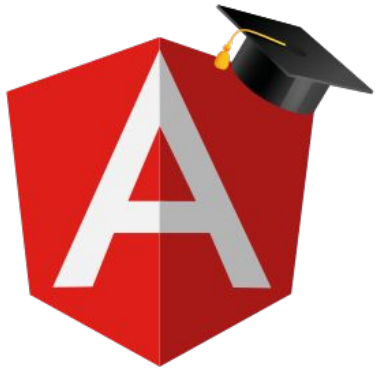
V-Services



LES SERVICES

Un service permet de centraliser des parties de votre code et des données qui sont utilisées par plusieurs parties de votre application ou de manière globale par l'application entière.

Les services permettent donc :

- de ne pas avoir le même code doublé ou triplé à différents niveaux de l'application - ça facilite donc la maintenance, la lisibilité et la stabilité du code ;

- de ne pas copier inutilement des données - si tout est centralisé, chaque partie de l'application aura accès aux mêmes informations, évitant beaucoup d'erreurs potentielles.

I - Injection et instances

Pour être utilisé dans l'application, un service doit être injecté, et le niveau choisi pour l'injection est très important.

Il y a trois niveaux possibles pour cette injection :

- dans AppModule (src/app/app.module.ts) : ainsi, la même instance du service sera utilisée par tous les composants de l'application et par les autres services ;
- dans AppComponent (src/app/app.component.ts) : tous les components auront accès à la même instance du service mais non les autres services ;
- dans un autre composant : le composant lui-même et tous ses enfants (c'est-à-dire tous les components qu'il englobe) auront accès à la même instance du service, mais le reste de l'application n'y aura pas accès.

Nous allons créer notre premier service: Utilisez la commande `ng generate service data`
OU créez un sous-dossier `service` dans app , et créez-y un nouveau fichier appelé `data.service.ts`

Écrivez y le code suivant:

```
export class DataService {

}
```

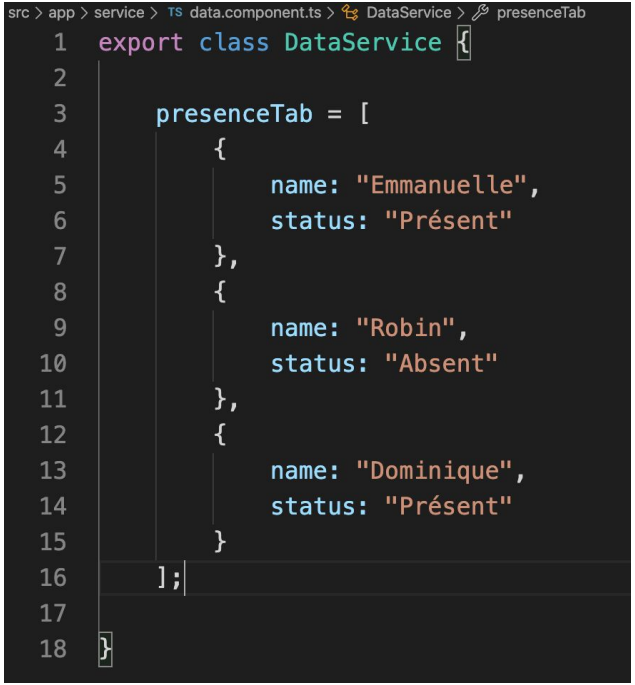
A présent, nous allons injecter ce service dans AppModule (src/app/app.module.ts) en l'ajoutant à l'array providers et en n'oubliez pas d'ajouter l'import correspondant en haut du fichier :

```ts
import { BrowserModule } from "@angular/platform-browser";
import { NgModule } from "@angular/core";

import { AppRoutingModule } from "./app-routing.module";
import { AppComponent } from "./app.component";
import { MonPremierComponentComponent } from "./mon-premier-component/mon-premier-component.component";
import { FormsModule } from "@angular/forms";
import { DataService } from "./service/data.service";

@NgModule({
  declarations: [
    AppComponent,
    MonPremierComponentComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule
  ],
  providers: [
    DataService
  ],
  bootstrap: [
    AppComponent
  ],
})
export class AppModule {}
```

Angular crée maintenant une instance du service DataService pour l'application entière.
Pour intégrer ce service dans un component, on le déclare comme argument dans son constructeur.

II - Utilisez les services

Le premier élément qu'il serait logique de déporter dans le service serait l'array presenceTab de app-component.ts.
Copiez-le depuis AppComponent , collez-le dans DataService



Il faut maintenant que AppComponent puisse récupérer les informations stockées dans DataService . Pour cela, vous allez implémenter la méthode ngOnInit() .

