Roles and Permissions in Java

By Thomas Pouw

1. Contents

[2. Roles and Permission and how to implement it in Java 2](#_Toc105751857)

[3. Why is it important to implement Roles and Permissions 2](#_Toc105751858)

[4. Roles and Permissions in Auth0 2](#_Toc105751859)

[5. Roles and Permissions in Java/SpringBoot 3](#_Toc105751860)

[6.Conclusion 3](#_Toc105751861)

[7. References 4](#_Toc105751862)

# 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 important for your project. As one of the ways to secure your endpoints from unwanted visitors its necessary to have roles and permissions correctly implemented. And you do not want unwanted visitors in your endpoints as some endpoints can hold more secure information that a certain user should not need to see. So, a normal user should not see all the data that an algorithm has collected or be able to make and change roles and permissions. And maybe you have a kind of hierarchy of owners, Head admins, admins, and moderators. There a good role system is also important. Or even when you are training a new moderator its good to have a different role for them. As you can not really trust them yet 100%. As sometimes they mess something up or just leak information to the outside world. Sometimes it is just better safe then sorry. Especially with power hungry people and not the brightest people around being on your website.  
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 is an authentication system created by Eugenio Pace and Matias Woloski in 2013 and ever since then Auth0 grew into the company we know today.

(Pace, 2022)

Auth0 has a good and easy way to add, edit and view roles and permissions as they made sure you could do that with just one URL request, or two if you count getting the API key as a URL request which it technically is.   
To add for example a role to the use you need three things, a User ID/sub, the Role ID, and the API key to get allowed by Auth0 to change anything within the Auth0 borders. To view certain roles permissions or permissions in your application you need the API key and the ID of your chosen subject. So, for permissions you need the permission ID. But also say in the URL that you are looking for a permission as the API otherwise could not really know what it is looking for.

You can also add new permissions by giving the system the new permission name and a permission descriptions. But you can not name it everything as 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 are not allowed to also be permission names.  
 (auth0 docs, 2022)

# 5. Roles and Permissions in Java/SpringBoot

To be able to use Auth0 in Java/SpringBoot you have to make sure that your security function is not on permit all and add @PreAuthorize before any function that needs it. So, if you want a function to have the restriction of only usable if the user is an admin you add “hasRole()” and set role in the @PreAuthorize. So that the system knows what to look for. It also counts for permissions, and everything said about add and view in the last chapter also applies here. As Auth0 does not care from what language that URL request came from. When they get it, they will check if you got the right API Key and then process it and give you your information, or an error if you did not give them the right information.

(JoseLion, 2022)

# 6.Conclusion

Roles and Permission at first look very weird and complicated, but when you look at them, they make so much more sense, as it is just a list of things that the program looks at to see if you are allowed to use a certain thing. You can even compare it to a convention ticket. As you must have one to enter, but normally you can have more things to do if you have more permissions added to it. Maybe you see a panel that you want to see, but you can only go there with the “See\_Panel” permission in the “User+” role. So, you just need the right permissions and roles to get the things you sometimes want. But unlike a Convention, you will not miss out on much as you hide things that non authorized people should not see in the first place.

Even if I already talked in full about roles and permissions it is better if you just look at the wiki instead as they have handy guides of what every URL request does and how they work. They even give you premade URL requests with the headers included if the language you chose is in their list.

# 

# 7. References

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

JoseLion. (2022, February 1). *Github.* Retrieved from 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

Pace, E. (2022, June 10). *Auth0 Blog*. Retrieved from Auth0: https://auth0.com/blog/authors/eugenio-pace/

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