

## Component file

```
import { Component, OnInit, ViewChild } from '@angular/core';
import { CanComponentDeactivate } from '../deactivate.guard'
import { Observable } from 'rxjs';
import { NgForm } from '@angular/forms';

@Component({
  selector: 'app-tab1',
  templateUrl: './tab1.component.html',
  styleUrls: ['./tab1.component.scss']
})
export class Tab1Component implements OnInit, CanComponentDeactivate {
  @ViewChild('frm') public frm: NgForm

  constructor() { }

  fname: string;

  ngOnInit(): void {
  }

  canDeactivate(): Observable<boolean> | Promise<boolean> | boolean {
    if(this.frm.dirty){
      return confirm("Do you want to discard changes!")
    }else{
      return true;
    }
  }
}
```

## Component HTML File

```
<p>tab1 works!</p>
<form #frm="ngForm">
  <input type="text" [(ngModel)]="fname" name="fname" minlength="2" required>
</form>
<a routerLink="../tab2">Tab2</a>
```

## Guard Deactivate

```

import { Injectable } from '@angular/core';
import { CanActivate, ActivatedRouteSnapshot, RouterStateSnapshot, UrlTree,
CanDeactivate } from '@angular/router';
import { Observable } from 'rxjs';

export interface CanComponentDeactivate {
  canDeactivate: () => Observable<boolean>| Promise<boolean> | boolean;
}

@Injectable({
  providedIn: 'root'
})
export class DeactiateGuard implements CanDeactivate<CanComponentDeactivate> {
  canDeactivate(
    component: CanComponentDeactivate,
    next: ActivatedRouteSnapshot,
    state: RouterStateSnapshot): Observable<boolean | UrlTree> | Promise<boolean |
UrlTree> | boolean | UrlTree {
    return component.canDeactivate();
  }
}

```

## Routing File

```

const routes: Routes = [
  {
    path: '', redirectTo: 'tab1', pathMatch: 'full'
  },
  {
    path: 'tab1', component: Tab1Component, canDeactivate: [DeactiateGuard]
  },
  {
    path: 'tab2', component: Tab2Component
  }
];

```