Remarque: ngOnInit() correspond à une "lifecycle hook". Le détail de ces hooks va au-delà du cadre de ce cours, mais pour l'instant, tout ce que vous avez besoin de savoir, c'est que la méthode ngOnInit() d'un component est exécutée une fois par instance au moment de la création du composant par Angular, et après son constructeur. On l'utilise très souvent pour initialiser des données une fois le component créé.

Créer la fonction ngOnInit() qu'on placera après le constructeur et avant les autres méthodes du component :

```
22  constructor(private dataService: DataService) {
23
24  }
25
26  ngOnInit() {
27
28  }
```

Ensuite, dans la déclaration de classe AppComponent , vous allez implémenter l'interface OnInit (en l'important depuis @angular/core en haut) et importer le service :

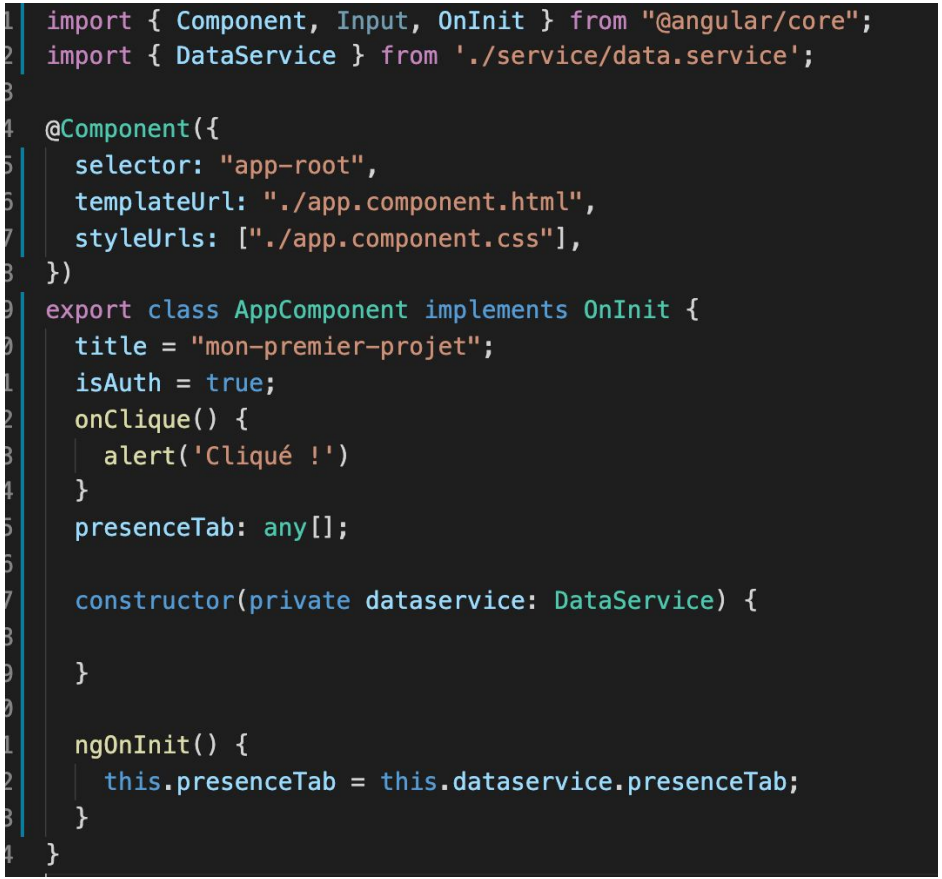
```
import { Component, Input, OnInit } from "@angular/core";
import { DataService } from './service/data.service';

@Component({
  selector: "app-root",
  templateUrl: "./app.component.html",
  styleUrls: ["./app.component.css"],
})
export class AppComponent implements OnInit {
```

Vous pouvez maintenant récupérer les informations depuis DataService dans la méthode ngOnInit() :

```
26    ngOnInit() {
27      this.presenceTab = this.dataService.presenceTab;
28    }
29
```

Votre application devrait fonctionner à nouveau, avec la liste des personnes ainsi que leur statut qui s'affiche comme avant.
Remarque: Il n'y a aucune différence visuelle, mais votre code est maintenant plus modulaire, et ce sera plus facile d'ajouter des fonctionnalités.

app-component.ts devrait ressembler à ceci



```ts
import { Component, Input, OnInit } from "@angular/core";
import { DataService } from './service/data.service';

@Component({
  selector: "app-root",
  templateUrl: "./app.component.html",
  styleUrls: ["./app.component.css"],
})
export class AppComponent implements OnInit {
  title = "mon-premier-projet";
  isAuth = true;
  onClique() {
    alert('Cliqué !')
  }
  presenceTab: any[];

  constructor(private dataservice: DataService) {

  }

  ngOnInit() {
    this.presenceTab = this.dataservice.presenceTab;
  }
}
```