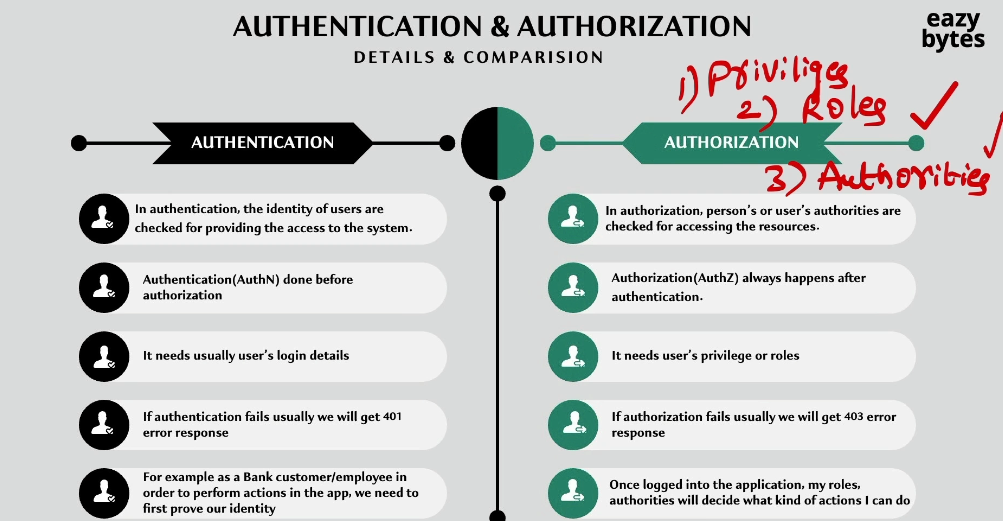
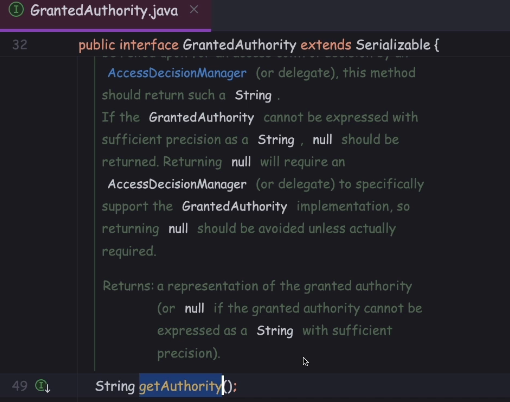
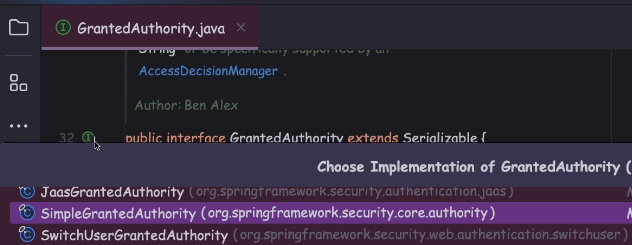
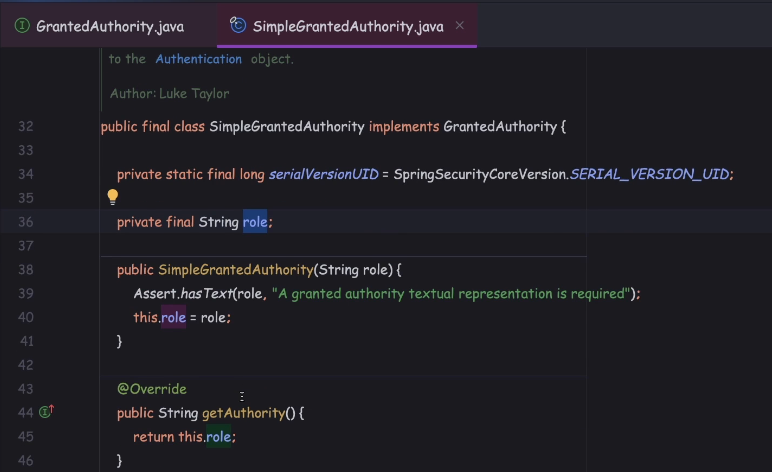
**8.authorization using authorities and roles section9**



