Sibling.component.ts:  
import { Component, OnInit,OnDestroy } from '@angular/core';

import { DataService } from "../data.service";

import { Subscription } from 'rxjs';

@Component({

selector: 'app-sibling',

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

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

})

export class SiblingComponent implements OnInit, OnDestroy {

message!:string;

subscription!: Subscription ;

constructor(private data: DataService) {

}

ngOnInit() {

this.subscription = this.data.currentMessage.subscribe(message => this.message = message);

}

ngOnDestroy() {

this.subscription.unsubscribe();

}

newMessage(){

this.data.changeMessage("Hello from sibling");

}

}

Parent.component.ts:  
import { Component, OnInit,OnDestroy } from '@angular/core';

import { DataService } from "../data.service";

import { Subscription } from 'rxjs';

@Component({

selector: 'app-parent',

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

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

})

export class ParentComponent implements OnInit, OnDestroy {

message!:string;

subscription!:Subscription;

constructor(private data: DataService) { }

ngOnInit() {

this.subscription = this.data.currentMessage.subscribe(message => this.message = message)

}

ngOnDestroy() {

this.subscription.unsubscribe();

}

}

Data-service.ts:

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

import { BehaviorSubject } from 'rxjs';

@Injectable({

providedIn: 'root'

})

@NgModule({

providers:[]

})

export class DataService {

private messageSource = new BehaviorSubject('default message');

currentMessage = this.messageSource.asObservable();

constructor() { }

changeMessage(message: string) {

this.messageSource.next(message)

}

}

