Développement Mobile IONIC

Séquence 4

Ionic Storage

- Ionic **Offline Storage** ou **localStorage** est un système de stockage de données multiplateforme qui fonctionne sur iOS et Android. Il facilite l'ajout de stockage hors ligne aux applications Ionic qui est sécurisé (chiffré sur l'appareil), très performant et fournit une interrogation avancée des données NoSQL.
- Le **localStorage** est une librairie utilisé sur les applications Angular pour leur permettre de stocker des informations en local. Elle fonctionne sur le principe de **clé – valeur** et est très utilisé pour faire des sauvegardes.
- Le **localStorage** est une alternative gratuite et open source pour les développeurs indépendants, les étudiants et les amateurs. Il fournit un moyen simple de stocker des paires clé / valeur et des objets JSON.

Ionic Storage

- Le **localStorage** est une alternative gratuite et open source pour les développeurs indépendants, les étudiants et les amateurs. Il fournit un moyen simple de stocker des paires clé / valeur et des objets JSON.
- Il utilise une variété de moteurs de stockage en dessous, en choisissant le meilleur disponible en fonction de la plate-forme:
  - Lorsqu'il s'exécute dans un contexte d'application native, le stockage priorise l'utilisation de SQLite, car il s'agit de l'une des bases de données basées sur des fichiers les plus stables et les plus utilisées, et évite certains des écueils tels que localstorage et IndexedDB, tels que le système d'exploitation décidant de supprimer ces données dans des situations de faible espace disque.
  - Lors de l'exécution sur le Web ou en tant qu'application Web progressive, le stockage tentera d'utiliser IndexedDB, WebSQL et localstorage, dans cet ordre.

Utilisation

- **SQLite :** Tout d’abord il faut installer le plugin

```
ionic cordova plugin add cordova-sqlite-storage
```

- Ensuite installation du package

```
npm install --save @ionic/storage
```

- Ajouter à la liste des imports module déclaré
  - ***src/app/app.module.ts***

Déclaration du module **Storage**

```
import { IonicStorageModule } from '@ionic/storage';

@NgModule({
  declarations: [
    // ...
  ],
  imports: [
    BrowserModule,
    IonicModule.forRoot(MyApp),
    IonicStorageModule.forRoot()
  ],
  bootstrap: [IonicApp],
  entryComponents: [
    // ...
  ],
  providers: [
    // ...
  ]
})
export class AppModule {}
```

Injection du storage dans le component ou la page

```
import { Storage } from '@ionic/storage';

export class MyApp {
  constructor(private storage: Storage) { }

  ...

  // set a key/value
  storage.set('name', 'Max');

  // Or to get a key/value pair
  storage.get('age').then((val) => {
    console.log('Your age is', val);
  });
}
```

Configuration Strorage

- Le moteur de stockage peut être configuré à la fois avec des priorités de moteur de stockage spécifiques ou des options de configuration personnalisées à transmettre à localForage.

```
import { IonicStorageModule } from '@ionic/storage';

@NgModule({
  declarations: [...],
  imports: [
    IonicStorageModule.forRoot({
      name: '__mydb',
driverOrder: ['indexeddb', 'sqlite', 'websql']
    })
  ],
  bootstrap: [...],
  entryComponents: [...],
   providers: [...]
})
export class AppModule { }
```

TP Storage avec Les opérations CRUD

Creation Projet

- Crééer le projet: `>ionic start crudOperation blank`
- Lancer la commande si dessus et choisir Angular. Se pointer au répertoire du projet par la commande `cd crudOperation`
- Ensuite Taper les commandes suivantes pour installer les librairies ***Ionic*** et ***cordova*** du plugin ***localStorage***

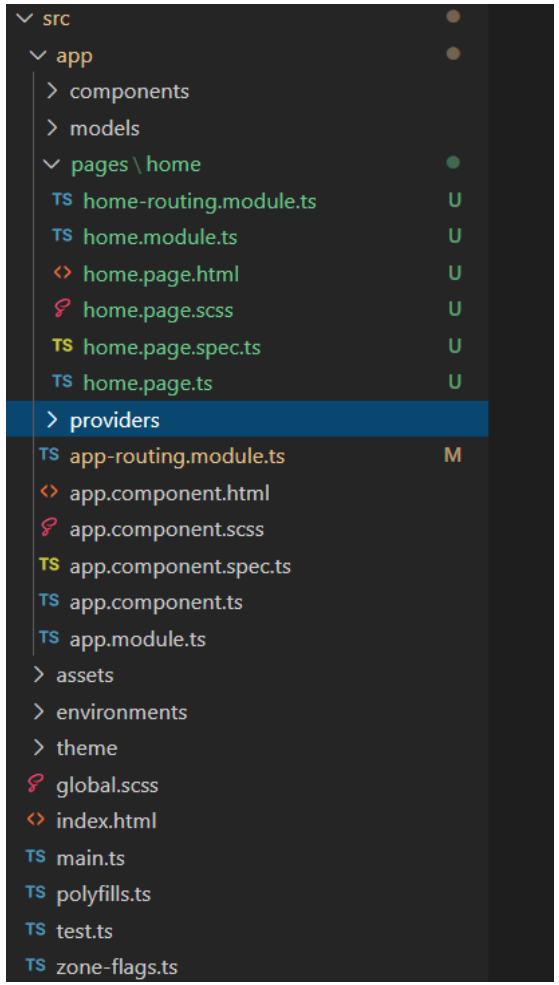
```
ionic cordova plugin add cordova-sqlite-storage
```

```
>npm install --save @ionic/storage
```

