

Working with Bearer Tokens in Angular



Paul D. Sheriff

BUSINESS SOLUTIONS ARCHITECT, FAIRWAY TECHNOLOGIES, INC.

www.fairwaytech.com psheriff@fairwaytech.com



Goals



Pass bearer token to server

Create HTTP interceptor

Add security policy



Pass Bearer Token to Server



Pass Bearer Token to Server

Inject security service into
product service

Create HttpHeaders

Set Authorization header to
bearer token

Pass header with Web API
get() call



Demo



Pass bearer token to server

Secure the complete product controller



Create HTTP Interceptor



Don't want to pass
bearer token each time

Lots of code to write

Create HTTP interceptor

Intercepts each
HTTP call

Adds Authorization
header

All calls have the header
added automatically

```
import { Injectable, NgModule } from '@angular/core';
import { Observable } from 'rxjs/Observable';
import { HttpEvent, HttpInterceptor, HttpHandler,
      HttpRequest } from '@angular/common/http';
import { HTTP_INTERCEPTORS } from '@angular/common/http';

@Injectable()
export class HttpRequestInterceptor implements HttpInterceptor {
  intercept(req: HttpRequest<any>, next: HttpHandler):
    Observable<HttpEvent<any>> {
    var token = localStorage.getItem("bearerToken");

    if(token) {
      const newReq = req.clone(
        {
          headers: req.headers.set('Authorization',
            'Bearer ' + token)
        });
      return next.handle(newReq);
    }
    else {
```



Demo



Create HTTP interceptor



Add Security Policy



Add Security Policy

Secure method
based on claim

Configure service
to use policies

```
[Authorize(Policy  
= "ClaimType")]
```



Add code in
ConfigureServices()

Add a policy name

Specify claim type
and value

```
services.AddAuthorization(cfg =>
{
    // NOTE: The claim type and value are case-sensitive
    cfg.AddPolicy("CanAccessProducts", p =>
        p.RequireClaim("CanAccessProducts", "true"));
});
```



Add [Authorize]
attribute with *Policy*
parameter

Only those users with
claim type and value
that match may run
method

```
[Route("api/[controller]")]
[Authorize]
public class ProductController : BaseApiController
{
    [HttpGet]
    [Authorize(Policy = "CanAccessProducts")]
    public IActionResult Get()
    {
        IActionResult ret = null;
        List<Product> list = new List<Product>();

        try
```



Demo



Add security policy



Summary



Passed bearer token to server

Created HTTP interceptor

Added security policy





Coming up in the next module...

Enterprise security

Create array of user claims from a table

Create structural directive for securing UI

