

# BE NodeJS

## Tugas: API Auth

Leonardho R. Sitanggang

Jawab (Penerapan Authorization) :

- Buat middleware

```
tugas-api-auth > src > middlewares > TS rbac.middleware.ts > default
1  import { Response, NextFunction } from "express";
2  import { IRequestWithUser } from "../auth.middleware";
3
4  export default (roles: string[]) =>
5  (req: IRequestWithUser, res: Response, next: NextFunction) => {
6      const userRoles = req.user?.roles;
7
8      if (!userRoles || !userRoles.some((userRole) => roles.includes(userRole))) {
9          return res.status(403).json({
10             message: "Forbidden",
11         });
12     }
13
14     next();
15 }
```

- Daftarkan akun baru dengan role yang sesuai pada config route (Admin)

```
6
7  // Auth
8  router.post("/auth/login", authController.login);
9  router.post("/auth/register", authController.register);
10 router.get("/auth/me", [authMiddleware, rbacMiddleware(["admin"])], authController.me);
11 router.put("/auth/update-profile", authMiddleware, authController.profile);
12
```

POST http://localhost:3000/api/auth/register Send

Query Headers<sup>2</sup> Auth **Body<sup>1</sup>** Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "email": "admin@toko23.com",
3   "fullName": "Admin",
4   "password": "12341234",
5   "username": "admintoko23",
6   "roles":["admin"]
7 }
```

Status: 200 OK Size: 280 Bytes Time: 492 ms

Response Headers<sup>6</sup> Cookies Results Docs

```
1 {
2   "message": "registration success!",
3   "data": {
4     "fullName": "Admin",
5     "username": "admintoko23",
6     "email": "admin@toko23.com",
7     "roles": [
8       "admin"
9     ],
10    "profilePicture": "user.jpg",
11    "_id": "673cb5b2e865f672e8b2a5df",
12    "createdAt": "2024-11-19T15:58:42.503Z",
13    "updatedAt": "2024-11-19T15:58:42.503Z",
14    "__v": 0
15  }
16 }
```

- Login untuk mendapatkan token JWT

POST http://localhost:3000/api/auth/login Send

Query Headers<sup>2</sup> Auth **Body<sup>1</sup>** Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "email": "admin@toko23.com",
3   "password": "12341234"
4 }
```

Status: 200 OK Size: 232 Bytes Time: 57 ms

Response Headers<sup>6</sup> Cookies Results Docs {} ≡

```
1 {
2   "message": "login success",
3   "data": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY3M2NiNWlyZTg2NWY2NzJlOGIyYTVkZiIsInJvbGVzIjpjbImFkbWl1I0sImldCI6MTczMjAzMTk5MSwiZmxwIjozNzMyMDM1NTkxfQ.cqJyQ88jB08dwYlB1kQt5-uh7Ip0jv0W7-UjTZ0zSAM"
4 }
```

- Gunakan token JWT sebagai bearer token pada header

GET http://localhost:3000/api/auth/me Send

Query Headers<sup>2</sup> **Auth<sup>1</sup>** Body Tests Pre Run

None Basic **Bearer** OAuth 2 NTLM AWS

Bearer Token

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY3M2NiNWlyZTg2NWY2NzJlOGIyYTVkZiIsInJvbGVzIjpjbImFkbWl1I0sImldCI6MTczMjAzMTk5MSwiZmxwIjozNzMyMDM1NTkxfQ.cqJyQ88jB08dwYlB1kQt5-uh7Ip0jv0W7-UjTZ0zSAM
```

Status: 200 OK Size: 285 Bytes Time: 169 ms

Response Headers<sup>6</sup> Cookies Results Docs

```
1 {
2   "message": "success fetch user profile",
3   "data": {
4     "_id": "673cb5b2e865f672e8b2a5df",
5     "fullName": "Admin",
6     "username": "admintoko23",
7     "email": "admin@toko23.com",
8     "roles": [
9       "admin"
10    ],
11    "profilePicture": "user.jpg",
12    "createdAt": "2024-11-19T15:58:42.503Z",
13    "updatedAt": "2024-11-19T15:58:42.503Z",
14    "__v": 0
15  }
16 }
```