**53.1. Spring Security - Restrict Access – Overview**

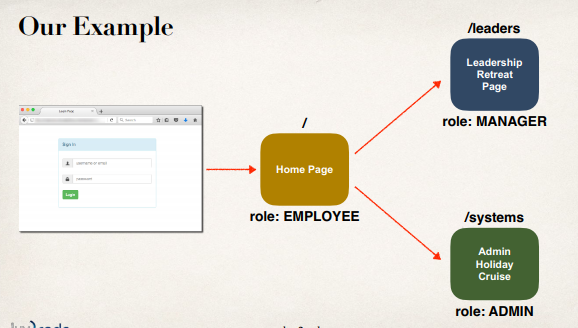
**Restrict Access Based on Roles**:

Here we will restrict access based on roles.

Here we have a login page. Once we login we will go to our home page. Home page will be restricted to only people with the role of **Employee**.

And then we will go to another page "**/leaders**". This is the leadership restrict page. This will only be available to users with the role of **Manager**.

And then we will have another link to a page. "**/systems**". This will only be available to users with the role **Admin**.



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

1. Create supporting controller code and view pages
2. Update user roles
3. Restrict Access based on Roles

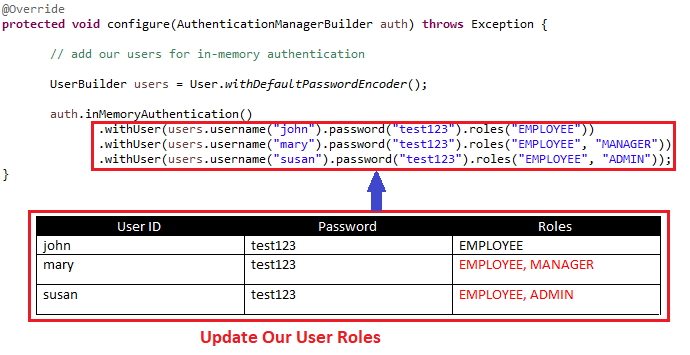
**Step-1: Create supporting controller code and view pages**:

This is all plain vanilla Spring MVC work like setting up request mapping and setting up JSP pages. Nothing security specific.

**Step-2: Update user Roles**:

|  |  |  |
| --- | --- | --- |
| User ID | Password | Roles |
| john | test123 | EMPLOYEE |
| mary | test123 | EMPLOYEE, MANAGER |
| susan | test123 | EMPLOYEE, ADMIN |

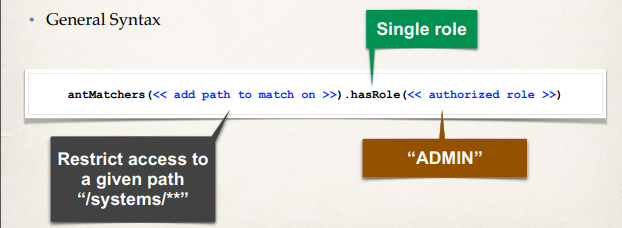
We can give any names for user roles.



**Step 3: Restricting Access to Roles**:

* Update your Spring Security Java configuration file (.java)
* General Syntax

**Diagram**:





**Code**:

//Restrict access based on the HttpServletRequest

http.authorizeRequests()

// strict access based on specific role

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

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

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

.and()

...



53.1. Spring Security - Restrict Access - Overview