



JWT Integration



Install jsonwebtoken

```
$ npm install jsonwebtoken@8.5.1
```

Import the module into app.js



Create middleware

Create new file authenticate.js and paste code

```
const jwt = require("jsonwebtoken");
const authenticate = (req, res, next) => {
  try {
    const token = req.headers.authorization.split(" ")[1];
    jwt.verify(token, "Testing");
    next();
  } catch (error) {
    console.log("Here error", error);
    res.status(401).json({ message: "Auth failed!" });
  }
};
module.exports = authenticate;
```



Import authenticate

Import authenticate into app.js

```
const authenticate= require("../middleware/authenticate")
```



FE interceptor

```
constructor(private jwtHelper: JwtHelperService) { }

intercept(request: HttpRequest<any>, next: HttpHandler) {
  const token = localStorage.getItem('token');

  if (token && !this.jwtHelper.isTokenExpired(token)) {
    request = request.clone({
      setHeaders: {
        Authorization: `Bearer ${token}`
      }
    });
  }

  return next.handle(request);
}
```

Add into app.module.ts: { provide: HTTP_INTERCEPTORS, useClass: JwtInterceptorService, multi:true }



Modifying Sign In Business Logic

```
const token = jwt.sign(  
  {  
    email: findedUser.email,  
    userId: findedUser._id,  
    userRole: findedUser.role,  
  },  
  "Testing",  
  { expiresIn: "1min" }  
);  
let userToSend = {  
  id: findedUser._id,  
  firstName: findedUser.firstName,  
  lastName: findedUser.lastName,  
  role: findedUser.role,  
  jwt: token,  
  expiresIn: 60,  
};
```



UserService

```
userUrl: string = "http://localhost:3000/allUsers" ;
public token: string;
private authStatusListener = new Subject<boolean>();
private isUserAuthenticated = false;

private name: string;

constructor(private http: HttpClient, private router: Router) { }

getToken() {
    return this.token;
}

getAuthStatusListener() {
    return this.authStatusListener.asObservable();
}

isUserAuth() {
    return this.isUserAuthenticated;
}

getName(){
    return this.name;
}

signup(obj, img: File) {
    let formData = new FormData();
    formData.append( "firstName", obj.firstName);
    formData.append( "lastName", obj.lastName);
    formData.append( "email", obj.email);
    formData.append( "pwd", obj.pwd);
    formData.append( "img", img);
    return this.http.post<{ message: string }>( this.userUrl + "/subscription", formData);
}

editProfile(newUser) {
    return this.http.put( this.userUrl, newUser);
}
```

```
login(user) {
    this.http.post<{ user: any, message: string }>( this.userUrl + "/signin", user).subscribe(
        (res) => {
            const token = res.user.jwt;
            this.token = token;
            if (token) {
                this.isUserAuthenticated = true;
                this.name = res.user.firstName;
                this.authStatusListener.next( true);
                localStorage.setItem( 'token', token);
                localStorage.setItem( 'name', res.user.firstName);
                this.router.navigate([ '/']);
            }
        }
    )
}

logout() {
    localStorage.removeItem( 'token');
    localStorage.removeItem( 'name');
    this.isUserAuthenticated = false;
    this.authStatusListener.next( false);
    this.router.navigate([ '/']);
}
```



Header Component

```
userIsAuthenticated = false;
user: any;
private authListenerSubs: Subscription;
constructor(private userService: UserService) { }

ngOnInit() {
  this.authListenerSubs = this.userService.getA <li><a routerLink="signin" class="nav-link"
*ngIf="!userIsAuthenticated" >Login</a></li>
    <li><a class="nav-link" *ngIf="userIsAuthenticated" >{{user}}</a></li>
    <li><a *ngIf="userIsAuthenticated" (click)="logout()" >Logout</a></li>
    <li><a *ngIf="userIsAuthenticated" routerLink="subscription"
class="nav-link">Signup</a></li>
  uthStatusListener().subscribe(
    isAuthenticated => {
      this.userIsAuthenticated = isAuthenticated;
      this.user = this.userService.getName();
    }
  )
}

ngOnDestroy() {
  this.authListenerSubs.unsubscribe();
}

logout() {
  this.userService.logout();
}
```

```
<li><a routerLink="signin" class="nav-link"
*ngIf="!userIsAuthenticated" >Login</a></li>
    <li><a class="nav-link"
*ngIf="userIsAuthenticated" >{{user}}</a></li>
    <li><a
*ngIf="userIsAuthenticated"
(click)="logout()" >Logout</a></li>
    <li><a
*ngIf="!userIsAuthenticated"
routerLink="subscription"
class="nav-link">Signup</a></li>
```