Structure du projet

Créeer les dossiers suivants dans app:

- Components, models, pages, providers.
- Puis déplacer le dossier ***home*** dans le dossier ***pages***



Changement adresse point d'entrée

```
import { NgModule } from '@angular/core';      You, an hour ago • Initial commit
import { PreloadAllModules, RouterModule, Routes } from '@angular/router';

const routes: Routes = [
  {
    path: 'home',
    loadChildren: () => import('./pages/home/home.module').then( m => m.HomePageModule)
  },
  {
    path: '',
    redirectTo: 'home',
    pathMatch: 'full'
  },
];

You, an hour ago | 1 author (You)
@NgModule({
  imports: [
    RouterModule.forRoot(routes, { preloadingStrategy: PreloadAllModules })
  ],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

Configuration du fichier *src/app/app-module.ts*

- **__mydb** représente le nom de notre base de données;

- **‘indexeddb’, ‘sqlite’, ‘websql**: représentent les différents type de SGBD sur lesquelles notre application pourra sauvegarder ses données.

```
import { AppComponent } from './app.component';
import { AppRoutingModule } from './app-routing.module';
import { FormModule } from './models/form/form.module';
import {IonicStorageModule} from '@ionic/storage';

You, a few seconds ago | 1 author (You)
@NgModule({
  declarations: [AppComponent],
  entryComponents: [],
  imports: [BrowserModule, IonicModule.forRoot(),
    IonicStorageModule.forRoot({
      name: '__mydb',        You, a few seconds ago • Uncommitted changes
      driverOrder: ['indexeddb', 'sqlite', 'websql']
    }),
    AppRoutingModule,
    FormModule
  ],
  providers: [
    StatusBar,
    SplashScreen,
    { provide: RouteReuseStrategy, useClass: IonicRouteStrategy }
  ],
  bootstrap: [AppComponent]
})
export class AppModule {}
```

Création d'un Provider

- Taper la commande :

```
>ionic g service providers/dataProvider
```

```
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root'
})
export class DataProviderService {

  constructor() { }
}
```

loadData() qui cette fonction là qui nous permettra de lire les données dans notre base de donnée

updateData() qui aura pour objectif de mettre à jour les informations dans notre base de données

```
import {Injectable} from '@angular/core';
import {Storage} from '@ionic/storage';

@Injectable({
    providedIn: 'root'
})
export class DataProviderService {
    data = [];
    constructor(private storage: Storage) {
    }
    loadData() {
        return this.storage.get('data').then(value => {
            if (value) {
                this.data = JSON.parse(value);
            }
        });
    }
    updateData() {
        this.storage.set('data', JSON.stringify(this.data));
    }
}
```

src\app\app.component.ts

```
import { Component } from '@angular/core';
import { Platform } from '@ionic/angular';
import { SplashScreen } from '@ionic-native/splash-screen/ngx';
import { StatusBar } from '@ionic-native/status-bar/ngx';
import {DataProviderService} from './providers/data-provider.service';
You, a few seconds ago | 1 author (You)
@Component({
  selector: 'app-root',          You, 2 hours ago • Initial commit
  templateUrl: 'app.component.html',
  styleUrls: ['app.component.scss']
})
export class AppComponent {
  constructor(
    private platform: Platform,
    private splashScreen: SplashScreen,
    private statusBar: StatusBar,
    private dataProviderService: DataProviderService
  ) {
    this.initializeApp();
    dataProviderService.loadData();
  }

  initializeApp() {
    this.platform.ready().then(() => {
      this.statusBar.styleDefault();
      this.splashScreen.hide();
    });
  }
}
```

src\app\app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { RouteReuseStrategy } from '@angular/router';
import { IonicModule, IonicRouteStrategy } from '@ionic/angular';
import { SplashScreen } from '@ionic-native/splash-screen/ngx';
import { StatusBar } from '@ionic-native/status-bar/ngx';
import { AppComponent } from './app.component';
import { AppRoutingModule } from './app-routing.module';
import { FormModule } from './modals/form/form.module';
import {IonicStorageModule} from '@ionic/storage';       You, a few seconds
You, a few seconds ago | 1 author (You)
@NgModule({
  declarations: [AppComponent],
  entryComponents: [],
  imports: [BrowserModule, IonicModule.forRoot(),
    IonicStorageModule.forRoot({
      name: '__mydb',
      driverOrder: ['indexeddb', 'sqlite', 'websql']
    }),
    AppRoutingModule,
    FormModule
  ],
  providers: [
    StatusBar,
    SplashScreen,
    { provide: RouteReuseStrategy, useClass: IonicRouteStrategy }
  ],
  bootstrap: [AppComponent]
})
export class AppModule {}
```

