|  |
| --- |
|  |
| Angular 8 |
|  |
| Angular – Login Logout |

**TechBrain Express**

December 24, 2019

Authored by: Ashu

Angular – Login And Logout

* **Demonstrate authentication using SessionStorage**

<https://www.javainuse.com/spring/ang7-login>

## Client :

The two most important types of storage properties of any browser are the localStorage and the sessionStorage; they allow you to store key-value pairs in the browser.

* localStorage  
  It is a persistent storage option when you don’t want the user to be logged out on refresh or closing of a browser tab. The localStorage stores the data without expiration date by default. It is a read-only property.
* sessionStorage  
  As the name indicates, it stores data for a particular session only. It is less persistent, and the user data gets lost on the closing of the browser tab or refreshing.

Lab: Activate the logout and Home tab only after Login .

Hint: Setup AuthGuard (Deactivate/Activate)

1. Create new project

## Setting Bootstrap

* ng new loginlogoutdemo
* Install bootstrap/jquery/popper.js
* Make sure added in node\_module
* Update angular.json to set above script and style

"styles": [

"src/styles.css",

"./node\_modules/bootstrap/dist/css/bootstrap.min.css"

],

"scripts": [ "./node\_modules/jquery/dist/jquery.min.js",

"./node\_modules/bootstrap/dist/js/bootstrap.min.js"

]

* Update index.html

<link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">

* Create navbar component



## Create Home and Login and Logout

Extra: <https://colorlib.com/wp/template/login-form-v12/>

* Create Login component

<hr>

<div class="container">

<div>

User Name : <input type="text" name="username" [(ngModel)]="username">

Password : <input type="password" name="password" [(ngModel)]="password">

</div>

<button (click)=checkLogin() class="btn btn-success">

Login

</button>

</div>

## Update app.component.html

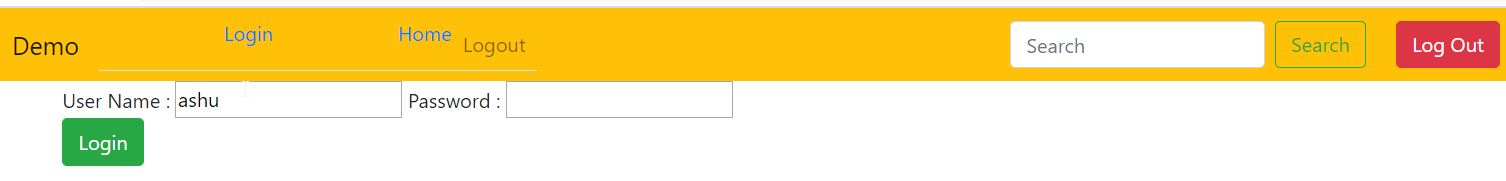
<app-navbar></app-navbar>

<router-outlet></router-outlet>

## Update app.module.ts

Add Forms module

## Create Home Component

* Set Route
* 

## Set Route

const routes: Routes = [

{path: 'Home', component: HomeComponent },

{ path: 'Login', component: LoginComponent},

{ path: 'Logout', component: LogoutComponent }

];

## Create Service AuthService

import { Injectable } from '@angular/core';

@Injectable({

providedIn: 'root'

})

export class AuthenticationServiceService {

constructor() { }

authenticate(username, password) {

if (username === "Ashu" && password === "password") {

sessionStorage.setItem('username', username)

return true;

} else {

return false;

}

}

isUserLoggedIn() {

let user = sessionStorage.getItem('username')

console.log(!(user === null))

return !(user === null)

}

logOut() {

sessionStorage.removeItem('username')

}

}

## Update Login.html

<div class="collapse navbar-collapse" id="navbarText">

<ul class="navbar-nav mr-auto">

<ul class="nav nav-tabs">

<li routerLinkActive="active">

<a \*ngIf="loginService.isUserLoggedIn()" routerLink="Home" style="padding-left:100px">Home </a>

</li>

<li routerLinkActive="active">

<a routerLink="Login" style="padding-left:100px">Login </a>

</li>

<li class="nav-item">

<a \*ngIf="loginService.isUserLoggedIn()" class="nav-link" href="Logout">Logout</a>

</li>

</ul>

</ul>

<form class="form-inline my-2 my-lg-0">

<input class="form-control mr-sm-2" type="search" placeholder="Search">

<button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>

</form>

<a href="logout"> <button type="button" class="btn btn-danger ml-4">Log Out</button> </a>

</div>

## Update Logout.component.ts

import { Component, OnInit } from '@angular/core';

import { AuthenticationServiceService } from '../authentication-service.service';

import { Router } from '@angular/router';

@Component({

selector: 'app-logout',

templateUrl: './logout.component.html',

styleUrls: ['./logout.component.css']

})

export class LogoutComponent implements OnInit {

constructor( private authentocationService: AuthenticationServiceService,

private router: Router) { }

ngOnInit() {

this.authentocationService.logOut();

this.router.navigate(['Login']);

}

}

**To activate a particular route only if the user is logged in.**

## Create a Service guard Authguard

* Ng g s authguard

import { Injectable } from '@angular/core';

import { CanActivate, Router, RouterStateSnapshot, ActivatedRouteSnapshot } from '@angular/router';

import { AuthenticationServiceService } from './authentication-service.service';

@Injectable({

providedIn: 'root'

})

export class AuthguradService implements CanActivate {

constructor(private router: Router,

private authService: AuthenticationServiceService) { }

canActivate(route: ActivatedRouteSnapshot, state: RouterStateSnapshot) {

if (this.authService.isUserLoggedIn())

return true;

this.router.navigate(['login']);

return false;

}

}

## Map guard with HomeComponent

import { NgModule } from '@angular/core';

import { Routes, RouterModule } from '@angular/router';

import { LoginComponent } from './login/login.component';

import { HomeComponent } from './home/home.component';

import { LogoutComponent } from './logout/logout.component';

import { AuthguradService } from './authgurad.service';

const routes: Routes = [

{path: 'Home', component: HomeComponent, canActivate:[AuthguradService]},

{ path: 'Login', component: LoginComponent},

{ path: 'Logout', component: LogoutComponent },

];

@NgModule({

imports: [RouterModule.forRoot(routes)],

exports: [RouterModule]

})

export class AppRoutingModule { }