



Showing responses for:

3 ways to communicate between Angular components
Miro Koczka

5.8K

34

Responses



Write a response...



Alok Vishwakarma

Jul 11, 2018

50 claps in advanced without going through article because you are the one who really understand the problem on beginners...

This line made my day... :)

"I chose this title because I think developers struggling with this will be googling it this way"



▼



Pablo Reyes

Oct 22, 2017

There is also the “redux way” which I could cover in another article in the future.

Espero sea pronto



▼



Alex

Aug 22, 2017

IMO, if you’re going to store state in a Service, it’s best to use a Subject or a BehaviorSubject, since a component that relies on SideBarService might not be initialized yet, and won’t receive the emitted event once it’s created.



1 response

▼



Jerson Antonio

Sep 20, 2017

thanks your post helped me a lot, investigate for more than 12 hours and I did not find any solution your post solved my doubt a great experience bro



v



Dileep Kumar Kottakota

Dec 14, 2017 · 1 min read

Hi,

I was gone through the problem where there is no relationship between components but i have to call a function of one component from another component.

May be am wrong, but i have resolved this like below:

```
@Injectable()
export class SideBarService {
  public sideBarComp;

}

@Component({
  selector: './app-side-bar',
  templateUrl: './side-bar.component.html',
  styleUrls: ['./side-bar.component.css']
})
export class SideBarComponent {

  @HostBinding('class.is-open')
  isOpen = false;

  constructor(
    private sideBarService: SideBarService
  ) {}

  ngOnInit() {
    this.sideBarService.sideBarComp = this;
  }

  toggle() {
    // out logic
  }
}
```

```
}
```

```
@Component({
  selector: './app-side-bar-toggle',
  templateUrl: './side-bar-toggle.component.html',
  styleUrls: ['./side-bar-toggle.component.css']
})
export class SideBarToggleComponent {

  constructor(
    private sideBarService: SideBarService
  ) {}
```

```
@HostListener('click')
click() {
  this.sideBarService.sideBarComp.toggle();
}
```



1 response



TheBjarni

Apr 13, 2018 · 1 min read

I've created another way of communication between components in Angular. I use one service in whole application, based on rxjs(event.service.ts):

```
@Injectable()
export class EventService {

  private eventsNames: string[] = [
    'toggleLeftPanel', 'toggleRightPanel', 'clear'
  ];

  private registeredEvents: { [name: string]: { source: Subject<Event> } } = {};

  constructor() {
    this.eventsNames.forEach(name => this.registeredEvents[name] =
      { source: new Subject<Event>() });
  }

  public pushEvent(event: Event) {
    try {
      this.registeredEvents[event.name].source.next(event);
    } catch (e) {
      console.warn(e);
    }
  }

  public addEventHandler(name: string, handler: (event: Event) =>
void) {
  this.registeredEvents[name].source.asObservable().subscribe(handler
  );
}
```

```
''
```

Class Event.ts:

```
export class Event {
  public data?: { [key: string]: any; } = {};
  constructor(public name: string) {}
}
```

It allows me to:

- send messages to any component with any additional data
- receive messages from any component
- communicate between components in different modules

For example, component ‘top-toolbar’ can send event ‘clear’ and both components ‘left-panel’ and ‘right-panel’ handle this event, but they can do different work after receive this event. Usage:

top-toolbar.component.ts:

```
export class TopToolbar implements OnInit {
(...)

constructor(
  private eventService: EventService
) {}

public clearOptionChosen() {
  this.eventService.pushEvent(new Event('clear'));
}
```

left-panel.component.ts:

```
export class LeftPanel implements OnInit {
(...)

constructor(
  private eventService: EventService
) {}

public ngOnInit() {
  this.eventService.addEventHandler('clear', event =>
    alert('Nothing to clear')
);
}
```

right-panel.component.ts:

```
export class RightPanel implements OnInit {
(...)

constructor(
  private eventService: EventService
) {}

public ngOnInit() {
  this.eventService.addEventHandler('clear', event =>
    // I can pass any data with event and use it here

    if (event.data['changeBackground'] != null) {
      // do something
    }
    clearPanel();

  );
}
}
```



53



Javier Font

May 19, 2018

I've translated this article to Spanish if anyone is interested

<https://medium.com/@javifont/3-formas-de-comunicarse-entre-componentes-angular-b349ada25797>



3



Wilder Perozo

Mar 14, 2018

Thanks for the post, I have problem with this in my project, I work with BehaviorSubject but affecting performance of my application, it doed the work but is not ideal, I will try with the 3rd way



11



mohammed haneefa

Mar 20, 2018

thanks for the post helped a lot



10



Jette Thrane

Jun 26, 2018

Excellent... helped me create a service to open and close ng-sidebar from another component than the one containing it. Saved me a bunch of time :-)



 **Alfaz Jikani**
Jul 2, 2018

Thanks a lot Sir:)



 **David Rehle**
Feb 22, 2019

Thanks for this article.