src/app/pages/home/home.page.ts

```ts
import {Component} from '@angular/core';
import {AlertController, ModalController, ToastController} from '@ionic/angular';
import {DataProviderService} from '../../providers/data-provider.service';
import {FormModal} from '../../modals/form/form.modal';
@Component({
    selector: 'app-home',
    templateUrl: 'home.page.html',
    styleUrls: ['home.page.scss'],
})
export class HomePage {

    constructor(private modalController: ModalController,
                public alertController: AlertController,
                public toastController: ToastController,
                public dataProviderService: DataProviderService) {
    }

    async openModal(data = null) {
      const modal = await this.modalController.create({
          component: FormModal,
          componentProps: {data: data},
          animated: true
      });
      return await modal.present();
  }
```

```ts
  async deleteItem(index) {
    const alert = await this.alertController.create({
        header: 'Supprimer!',
        message: 'Voullez vous vraiment supprimer?',
        buttons: [
            { text: 'Annuler', role: 'cancel',  cssClass: 'secondary', handler: (blah) => {
                }
            }, {
                text: 'Supprimer',
                handler: () => {
                    this.dataProviderService.data.splice(index, 1);
                    this.dataProviderService.updateData();
                    this.deleteMessage();
                }
            }
        ]
    });
    await alert.present();
}

async deleteMessage() {
    const toast = await this.toastController.create({
        message: 'Item correctement supprimé',
        duration: 2000
    });
    toast.present();
}
```

src/app/pages/home/home.page.html

```
<ion-header>
  <ion-toolbar color="primary">
    <ion-title>
      CRUD Etudiant
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <div >
    <ion-list *ngIf="dataProviderService.data.length>0">
      <ion-list-header>
        Liste de etudiants
      </ion-list-header>
      <ion-item-sliding
                        *ngFor="let d of dataProviderService.data; let i=index ">
        <ion-item-options side="start">
          <ion-item-option (click)="deleteItem(i)" color="danger" expandable>
            Supprimer
          </ion-item-option>
        </ion-item-options>
```

```
          <ion-item (click)="openModal({data:d,index:i})">
            <div>
              <div class="title">{{d.nom}}</div>
              <div class="title">{{d.prenom}}</div>
              <div class="title">{{d.filiere}}</div>
              <div class="title">{{d.genre}}</div>

            </div>
          </ion-item>
        </ion-item-sliding>
      </ion-list>

      <div *ngIf="dataProviderService.data.length<1" class="empty-item">
        <span> Aucun etudiant n'est disponible</span>
      </div>
    </div>
    <ion-fab vertical="bottom" horizontal="end" slot="fixed">
      <ion-fab-button (click)="openModal()">
        <ion-icon name="add"></ion-icon>
      </ion-fab-button>
    </ion-fab>
</ion-content>
```

src/app/pages/home/home.page.scss

```scss
ion-content {
  .empty-item {
    text-align: center;
    margin-top: 65%;
  }

  ion-list {
    ion-item {
      .title {
        font-size: 15px;
        font-weight: bold;
      }

      .desc {
        margin-top: 10px;
        margin-bottom: 10px;
        font-style: italic;
        font-size: 13px;
      }
    }
  }
}
```

src/app/pages/home/home.module.ts

```ts
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { IonicModule } from '@ionic/angular';
import { FormsModule } from '@angular/forms';
import { HomePage } from './home.page';

import { HomePageRoutingModule } from './home-routing.module';


@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    HomePageRoutingModule
  ],
  declarations: [HomePage]
})
export class HomePageModule {}
```

