

1. Guards
2. AdminGuard

- [Info](#)
- [Source](#)

## File

src/app/services/admin.guard.ts

## Description

Admin guard, controls whether the user is an administrator or not

## Index

### Methods

- [canActivate](#)

## Constructor

constructor(\_UserService: [UserService](#), \_router: [Router](#))

Defined in [src/app/services/admin.guard.ts:12](#)

Constructor in which we inject User Service and Router service

### Parameters :

Name	Type	Optional
_UserService	<a href="#">UserService</a>	No
_router	<a href="#">Router</a>	No

## Methods

### canActivate

canActivate()

Defined in [src/app/services/admin.guard.ts:31](#)

Returns : [boolean](#)

boolean, if the user is admin returns true if not false

```
import { Injectable } from '@angular/core'
import { CanActivate, Router } from '@angular/router'
import { UserService } from '../user.service'

/**
 * Admin guard, controls whether the user is an administrator or not
 */
@Injectable({
  providedIn: 'root'
})

export class AdminGuard implements CanActivate {
  /**
   * Constructor in which we inject User Service and Router service
   */
  constructor(
    /**
     * variable to save user service
     */
    private _UserService: UserService,

    /**
```

```

        * variable to save router service
        */
        private _router: Router
    ) { }

    /**
     * @returns boolean, if the user is admin returns true if not false
     */
    canActivate(): boolean {
        /**
         * Store the user that is in the session
         */
        let identity = this._UserService.getIdentity()

        if (identity && identity.admin == 1) {
            return true
        } else {
            this._router.navigateByUrl('/')
            return false
        }
    }
}

```

**result-matching ""**

**No results matching ""**