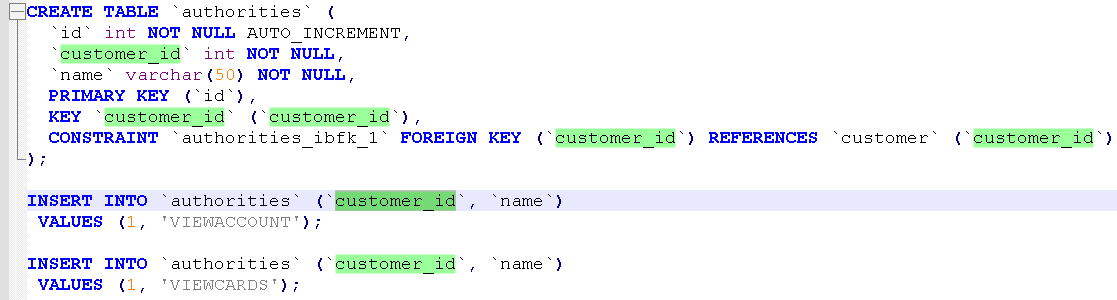
How Authorities stored inside Spring Security

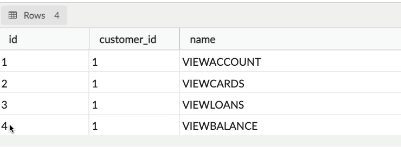




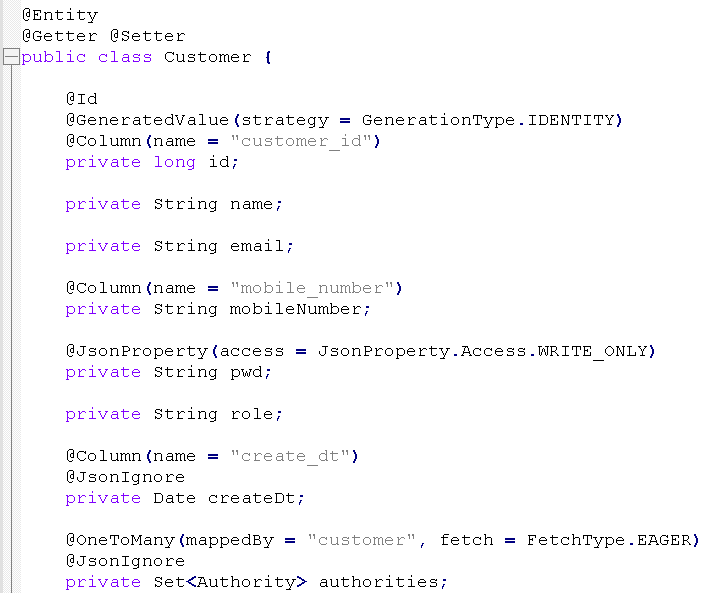
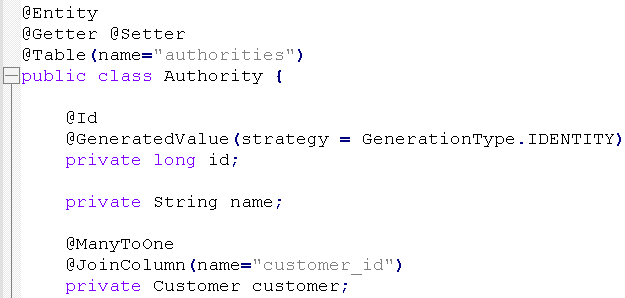


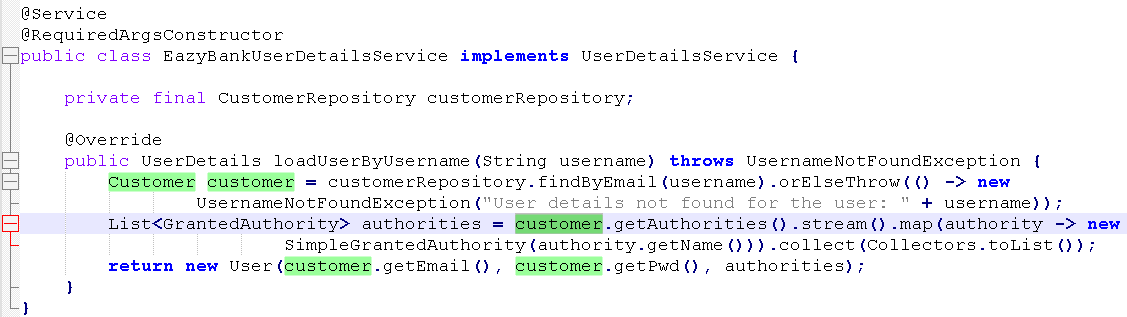
Creating new table authorities to store multiple roles or authorities