src/app/modals/form/form.page.html

```
<ion-header>
    <ion-toolbar color="primary">
        <ion-buttons slot="start">
            <ion-button (click)="closeModal()">
                <ion-icon class="close-icon" name="close"></ion-icon>
            </ion-button>
        </ion-buttons>
        <ion-title>{{itemData.index != -1 ? 'Modifier' : 'Ajouter'}} un etudiant</ion-title>
    </ion-toolbar>
</ion-header>

<ion-content>
    <div padding>
        <form [formGroup]="onForm" class="list-form">
            <ion-item>
                <ion-label position="floating" color="primary">Nom</ion-label>
                <ion-input type="text" formControlName="nom" placeholder="Nom"></ion-input>
            </ion-item>
            <ion-item>
                <ion-label position="floating" color="primary">Prenom</ion-label>
                <ion-input placeholder="Prenom" formControlName="prenom"></ion-input>
            </ion-item>
            <ion-item>
                <ion-label position="floating" color="primary">Filiere</ion-label>
                <ion-input placeholder="filiere" formControlName="filiere"></ion-input>
            </ion-item>
```

```
            <ion-item>
                <ion-label>Genre</ion-label>
                <ion-select formControlName="genre" placeholder="Select One">
                    <ion-select-option value="f">Female</ion-select-option>
                    <ion-select-option value="m">Male</ion-select-option>
                </ion-select>
            </ion-item>
            <ion-button expand="block" (click)="saveData()" [disabled]="!onForm.valid" fill="outline" color="primary">
                Enregistrer
            </ion-button>
        </form>
    </div>
</ion-content>
```

src/app/modal/form/form.page.ts

```ts
import {Component, OnInit} from '@angular/core';
import {ModalController, NavParams, ToastController} from '@ionic/angular';
import {DataProviderService} from '../../providers/data-provider.service';
import {FormBuilder, FormControl, FormGroup, Validators} from '@angular/forms';

@Component({
    selector: 'app-form',
    templateUrl: 'form.modal.html',
    styleUrls: ['form.modal.scss'],
})
export class FormModal implements OnInit {
    onForm: FormGroup;
    itemData: any = {data: {nom: '', prenom: '', filiere: '', genre: ''}, index: -1};

    constructor(private modalController: ModalController,
                private formBuilder: FormBuilder,
                private navParams: NavParams,
                public toastController: ToastController,
                public dataProviderService: DataProviderService) {
        if (navParams.data.data) {
            this.itemData = navParams.data.data;
        }
    }
    async closeModal() {
        await this.modalController.dismiss('');
    }
```

```ts
    ngOnInit(): void {
        this.onForm = this.formBuilder.group({
            nom: new FormControl(this.itemData.data.title,
                Validators.required),
            prenom: new FormControl(this.itemData.data.prenom),
            filiere: new FormControl(this.itemData.data.filiere),
            genre: new FormControl(this.itemData.data.genre)

        });
    }

    saveData() {
        if (this.itemData.index !== -1) {
            // @ts-ignore
            this.dataProviderService.data[this.itemData.index] = this.onForm.value;
        } else {
            // @ts-ignore
            this.dataProviderService.data.push(this.onForm.value);
        }
        this.dataProviderService.updateData();
        this.updateMessage();
        this.closeModal();
    }

    async updateMessage() {
        const toast = await this.toastController.create({ message: 'Etudiant enregistré',  duration: 3000
        });
        toast.present();
    }
}
```

src/app/modal/form/form.module.ts

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { IonicModule } from '@ionic/angular';
import {FormsModule, ReactiveFormsModule} from '@angular/forms';
import { RouterModule } from '@angular/router';
import { FormModal } from './form.modal';

@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    ReactiveFormsModule,
    RouterModule.forChild([
      {
        path: '',
        component: FormModal
      }
    ])
  ],
  declarations: [FormModal]
})
export class  FormModule {}
```

src\app\providers\data-provider.service.ts

```ts
import {Injectable} from '@angular/core';
import {Storage} from '@ionic/storage';

@Injectable({
    providedIn: 'root'
})
export class DataProviderService {
    data = [];
    constructor(private storage: Storage) {
    }
    loadData() {
        return this.storage.get('data').then(value => {
            if (value) {
                this.data = JSON.parse(value);
            }
        });
    }
    updateData() {
        this.storage.set('data', JSON.stringify(this.data));
    }
}
```

CRUD Etudiant

Aucun etudiant n'est disponible

+

× Ajouter un etudiant

Nom
BADJI

Prenom
INSA

Filiere
IDA

Genre Male

ENREGISTRER

CRUD Etudiant

Liste de etudiants

BADJI
INSA
IDA
m

Thiam
Yacine
Anglais
f

CRUD Etudiant

Liste de etudiants

SUPPRIMER

BADJI
INSA
IDA
m

Thiam
Yacine
Anglais
f

CRUD IONIC with MySQL

- Création API :
  - Créer un fichier de connexion qui s'appelle connexionDB

```php
<?php

/**
 * @return mysqli
 */
function db_connexion()
{
    $servername = "localhost";
    $username = "root";
    $password = "";
    $db = "tpcruduvsida";
    return new mysqli($servername, $username, $password,$db,"3308");
}
```

CRUD IONIC with MySQL

- Créer un fichier de index.php

```php
<?php
header('Access-Control-Allow-Origin: *');
require 'connexionBD.php';
require 'model.php';
$model = new Model();

if (isset($_GET['action']) && $_GET['action'] == 'list') {
    $data = $model->index();

    echo json_encode(array('success' => true, 'data' => $data));
}
if (isset($_GET['action']) && $_GET['action'] == 'update') {
    $inputJSON = file_get_contents('php://input');
    $postData = json_decode($inputJSON, TRUE);

