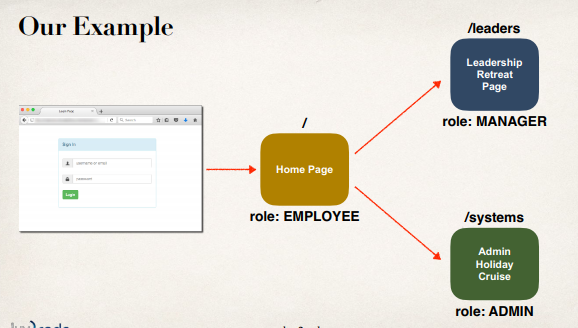
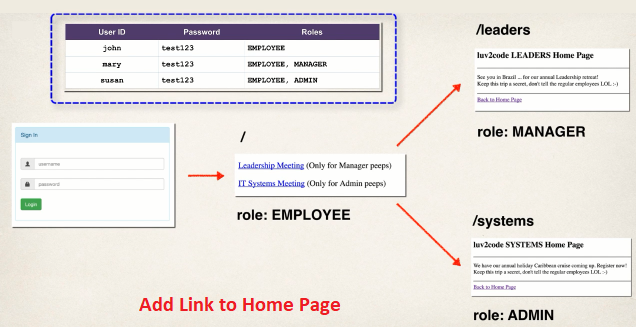
**53.2. Spring Security - Restrict Access – Coding**

**Update Home Page**:

In home page we are going to add links to go to "**/leaders**" for the Leadership page and also "**/systems**" for the Admin page.



In our Home page in the center, we have to add some link for point to the given pages.



**Development Process (Step-by-Step)**:

1. Create supporting controller code and view pages
   1. Update Home page
   2. Create **leaders.jsp** page
   3. Create **systems.jsp** page
   4. Update controller
2. Update user roles
3. Restrict Access based on Roles

**1.a – Update Home page**:

<hr />

<p>

<!-- Add a link to point to /leaders ... this is for the managers -->

<a href=*"*${pageContext.request.contextPath}*/leaders"* >LeaderShip Meeting</a>

(Only for Manager peeps)

</p>

<hr />

<p>

<!-- Add a link to point to /systems ... this is for the admins -->

<a href=*"*${pageContext.request.contextPath}*/systems"* >IT Systems Meeting</a>

(Only for Admins peeps)

</p>

<hr />

**1.b – Create leaders.jsp page**:

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>My Leadership Page</title>

</head>

<body>

<h2>Learn Programming Step by Step -> Leaders Page</h2>

<p>

See you in Brazil ... for our annual Leadership retreat!

<br />

Keep this trip a secret, don't tell the regular employees :-)

</p>

<hr />

<a href=*"*${pageContext.request.contextPath}*/"*> Back to Home Page</a>

</body>

</html>

**1.c – Create systems.jsp page**:

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>SYSTEMS Home Page</title>

</head>

<body>

<h2>Our SYSTEMS Home Page</h2>

<hr />

<p>

We have our annual holiday Caribbean cruise coming up. Register now!!!

<br />

Keep this trip a secret, don't tell the regular employees :-)

</p>

<hr />

<a href=*"*${pageContext.request.contextPath}*/"*> Back to Home Page</a>

</body>

</html>

**1.d – Update Controller (DemoController.java)**:

// add request mapping for /leaders

@GetMapping("/leaders")

**public** String showLeaders() {

**return** "leaders";

}

// add request mapping for /systems

@GetMapping("/systems")

**public** String showSystems() {

**return** "systems";

}

**2. Update user roles**:

auth.inMemoryAuthentication()

.withUser(users.username("john").password("test123").roles("EMPLOYEE"))

.withUser(users.username("mary").password("test123").roles("EMPLOYEE", "MANAGER"))

.withUser(users.username("susan").password("test123").roles("ADMIN", "EMPLOYEE"));

Now when we run our project, we will see the login page and after logging we will see our home page. In our home page we will see the link of **leaders.jsp** page. But this link is accessible for all roles (**EMPLOYEE**, **MANAGER**, and **ADMIN**). Here is the problem.

We have to restrict our access based on roles.

**3. Restrict access based on Roles**:

@Override

//Configure security of web paths in application, login, logout etc.

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

//Restrict access based on the HttpServletRequest

http.authorizeRequests()

// Restrict access based on roles

.antMatchers("/").hasRole("EMPLOYEE")

.antMatchers("/leaders/\*\*").hasRole("MANAGER")

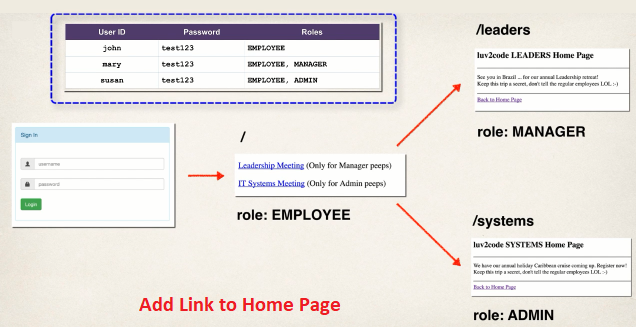
.antMatchers("/systems/\*\*").hasRole("ADMIN")

.and()

...

}

Now our access is restricted based on roles.



53.2. Spring Security - Restrict Access - Coding