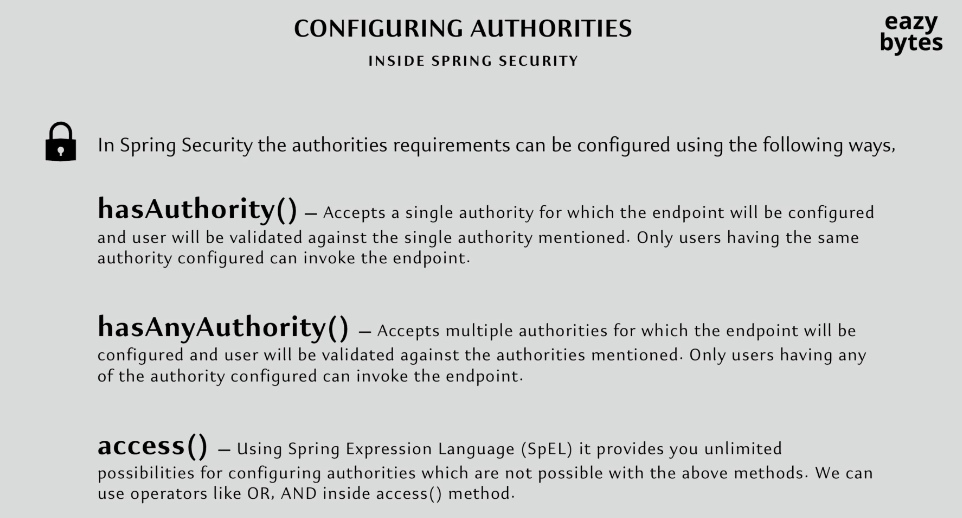


Making backend changes to load authorities from new DB table

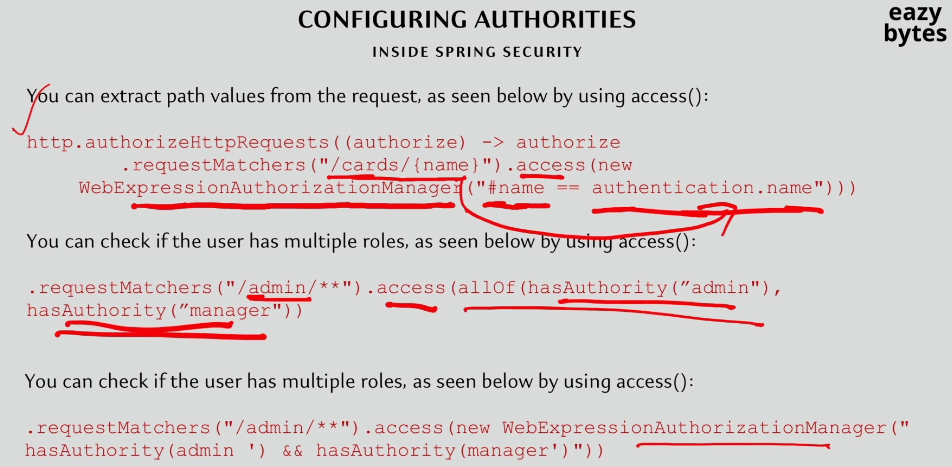
 



 Configuring Authorities inside web application using Spring Security







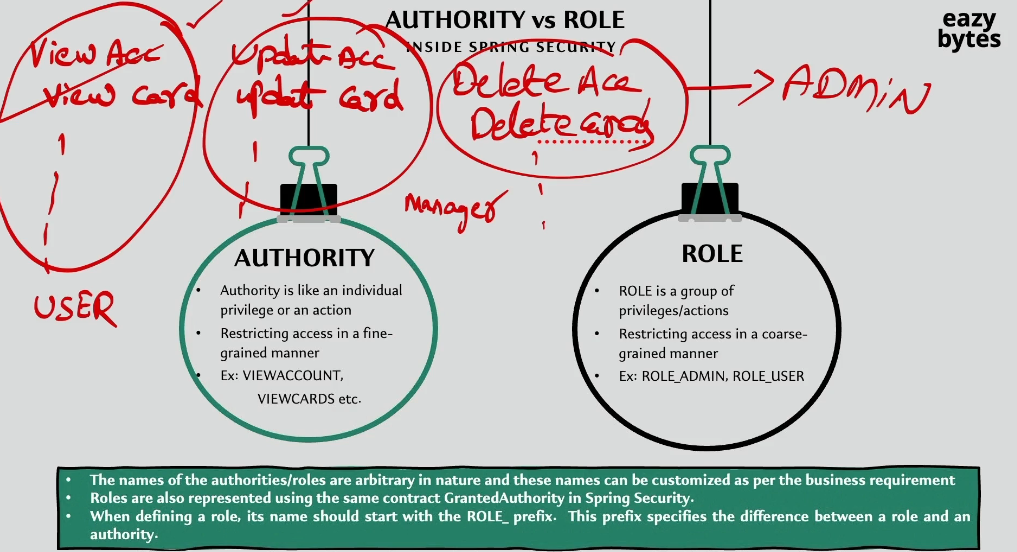
 Authority Vs Role in Spring Security