    $result = $model->update($postData);
    echo json_encode(array('success' => $result));
}
```

```php
if (isset($_GET['action']) && $_GET['action'] == 'create') {
    $inputJSON = file_get_contents('php://input');
    $postData = json_decode($inputJSON, TRUE);
    $result = $model->insert($postData);
    echo json_encode(array('success' => true, 'data' => $data));
}
if (isset($_GET['action']) && $_GET['action'] == 'detail') {
    $data = $model->get($_GET['id']);

    echo json_encode(array('success' => true, 'data' => $data));
}

if (isset($_GET['action']) && $_GET['action'] == 'delete') {
    $result = $model->delete($_GET['id']);

    echo json_encode(array('success' => $result));
}
```

CRUD IONIC with MySQL

- Créer un fichier de model.php

```php
<?php

class Model
{

    private $con;
    private $table = 'etudiant';

    /**
     * Model constructor.
     */
    public function __construct()
    {
        $this->con = db_connexion();
    }

    /**
     * @return array|bool
     */
    public function index()
    {
        $result = mysqli_query($this->con, 'SELECT * FROM ' . $this->table);
        $data = array();
        if ($result) {
            while ($row = mysqli_fetch_assoc($result)) {
                $data[] = $row;
            }
            return $data;
        }
        return null;
    }
```

```php
    /**
     * @param $data
     * @return array|bool
     */
    public function insert($data)
    {
        if ($data) {
            $query = 'INSERT INTO ' . $this->table . ' (nom,prenom, age, filiere)
            VALUES ("' . $data['nom'] . '", "' . $data['prenom'] . '" , "' . $data['age'] . '", "' . $data['filiere'] . '")';
            $res = mysqli_query($this->con, $query);
            if ($res) {
                return $this->get(mysqli_insert_id($this->con));
            }
            return false;
        }
    }

    /**
     * @param $data
     * @return bool
     */
    public function update($data)
    {
        if (isset($data)) {
            $query = 'UPDATE ' . $this->table . ' SET nom="' . $data['nom'] . '", prenom="' . $data['prenom'] . '",
            age="' . $data['age'] . '", filiere="' . $data['filiere'] . '" WHERE id=' . $data['id'];
            $res = mysqli_query($this->con, $query);
            if ($res)
                return true;
            return false;
        }
    }
```

CRUD IONIC with MySQL

- Créer un fichier de model.php

```php
/**
 * @param $id
 * @return array|null
 */
public function get($id)
{
    $result = mysqli_query($this->con, 'SELECT * FROM ' . $this->table . ' WHERE id=' . $id);
    if ($result) {
        return mysqli_fetch_assoc($result);
    }
    return null;

