

Angular 5 / 6 : interaction entre composant

Achref El Mouelhi

Docteur de l'université d'Aix-Marseille
Chercheur en Programmation par contrainte (IA)
Ingénieur en Génie logiciel

`elmouelhi.achref@gmail.com`

Plan

- 1 Introduction
- 2 Exemple avec `@Input ()`
- 3 Exemple avec `@Output ()`

Angular

Première forme d'interaction

- Une application Angular est composée de plusieurs composants
- En utilisant des formulaires et des liens, on peut envoyer des données d'un composant à un autre

Angular

Première forme d'interaction

- Une application Angular est composée de plusieurs composants
- En utilisant des formulaires et des liens, on peut envoyer des données d'un composant à un autre

Autre forme d'interaction

- On peut ajouter le sélecteur d'un premier composant dans la template d'un deuxième composant
 - on appelle le premier composant : composant fils
 - on appelle le deuxième composant : composant parent
- En utilisant les décorateurs `@Input ()` et `@Output ()` les deux composants peuvent échanger de données

Angular

Les décorateurs @Input () et @Output ()

- @Input () : permet à un composant fils de récupérer des données de son composant parent
- @Output () : permet à un composant parent de récupérer des données de son composant enfant

Angular

Avant de commencer

- Considérons deux composants `personne` et `fils`
- Ajoutons le sélecteur du composant `fils` `app-fils` dans `personne.component.html`
- Définissons une route `/personne` pour le composant parent

Angular

Avant de commencer

- Considérons deux composants `personne` et `fils`
- Ajoutons le sélecteur du composant `fils` `app-fils` dans `personne.component.html`
- Définissons une route `/personne` pour le composant parent

Dans cet exemple

- Chaque élément `fils` aura deux attributs `nom` et `villeNaissance` affichés dans la template
- Le composant `personne` attribuera les valeurs aux deux attributs de la classe `fils.component.ts`

Angular

Le fichier `files.component.ts`

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

@Component({
  selector: 'app-fils',
  templateUrl: './files.component.html',
  styleUrls: ['./files.component.css']
})
export class FilsComponent implements OnInit {
  @Input() ordre: string;
  @Input() villeNaissance: string;

  constructor() { }

  ngOnInit() { }
}
```

Le fichier `files.component.html`

```
<li>
  <h5> Je suis le {{ ordre }} fils et suis de {{ villeNaissance }} </h5>
</li>
```


Le fichier `personne.component.ts`

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

@Component({
  selector: 'app-personne',
  templateUrl: './personne.component.html',
  styleUrls: ['./personne.component.css']
})
export class PersonneComponent implements OnInit {
  tab:Array<string> = ["premier","deuxieme","troisieme"];
  nord: string = "Lille";
  sud: string = "Marseille";
  capitale: string = "Paris";
  constructor() { }

  ngOnInit() { }
}
```

Le fichier `personne.component.html`

```
<ul>
  <app-fils [ordre]="tab[0]" [villeNaissance]="sud" ></app-fils>
  <app-fils [ordre]="tab[1]" [villeNaissance]="nord" ></app-fils>
  <app-fils [ordre]="tab[2]" [villeNaissance]="capitale" ></app-fils>
</ul>
```

Angular

Dans cet exemple

- Chaque élément `fil`s aura un champ texte pour saisir une note et un bouton pour envoyer la valeur au parent
- Le bouton sera désactivé après envoi
- Chaque fois que le père reçoit une note d'un de ses fils, il recalcule la moyenne et il l'affiche

Angular

Dans cet exemple

- Chaque élément `fil`s aura un champ texte pour saisir une note et un bouton pour envoyer la valeur au parent
- Le bouton sera désactivé après envoi
- Chaque fois que le père reçoit une note d'un de ses fils, il recalcule la moyenne et il l'affiche

Le fichier `fil.component.html`

```
<h6>{{ nom }} </h6>
<input type=number name=note [(ngModel)]=note />
<button (click)="send()" [disabled]=buttonStatus >
  Send
</button>
```

Angular

Le fichier `files.component.ts`

```
import { Component, OnInit, Input, Output, EventEmitter } from '@angular/core';

@Component({
  selector: 'app-files',
  templateUrl: './files.component.html',
  styleUrls: ['./files.component.css']
})
export class FilesComponent implements OnInit {
  @Input() nom: string;
  @Output() message = new EventEmitter<number>();
  note: number;
  buttonStatus: boolean = false;

  send() {
    this.message.emit(this.note);
    this.buttonStatus=true;
  }
  constructor() { }

  ngOnInit() { }
}
```

Angular

Le fichier `personne.component.ts`

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

@Component({
  selector: 'app-personne',
  templateUrl: './personne.component.html',
  styleUrls: ['./personne.component.css']
})
export class PersonneComponent implements OnInit {
  moyenne : number = 0;
  somme : number =0;
  nbr:number =0;
  enfants = ['Wick', 'Hoffman', 'Abruzzi'];

  constructor() { }
  ngOnInit() { }
  computeAvg(note: number) {
    this.somme += note;
    this.nbr++;
    this.moyenne= this.somme/this.nbr;
  }
}
```

Angular

Le fichier `personne.component.html`

```
<h3> Moyenne de mes enfants {{ moyenne }} </h3>
<app-fils *ngFor="let _enfant_of_enfants" [nom]="
  enfant" (message)="computeAvg($event)">
</app-fils>
```