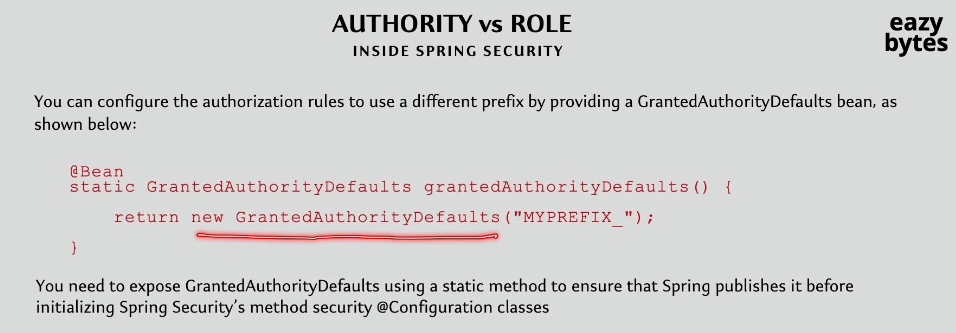
So far, we implemented authorization with the help of authorities.

Apart from authorities, Spring Security also allow us to enforce authorization

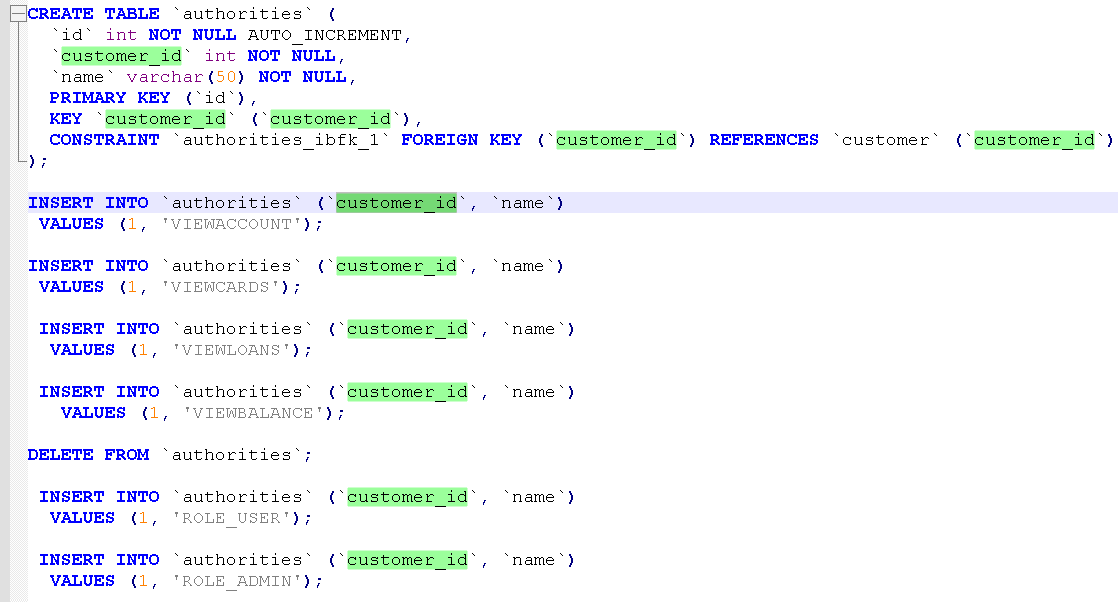
with the help of ROLES concept. There is a very minute difference

between AUTHORITY and ROLE.