}

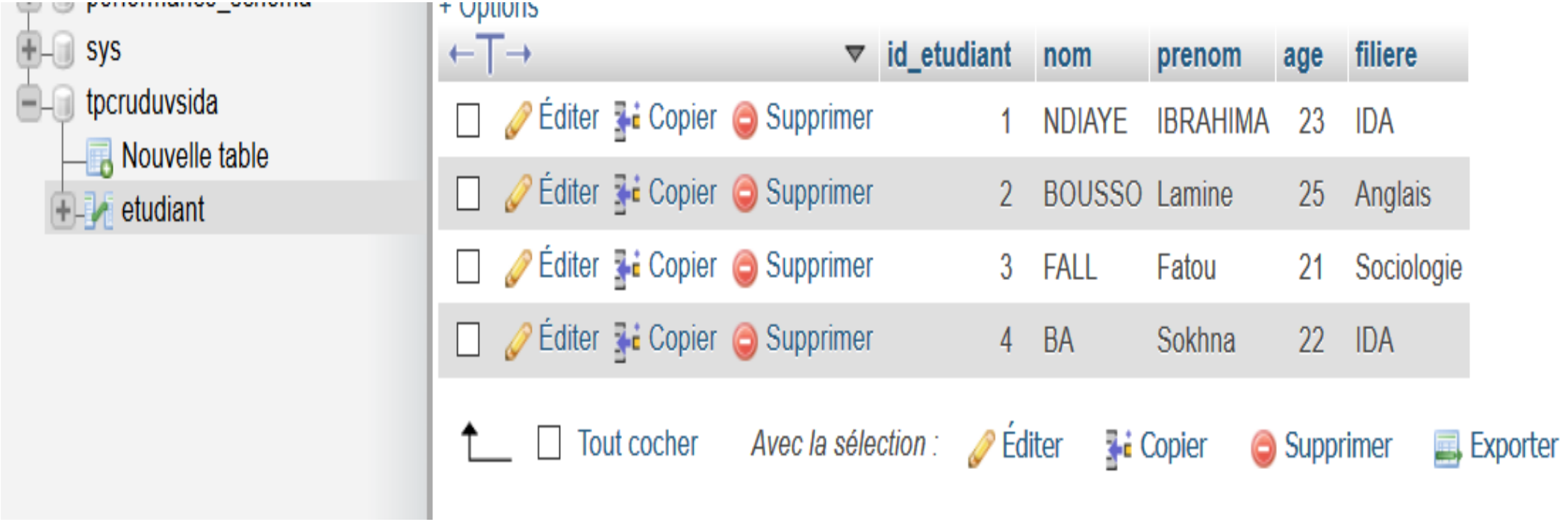
/**
 * @param $id
 * @return bool
 */
public function delete($id)
{
    $res = mysqli_query($this->con, 'DELETE FROM ' . $this->table . ' WHERE id=' . $id);
    return $res ? true : false;
}
```

Lien pour API

L'utilisation de notre api par notre application ***Ionic*** se fera grâce aux URLs suivantes :

- ***http://localhost/mon-api/?action=list*** : Elle permettra de retourner la liste de tout les éléments présent dans notre base de donnée food. Ceci depuis notre application mobile, elle utilise la méthode de transport de donnée **GET** ;
- ***http://localhost/mon-api/?action=create*** : Elle nous permettra d'enregistre un élément dans notre base de donnée. Elle utilisera le méthode de transport de donnée **POST** ;
- ***http://localhost/mon-api/?action=update***: Elle nous permettra mettre à jour un élément dans notre base de donnée. Elle utilisera le méthode de transport de donnée **POST** ;
- ***http://localhost/mon-api/?action=delete&id="id de élément"***: Suppression d'un élément , méthode **DELETE** ;
- ***http://localhost/mon-api/?action=detail&id="id de élément"***: Elle nous permettra d'avoir le détail sur un élément, méthode **GET**.

Table Etudiant

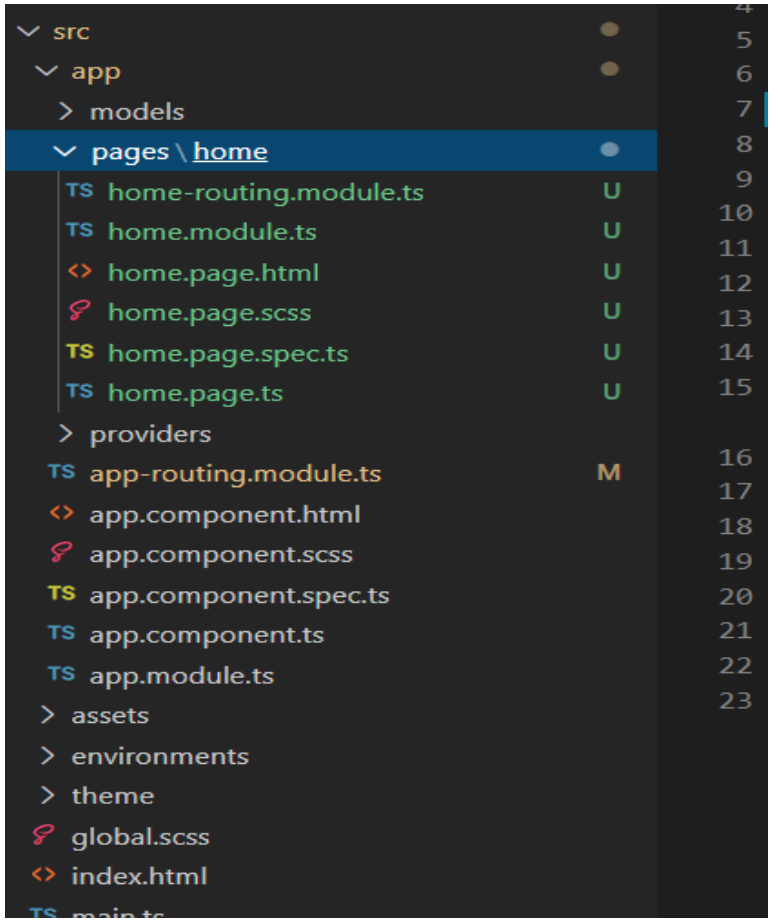


http://localhost/apiIonic/?action=list


