

# User Authentication and Application Security part 2

Node.js Accelerator – Jan'23

# Agenda

1. Auth0 Recap

---

2. Auth0 with Passportjs

---

3. Auth0 Roles

---

4. Auth0 Roles Rule

---

5. Permission Middleware

---



# Node.js Accelerator

Jan 2023

From previous lesson we worked with Auth0 with:

- Creating an application and API in Auth0 Dashboard
- Showed email customisation and built in features such as reset password
- Demonstrated one of the ways to attach an Auth0 id into your own database
- Suggested an optional way to use your own database (link [here](#))
- Added middleware to check user authentication

Another approach to control Auth0 in your application is to use [Passport.js](#) (tutorial [here](#))

- You'll need to configure all the routes such as login, logout, callback, redirect and middleware

Let's see a quick look in our code

```
this.app.route( prefix: '/v2/login').get(  
  passport.authenticate( strategy: 'auth0', options: {  
    scope: 'openid email profile',  
  } ),  
),
```


You can check the supplemental Udemy course about Passportjs [here](#)


Having the access token already is great but we need to be able to control permissions.

In Auth0 we can setup Roles and manage them:

 Getting Started

 Activity

 Applications >

 Authentication >

 Organizations

 User Management ▾

Users

Roles

 Branding >

 Security >

## Roles

Create and manage Roles for your applications. Roles contain collections of Permissions and can be assi

Name	Description
<a href="#">admin</a>	all possible admin actions
<a href="#">normal-user</a>	sample user

Once having roles set, you can attach them to the accessToken and/or the idToken

```
function (user, context, callback) {  
  var namespace = 'https://myrest.api/';  
  if(context.accessToken && context.authorization.roles){  
    context.accessToken[namespace + 'claims/' + "roles"] = context.authorization.roles;  
  }  
  if (context.idToken && user.user_metadata) {  
    context.idToken[namespace + 'user_metadata'] = user.user_metadata;  
    context.accessToken[namespace + 'user_metadata'] = user.user_metadata;  
  }  
  if (context.idToken && user.app_metadata) {  
    context.idToken[namespace + 'app_metadata'] = user.app_metadata;  
    context.accessToken[namespace + 'app_metadata'] = user.app_metadata;  
  }  
}
```

Let's move to the code and check one of the ways to control the permission from a request





# ACTION ITEMS

Week 4

1. We strongly recommend you to explore below module through Udemy this week and learn about Node.js essentials

Module	Topic	Description	Udemy Link	
Self-paced learning on Udemy	Optional: Dynamic Routing & modeling	Practice what you learned, give it a try to code navigation, routing while you implement an online shopping project	<a href="#">Click Here</a>	2 Hrs 15 Min
Self-paced learning on Udemy	Integrating with MongoDB	Learn about MongoDB, installing Mongo, connect to a mongodb database using mongoose library, modeling, CRUD operations, data validation, mongodb aggregation pipeline etc.	<a href="#">Click here</a>	6 Hrs 20 min