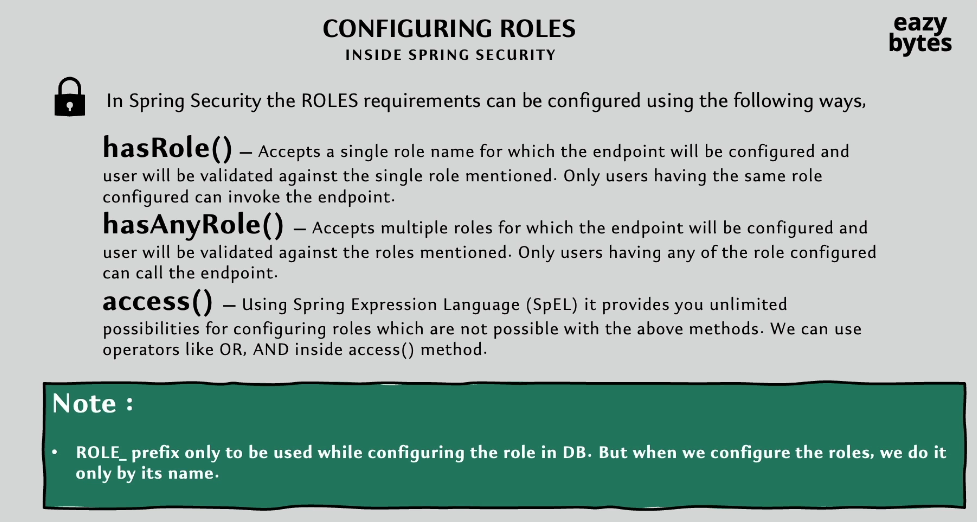




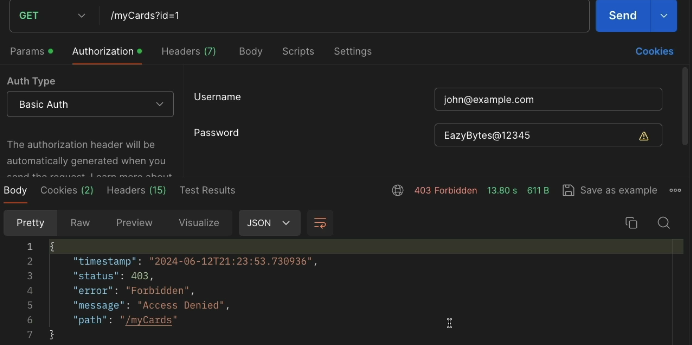
Configuring Roles Authorization inside web application using Spring Security







Listening to the Authorization events



Apart from handling the 403 on the client side, sometimes we may also want

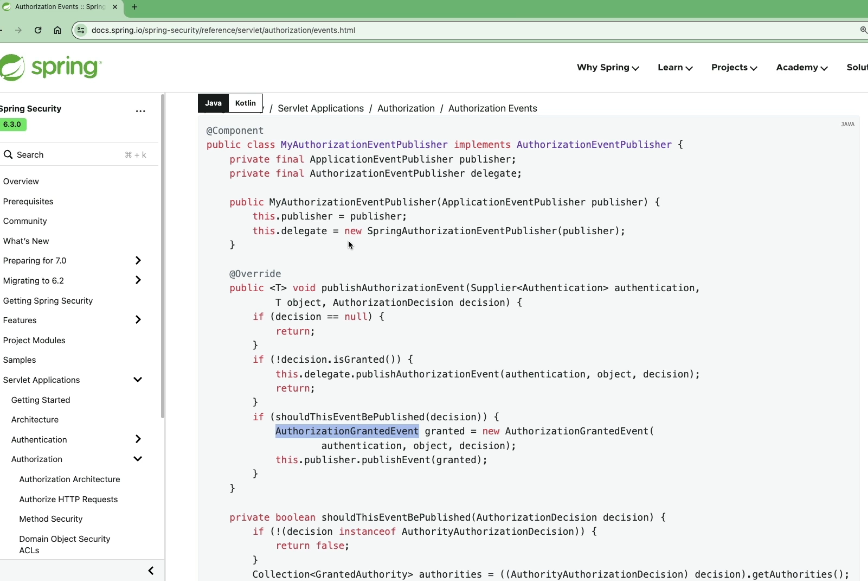
to trigger some business logic on the backend side as well.