```
{
  "success": true,
  "data": [
    {
      "id_etudiant": "1",
      "nom": "NDIAYE",
      "prenom": "IBRAHIMA",
      "age": "23",
      "filiere": "IDA"
    },
    {
      "id_etudiant": "2",
      "nom": "BOUSSO",
      "prenom": "Lamine",
      "age": "25",
      "filiere": "Anglais"
    },
    {
      "id_etudiant": "3",
      "nom": "FALL",
      "prenom": "Fatou",
      "age": "21",
      "filiere": "Sociologie"
    },
```

Ionic Application

- Architecture du projet



src/app/app.module.ts

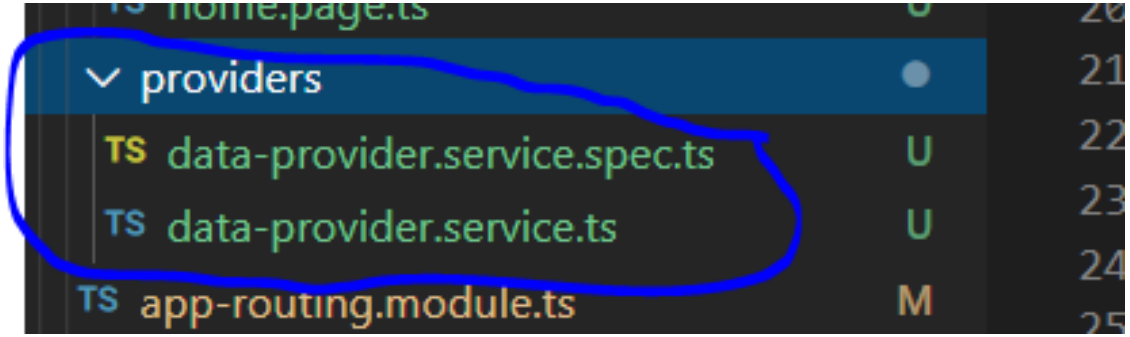
L'objet ***HttpClientModule*** nous permet d'éxécuter nos requêtes sur notre API

```
import {NgModule} from '@angular/core';
import {BrowserModule} from '@angular/platform-browser';
import {RouteReuseStrategy} from '@angular/router';
import {IonicModule, IonicRouteStrategy} from '@ionic/angular';
import {SplashScreen} from '@ionic-native/splash-screen/ngx';
import {StatusBar} from '@ionic-native/status-bar/ngx';
import {AppComponent} from './app.component';
import {AppRoutingModule} from './app-routing.module';
import {HttpClientModule} from '@angular/common/http';
import {FormModule} from './models/form/form.module';      You, a few
You, a few seconds ago | 1 author (You)
@NgModule({
    declarations: [AppComponent],
    entryComponents: [],
    imports: [
        BrowserModule,
        IonicModule.forRoot(),
        HttpClientModule,
        AppRoutingModule ],
    providers: [
        StatusBar,
        SplashScreen,
        {provide: RouteReuseStrategy, useClass: IonicRouteStrategy}
    ],
    bootstrap: [AppComponent]
})
export class AppModule {
}
```

Création du provider

- Le provider est un middleware pour communiquer avec notre API. Grâce au provider, on pourra avoir accès aux différentes méthodes de l'API pour faire des opérations.
- Nous allons générer notre provider par la commande suivante:

```
s\curd-mysql-php>ionic g service providers/dataProvider
```



src/app/providers/data-provider.service.ts

```typescript
import {Injectable} from '@angular/core';
import {HttpClient} from '@angular/common/http';

@Injectable({
    providedIn: 'root'
})
export class DataProviderService {
    host = 'http://localhost/apiIonic/';

    constructor(private http: HttpClient) {
    }

    list() {
        const url = this.host + '?action=list';
        return this.http.get<any>(url);
    }

    create(data) {
        const url = this.host + '?action=update';
        return this.http.post<any>(this.host, data);
    }

    update(data) {
        const url = this.host + '?action=list';
        return this.http.post<any>(this.host, data);
    }

    detail(id) {
        const url = this.host + '?action=detail&amp;id=' + id;
        return this.http.get<any>(url);
    }

    delete(id) {
        const url = this.host + '?action=list&amp;id=' + id;
        return this.http.get<any>(url);
    }

}
```

src/app/pages/home/home.page.html

```
<ion-header>
  <ion-toolbar color="primary">
    <ion-title>
      Accueil
    </ion-title>
  </ion-toolbar>
</ion-header>
<ion-content>
  <ion-refresher slot="fixed" (ionRefresh)="doRefresh($event)">
    <ion-refresher-content pullingIcon="arrow-dropdown"
      pullingText="Pull to refresh"
      refreshingSpinner="circles" refreshingText="Refreshing...">
    </ion-refresher-content>
  </ion-refresher>
  <div >
    <ion-list *ngIf="dataProviderService.data.length>0">
      <ion-list-header>
        Liste de repas
      </ion-list-header>
      <ion-item-sliding  *ngFor="let d of dataProviderService.data; let i=index ">
        <ion-item-options side="start">
          <ion-item-option (click)="deleteItem(i,d.id)" color="danger">
            <ion-icon name="trash"></ion-icon>
          </ion-item-option>
        </ion-item-options>
        <ion-item>
          <div>
```

