* In unele cazuri, s-ar putea sa nu vrem sa afisam anumite parti ale paginii pentru alte roluri.
* De ex, linkul de acces la Panoul de Control va fi prezent in pagina JSP, dar numai adminii trebuie sa il poata vedea.
* Pentru asta, folosim Spring Security JSP Tags



deci, tot ce se afla intre tagul <security:authorize> </security:authorize> va fi afisat pentru cei care au rolul indicat.

* Acum, partea cea mai buna e ca acest continut nu va fi nici macar adaugat in pagina HTML, deci cu inspecteaza sursa, userii neautorizati nu vor putea vedea asta in HTML codes.

Info about you:  
<**security:authentication** property="principal.username"/>  
<**security:authentication** property="principal.authorities"/>  
<hr>  
<br>  
  
<**security:authorize** access="hasRole('MANAGER') || hasRole('ADMIN')">  
<p>  
 <a href="/leaders">LeaderShip Meeting(ONLY FOR MANAGERS)</a>  
</p>  
</**security:authorize**>  
 <br>  
  
<**security:authorize** access="hasRole('ADMIN')">  
<p>  
 <a href="/systems">Systems Meeting(ONLY FOR ADMINS)</a>  
</p>  
</**security:authorize**>  
<br>  
<**form:form** method="post" action="logout">  
 <input type="submit" value="Logout"/>  
</**form:form**>

Vedem ca hasRole() ia doar un parametru, dar ea returneaza true sau false, deci putem sa o combinam cu || au && si tot asa.

Atentie! Metoda hasRole() functioneaza asa doar aici in JSP