Roles and Permissions in Java

By Thomas Pouw

1. Contents

[2. Roles and Permission and how to implement it in Java 2](#_Toc105662849)

[3. Why is it important to implement Roles and Permissions 2](#_Toc105662850)

[4. Roles and Permissions in Auth0 2](#_Toc105662851)

[5. Roles and Permissions in Java/SpringBoot 3](#_Toc105662852)

[6.Conclusion 3](#_Toc105662853)

[7. References 4](#_Toc105662854)

# 2. Roles and Permission and how to implement it in Java

Roles and Permission is the backbone of authentication, as it is like the company id of a person. It authenticates that you work for that company and that you are not just a stranger that is walking into a random building. But how are those Roles and permissions made in Auth0 and how can you use it in Java/SpringBoot?

# 3. Why is it important to implement Roles and Permissions

Roles and Permissions are so important for a simple reason. You do not want unauthorised people in some parts of your program. You do not want a normal user looking at the admin panel. You do not want a manager giving admin privileges. So, Roles and Permissions were made to organise them. You an see the Roles part as a standard container of permissions. Otherwise, you must copy paste a lot of permissions between people with the same roles. And nobody likes working double especially if you need to give 10 people 50 permissions each.   
So, you can see the roles as the online version of a do and do not list. But the only different is that the user will not see that list and the user will also not know what parts are blocked for them. Because of permissions you can even control if a user can create, edit, view, or delete certain files. So, if you want somebody to only view something but not edit it you can give that person only the role that has the permission to view the file.

When you first think about roles and permissions it looks so complicated, but when you figure out one part of it becomes 80% more logical. As it is just a list of permissions that the program looks for. Nothing more nothing less. (Zhang, 2020)

# 4. Roles and Permissions in Auth0

Auth0 has a very easy way to add, edit and view Roles and permissions, you just need the API key and other things that depends on the situation. If you want to view permissions of user you will need the user id, if you want to view permissions on roles you need the role id and if you want to add a permission for a user you need the user id and the permission id. Of course, these are not all of them. And I will talk about some more now but those were the easier ones.

As for example if you want to add a brand new permission you have to put the   
permission name/ PERMISSION\_NAME and the permission description/PERMISSION\_DESC into the scope of the data section of the URL.  
There are some restrictions placed on the permission names as address, created\_at, email, email\_verified, family\_name, given name, identities, name, nickname, offline\_access, openid, phone, picture and profile are all fields within the user module that auth0 uses and so the system would be confused if its both a field in the user table and a field in the permission table.  
 (auth0 docs, 2022)

# 5. Roles and Permissions in Java/SpringBoot

To excess roles and permissions in java/SpringBoot you must use @PreAuthorize before the method. In the () you must put the function you want to check, so for Role its “hasRole()” and for authentication/permission it is “hasAuthority()”. And of course, in the has functions you put the role or permission you want to check for.   
To get roles and permission to the backend you have to provide them. And you do that by sending it with the same call as you are making to access the endpoint. If you allow the Authentication header on CORS, it will read that for checking if you can enter. You can see it as a kind of id you have to show for when you buy alcohol. And so, it must be treated like that as well… you do not give an endpoint that has the job to edit product the same authority/permission as an endpoint that just gives a list of bakers. You must have two permission then. It does not matter if they are only used in one role. You are going to make two permissions. (JoseLion, 2022)

# 6.Conclusion

Roles and permissions are easy when you understand why they are needed. As it is just a kind of job description for parts of your code. And most of the work you have to do is already done by things like Auth0 and you just have to call certain URLs to get the information needed. And if you just go on their website, it will give you the URL and the settings you must include to get access to the information. Even though I am talking about auth0 most of the authentication system have kind of the same easy to use wiki pages. But just like the things they provide you must get used to the way they document things before you get what they offer.

# 7. References

auth0 docs. (2022, May 18). *Auth0*. Opgehaald van Manage Role-Based Access Control Permissions: https://auth0.com/docs/manage-users/access-control/configure-core-rbac/manage-permissions

JoseLion. (2022, February 1). *Github.* Opgehaald van Github: https://github.com/auth0-developer-hub/api\_spring-boot\_java\_hello-world/tree/basic-role-based-access-control/src/main/java/com/example/helloworld

Zhang, E. (2020, December 1). *digitalguardian*. Opgehaald van digitalguardian: https://digitalguardian.com/blog/what-role-based-access-control-rbac-examples-benefits-and-more