```
            <ion-item>
              <div>
                <div class="title">{{d.nom}}</div>
<ion-header>
  <ion-toolbar color="primary">
    <ion-title>
      Accueil
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <ion-refresher slot="fixed" (ionRefresh)="doRefresh($event)">
    <ion-refresher-content
      pullingIcon="arrow-dropdown"
      pullingText="Pull to refresh"
      refreshingSpinner="circles"
      refreshingText="Refreshing...">
    </ion-refresher-content>
  </ion-refresher>
  <div >
    <ion-list *ngIf="dataProviderService.data.length>0">
      <ion-list-header>
        Liste des etudiants
      </ion-list-header>

      <ion-item-sliding
                *ngFor="let d of dataProviderService.data; let i=index ">
        <ion-item-options side="start">
```

src/app/pages/home/home.page.html

```
            <ion-item-options side="start">
                <ion-item-option (click)="deleteItem(i,d.id)" color="danger">
                    <ion-icon name="trash"></ion-icon>
                </ion-item-option>
            </ion-item-options>

            <ion-item>
                <div>
                    <div class="title">{{d.nom}}</div>
                    <div class="desc">{{d.prenom}}</div>
                    <div class="desc">{{d.age}}</div>
                    <div class="desc">{{d.filiere}}</div>
                </div>
            </ion-item>
            <ion-item-options side="end">
                <ion-item-option (click)="openModal(d)" color="primary">
                    <ion-icon ios="ios-create" md="md-create"></ion-icon>
                </ion-item-option>
            </ion-item-options>
        </ion-item-sliding>
    </ion-list>

    <div *ngIf="dataProviderService.data.length<1" class="empty-item">
        <span> Liste vide</span>
    </div>
</div>
```

```
    <ion-fab vertical="bottom" horizontal="end" slot="fixed">
        <ion-fab-button (click)="openModal()">
            <ion-icon name="add"></ion-icon>
        </ion-fab-button>
    </ion-fab>
</ion-content>                    </div>
                </ion-item>
                <ion-item-options side="end">
                    <ion-item-option (click)="openModal(d)" color="primary">
                        <ion-icon ios="ios-create" md="md-create"></ion-icon>
                    </ion-item-option>
                </ion-item-options>
            </ion-item-sliding>
        </ion-list>

        <div *ngIf="dataProviderService.data.length<1" class="empty-item">
            <span> Liste vide</span>
        </div>
    </div>
    <ion-fab vertical="bottom" horizontal="end" slot="fixed">
        <ion-fab-button (click)="openModal()">
            <ion-icon name="add"></ion-icon>
        </ion-fab-button>
    </ion-fab>
</ion-content>
```

src/app/pages/home/home.page.ts

```ts
import {Component, OnInit} from '@angular/core';
import {AlertController, ModalController, ToastController} from '@ionic/angular';
import {FormModal} from '../../models/form/form.modal';
import {DataProviderService} from '../../providers/data-provider.service';
@Component({
    selector: 'app-home',
    templateUrl: 'home.page.html',
    styleUrls: ['home.page.scss'],
})
export class HomePage {
    constructor(private modalController: ModalController, public alertController: AlertController,
                public toastController: ToastController, public dataProviderService: DataProviderService) { }
    async deleteItem(index, id) {
        const alert = await this.alertController.create({
            header: 'Supprimer!', message: 'Voullez vous vraiment supprimer?',
            buttons: [
                {
                    text: 'Annuler', role: 'cancel', cssClass: 'secondary',
                    handler: (blah) => {}
                }, {
                    text: 'Supprimer',
                    handler: () => {
                        this.dataProviderService.data.splice(index, 1);
                        this.dataProviderService.delete(id).subscribe(value => {
                            this.deleteMessage();
                        });
                    }
                }
            ]
        });
        await alert.present();
    }
```

```ts
    async deleteMessage() {
        const toast = await this.toastController.create({
            message: 'Repas correctement supprimé',
            duration: 2000
        });
        toast.present();
    }

    async openModal(data = null) {
        const modal = await this.modalController.create({
            component: FormModal,
            componentProps: {data: data},
            animated: true
        });
        return await modal.present();
    }

    ionViewWillEnter() {
        this.loadData();
    }

    loadData() {
        this.dataProviderService.list().subscribe(value => {
            this.dataProviderService.data = value.data;
        });
    }

    doRefresh(event) {
        this.loadData();
        setTimeout(() => {
            event.target.complete();
        }, 3000);
    }
```

FIN DE LA SEQUENCE 3

References

- https://ionicframework.com/docs/
- https://blog.nusyce.com/blog/
- https://openclassrooms.com/fr/courses/5098931-developpez-une-application-mobile-multiplateforme-avec-ionic-3
- https://ionic.mobiletuto.com/v3/chap10/chap10-5.html
- https://ionicframework.com/docs/components
- https://ionicacademy.com/ionic-routing-navigation/