf/ Thymeleaf Extras


Thank you