x.x

Cours Angular

| TD CRUD

| HTTPClient Service

x] Réalisations

1) Création de l'application Angular :

```
ng new users-app
```

2) Utilisation de Bootstrap :

```
npm install --save bootstrap
```

Dans `angular.json` :

```
"node_modules/bootstrap/dist/css/bootstrap.min.css",
```

3) Création d'un back-end JSon Server :

```
npm install -g json-server
```

4) Création d'un dossier `backend` à la racine de l'application avec un fichier `database.json` :

```
{
    "users": [{
        "id": 1,
        "name": "Alex Terieur",
        "email": "alex.terieur@example.com",
        "phone": "0000000000"
    }, {
        "id": 2,
        "name": "Alain Terieur",
        "email": "alain.terieur@example.com",
        "phone": "0000000000"
    }, {
        "id": 3,
        "name": "Agathe Zeblouze",
        "email": "agathe.zeblouze@example.com",
        "phone": "0000000000"
    }, {
        "id": 4,
        "name": "Yvan Samob",
        "email": "yvan.samob@example.com",
        "phone": "0000000000"
    }]
}
```

5) Démarrage du serveur type NodeJS :

```
json-server --watch backend/database.json
```

6) Vérification dans Chrome :

URL : http://localhost:3000/users

7) Importation du module HTTP dans `app.module.ts` :

```
import { HttpClientModule } from '@angular/common/http';
```

Et dans la section `imports` :

```
HttpClientModule
```

8) Création du service crud :

```
ng generate service crud
```

9) Modification de `crud.service.ts` :

```
import { Injectable } from '@angular/core';
import { catchError } from 'rxjs/operators';
import { HttpClient, HttpHeaders, HttpErrorResponse } from '@angular/common/http';
import { Observable, throwError } from 'rxjs';

export interface User {
  id: string;
  name: string;
  email: string;
  phone: string;
}

const urlrest = 'http://localhost:3000';

@Injectable({
  providedIn: 'root'
})

export class CrudService {

  constructor(private http: HttpClient) { }

  httpHeader = {
    headers: new HttpHeaders({
      'Content-Type': 'application/json'
    })
  }

  getUsers(): Observable<any> {
    return this.http.get<User>(urlrest + '/users').pipe(
      catchError(this.handleError)
    );
  }

  getUser(id: string): Observable<any> {
    return this.http.get<User>(urlrest + '/users' + id).pipe(
      catchError(this.handleError)
    );
  }
  private handleError(error: HttpErrorResponse): any {
    if (error.error instanceof ErrorEvent) {
      console.error('An error occurred:', error.error.message);
    } else {
      console.error(
        `Backend returned code ${error.status}, ` +
        `body was: ${error.error}`);
    }

    return throwError(() => 'Something bad happened; please try again later.');
  }
}
```

10) Création du template dans `app.component.html` :

```
<table class="table table-bordered">
  <thead>
    <tr>
      <th scope="col">#Id</th>
      <th scope="col">Name</th>
      <th scope="col">Email</th>
      <th scope="col">Phone</th>
      <th scope="col">Action</th>
    </tr>
  </thead>
  <tbody>
    <tr *ngFor="let user of Users">
      <td>{{user.id}}</td>
      <td>{{user.name}}</td>
      <td>{{user.email}}</td>
      <td>{{user.phone}}</td>
      <td>

      </td>
    </tr>
  </tbody>
</table>
```

11) Modification du fichier `app.component.ts` :

```
import { Component, OnInit } from '@angular/core';
import { CrudService } from "./crud.service";

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})

export class AppComponent implements OnInit {

  Users: any = [];

  constructor(public crudService: CrudService) { }

  ngOnInit() {
    this.fetchUsers()
  }

  fetchUsers() {
    return this.crudService.getUsers().subscribe((data: {}) => {
      this.Users = data;
    })
  }
}
```

12) Lancement de l'application Angular…

x] Suppléments

1) Ajout de la suppression :

Dans le fichier du template (`td` vide) :

```
<span class="text-primary" (click)="remove(user.id)">Delete</span>
```

Dans le fichier `ts` du composant :

```
  remove(id: string) {
    this.crudService.delete(id).subscribe(res => {
      this.fetchUsers()
    })
  }
```

Dans le fichier du service :

```
  delete(id: string): Observable<any> {
    return this.http.delete<User>(urlrest + '/users/' + id).pipe(
      catchError(this.handleError)
    )
  }
```