

```

it('should retrieve company user data', async () => {
  await authService.SignIn('wasaf62813@evvgo.com', '123456');
  await component.getUserData();
  const Company = component.myUser as CompanyDTO;
  expect(Company.CompanyName).toEqual('Better call Saul');
});

it('should retrieve employee user data', async () => {
  await authService.SignIn('sanic29650@gosarlar.com', '123456');
  await component.getUserData();
  const Employee = component.myUser as EmployeeDTO;
  expect(Employee.CompanyName).toEqual('Better call Saul');
});

it('should retrieve public user data', async () => {
  await authService.SignIn('dojefe6817@giratex.com', '123456');
  await component.getUserData();
  const Public = component.myUser as UserDTO;
  expect(Public.FirstName).toEqual('Nick');
});

```

You will probably have to use the functions (`await authService.SignIn`) above for your testing, if you do you need the setting below

```

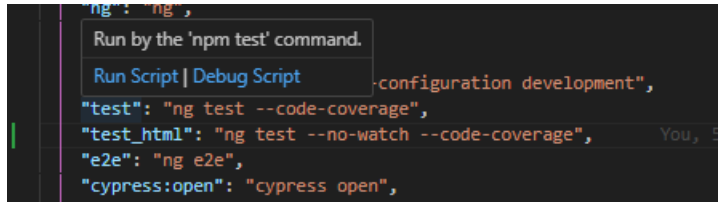
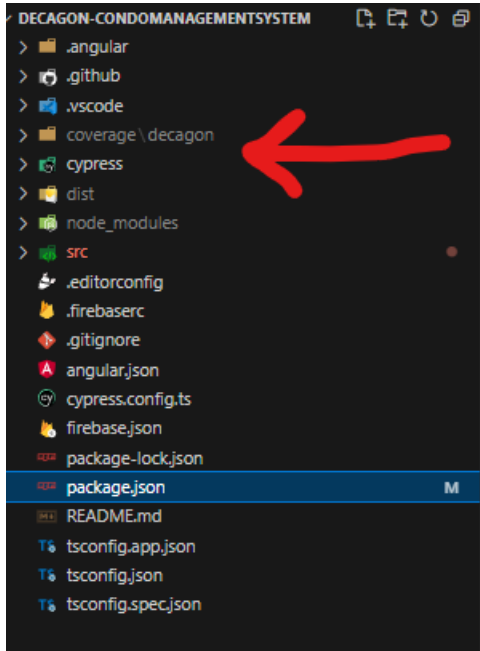
let component: HeaderComponent;
let fixture: ComponentFixture<HeaderComponent>;
let authService: AuthService;

beforeEach(() => {
  TestBed.configureTestingModule({
    imports: [AppModule],
    declarations: [HeaderComponent],
  });
  fixture = TestBed.createComponent(HeaderComponent);
  component = fixture.componentInstance;
  fixture.detectChanges();
  authService = TestBed.inject(AuthService);
});

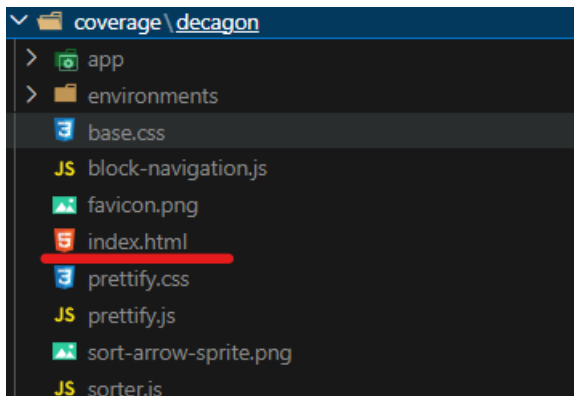
afterEach(async () => {
  await authService.SignOut();
});

```

npm run test\_html will create the folder below



Then you open the html folder below in a browser to see the code coverage (ex: can use extension live server)



Make sure to write run and pass the test before opening a pr with “npm run test” ber sure to have at least 80% statement coverage like the image below.