3. Communication through service

→why do we need @Output ? It is working with that annotation too.
(we are using angular v7)



 **Mark Onyango**
Jan 9, 2019 · 1 min read

Nice article!

Since this is for beginners, I would make it more apparent where the service has to be provided because it is not obvious from the article alone. Only when checking the app.module.ts on GitHub do you see it in the providers array. You could add it to the app.component.ts @component decorator as well, to make it...

[Read more...](#)



 **Ganesh Murthy**
May 26, 2018

Thanks for the post! good one. Good job!.



 **congdongdientu.com**
Oct 23, 2018

one of the best things on the web

...



1



v



rajan chauhan

Jun 8, 2018

Is it good practice to name classes differently from their respective file names?



v



imran ali

Dec 11, 2018

Good explanation...

Suppose there are two individual components which doesn't have relationship in between them, Then how we can send data to another component from one component?

Thanks in advance



v



Ashok M A

Feb 28, 2019

developers struggling with this will be googling it this way

You got me... ^_^\n



v



Sam Lefebvre

Mar 11, 2019

It looks like example 2 is incomplete? I do not see the relations between code parts. At least the filenames are wrong.



v



Kyle Henry

Apr 10, 2019

Wonderful. I can't thank you enough! You have really helped me discover the utility of angular



v



Cristian Macarie

Mar 6, 2019 · 1 min read

Another way of communicating between components is through Subjects:

- 1—Create a service containing a private Subject<?>
- 2—Provide a “sink” method (call subject.next(obj);) obj given as param.
- 3—Provide a “stream” method (return the subject as Observable)
- 4—Subscribe to the “stream” method in any component and push data through your “sink” method anywhere.
- 5—Enjoy!



Abhishek kandari

Aug 22, 2019

Thanks a lot Miro, this data sharing with service really saved me my time.

I had to show cart item count on header and had to update it when user added or removed item from any page.



Thilak Raj

May 21, 2018

HI,

I was trying communication through service. From service I am emitting the event, But it is not get triggered on component. I have posted the question in stack overflow. Could you please help me on this.

<https://stackoverflow.com/questions/50408989/how-to-call-sibling-component-method-in-angular-4>



Olajide Amoran (edijolaedoc)

May 13, 2019

Thank you. Will definitely be using these tips on my project.





Kaushik Das
Mar 13, 2018 · 1 min read

Hi Miro,

Thank you for the excellent article.. helped me a lot to understand the ways we can communicate with different components.

I have a basic query.. why the component selector starts with './'.

for e.g:

Read more...



1 response



Santosh Kumar
Oct 12, 2018

Hi,

very nice tutorial and useful, I am trying to do in Angular 6, there is no error but an event is not working can u upgrade your code with ng 6.



Duy Nguyễn
May 9, 2019

Hello [Miro Koczka](#),

It's useful for me!

By the way, would you mind telling me which software did you use to make the mockup? :)

Thank you.



1 response



Matthew Marberry
Jan 31, 2019 · 1 min read

Hey great article... I was wondering if you might have some time to help me with an Angular issue. I'm a React Dev—diving into and learning angular.

I'm having an issue with my project where I am trying to get the navbar to render different items based on whether or not the user is logged in.

Read more...





Leyla Salim

Jan 19, 2019

Great article man! this will surely help me think about that communication a different way.

;)



v



amine zawix

May 22, 2018 · 1 min read

Hi nice article ;

But i want to ask ,what to use to get data from a component in this case

```
<ndc-dynamic [ndcDynamicComponent] = "item.widget">
```

```
[ndcDynamicInputs] = "{StoredData:item.OldData}" </></ndc-dynamic>
```

[Read more...](#)



v



Arnav Simer

Oct 2, 2018

```
<app-side-bar #sideBar *ngIf = "1==1"></app-side-bar>
```

why reference is passing as undefined in case of *ngIf



v