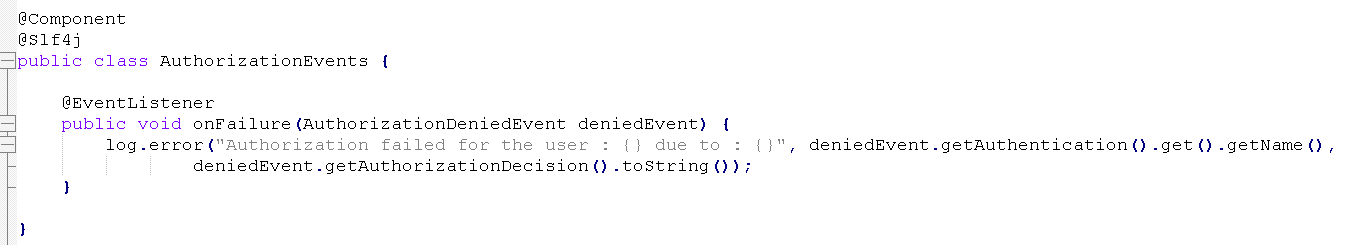
Maybe we want to trigger an email or maybe we want to make an audit entry

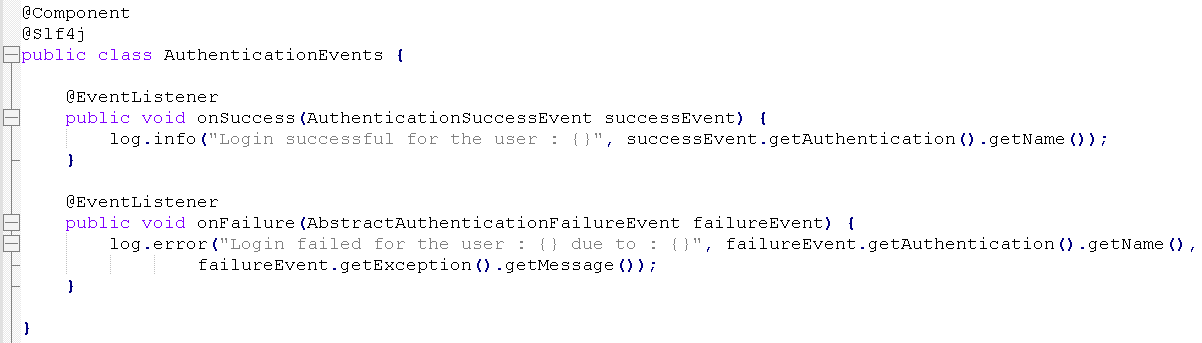
inside the database or maybe we want to write a error log message onto the logs.

The requirement can be anything. To support these kind of requirements,

Spring security framework, is going to publish an event whenever the authorization is failing. We as a Java developers, we just have to listen to that event and trigger the business logic that we want to execute.







Top of Form

Which of the following best describes Authorization in any web application?

**During Authorization, the person’s or user’s authorities are checked for accessing the resources.**

Top of Form

Which of the following best describes Authentication in any web application?

**During Authentication, the identity of users are checked for providing the access to the system.**

Top of Form

Which of the following is the correct combination of HTTP error codes for Authentication and Authorization failure scenarios?

**Authentication : 401, Authorization : 403**

Top of Form

Inside UserDetails interface which is a contract of the user inside the Spring Security, the authorities of an user stored in the form of collection of which interface?

**org.springframework.security.core.GrantedAuthority**

Top of Form

Which of the following methods can be used to configure authorities for a specific URL?

* 

**hasAuthority()**

* 

**hasAnyAuthority()**

* 

**access()**

* 

**All of the above**

Bottom of Form

Top of Form

Which of the following is the correct statement regarding Authority inside Spring Security?

* 

**Authority is like an individual privilege and represents fine-grained access like VIEWACCOUNTS, VIEWLOANS etc.**

* 

**Authority is a group of privileges and represents coarse-grained access like ROLE\_ADMIN, ROLE\_USER.**

* 

**Both Authority and Role are same in Spring Security.**

Top of Form

Which of the following methods can be used to configure roles based access for a specific URL inside Spring Security ?

* 

**hasRole()**

* 

**hasAnyRole()**

* 

**access()**

* 

**All of the above**

Bottom of Form

Top of Form

Which of the following code snippet is correct that can listen and handle authorization denied events ?@EventListener

* 1. public void onFailure(AuthorizationDeniedEvent deniedEvent) {
  2. // Logic to handle authorization denied event
  3. }