

Programación para móviles

Investigación de metodología para desarrollo de proyectos

Diego Díaz Andeola
Matrícula: 2103150060
8°A ISW Tradicional

Docente: Fued Alejandro Majul Ramirez

04/03/2024



Universidad Politécnica de Durango
Institución Pública de la más alta calidad



```

Trabajo > src > app > interfaces > TS index.ts > TopLevel
1  export interface TopLevel {
2      id_mae?: number;
3      nombre?: string;
4      apodo?: string;
5      tel?: string;
6      foto?: string;
7  }
8

```

Las variables que estuve usando son aún las que estuvimos usando de ejemplo toda la unidad porque todavía no decidía la cantidad de variables que usaré para mi app. Estas variables son las que maneja el method de mi api para guardar los datos dentro de mi base de datos.

Api service

```

Trabajo > src > app > services > TS api.service.ts > ApiService > postDatos
7  @Injectable({
8      providedIn: 'root',
9  })
10 export class ApiService {
11     public apiUrl = 'http://127.0.0.1:80/api1/method.php'; // Reemplaza con la URL de tu API
12     constructor(private http: HttpClient) {}
13
14     getTopHeadlines(): Observable<TopLevel> {
15         return this.http
16             .get<TopLevel>('http://127.0.0.1:80/api1/method.php')
17             .pipe(map((resp) => resp));
18     }
19
20     // Método para enviar datos por POST
21     postDatos(datos: TopLevel): Observable<any> {
22         return this.http.post<any>(this.apiUrl, datos, {
23             responseType: 'text' as 'json',
24         });
25     }
26
27     eliminarDato(id_mae: number): Observable<string> {
28         return this.http.delete<string>(`${this.apiUrl}?id_mae=${id_mae}`, {
29             responseType: 'text' as 'json',
30         });
31     }
32 }
33

```

Aquí en el api service es en donde están las funciones para hacer el uso de mi api, donde están el get, post, y el delete.

Method

El archivo de method.php es el que tengo en mi carpeta de htdocs de xampp, en donde la api hace las consultas y todo lo que vamos a usar mediante el uso de la

pagina que creamos con el get, post, delete, y el put y patch. Las siguientes capturas son del get, el post y el delete, que son los que uso en la app.

```
> xampp > htdocs > api1 > method.php
14
15 require "config/Conexion.php";
16 parse_str(file_get_contents("php://input"), $datos);
17 //print_r($_SERVER['REQUEST_METHOD']);
18 switch ($_SERVER['REQUEST_METHOD']) {
19     case 'GET':
20         // Consulta SQL para seleccionar datos de la tabla de maestro
21         $sql = "SELECT id_mae, nombre, apodo, tel, foto FROM maestro";
22
23         $query = $conexion->query($sql);
24
25         if ($query->num_rows > 0) {
26             $data = array();
27             while ($row = $query->fetch_assoc()) {
28                 $data[] = $row;
29             }
30             // Devolver los resultados en formato JSON
31             header('Content-Type: application/json');
32             echo json_encode($data);
33         } else {
34             echo "No se encontraron registros en la tabla de maestro.";
35         }
36
37         $conexion->close();
38         break;
39 }
```

```
case 'POST':
    // Recibir los datos en formato JSON
    $data = json_decode(file_get_contents('php://input'), true);

    // Verificar si se han recibido los datos esperados
    if (isset($data['nombre']) && isset($data['apodo']) && isset($data['tel']) && isset($data['foto'])) {
        // Obtener los datos del arreglo JSON
        $nombre = $data['nombre'];
        $apodo = $data['apodo'];
        $tel = $data['tel'];
        $foto = $data['foto'];

        // Insertar los datos en la base de datos
        $sql = $conexion->prepare("INSERT INTO maestro (nombre, apodo, tel, foto) VALUES (?, ?, ?, ?)");
        $sql->bind_param("ssss", $nombre, $apodo, $tel, $foto);
        if ($sql->execute()) {
            echo "Datos insertados con éxito";
        } else {
            echo "Error al insertar datos: " . $sql->error;
        }
        $sql->close();
    } else {
        // Si faltan datos
        echo "Error: Faltan datos en la solicitud.";
    }
    break;
```

```

case 'DELETE':
    // Obtener el ID de usuario del arreglo $datos
    $id_mae = isset($_GET['id_mae']) ? $_GET['id_mae'] : null;

    // Verificar si se proporcionó el ID de usuario
    if ($id_mae === null) {
        echo "ID de usuario no proporcionado.";
        break; // Sale del switch si el ID de usuario no está presente
    }

    // Preparar la consulta de eliminación
    $stmt = $conexion->prepare("DELETE FROM maestro WHERE id_mae = ?");

    // Verificar si la preparación de la consulta fue exitosa
    if ($stmt === false) {
        echo "Error en la preparación de la consulta: " . $conexion->error;
        break;
    }

    // Vincular el parámetro ID de usuario
    $stmt->bind_param("i", $id_mae);

    // Ejecutar la consulta de eliminación
    if ($stmt->execute()) {
        echo "Registro eliminado con éxito.";
    } else {
        echo "Error al eliminar registro: " . $stmt->error;
    }
    break;

```

Rutas

```

Trabajo > src > app > TS app-routing.module.ts
1  import { NgModule } from '@angular/core';
2  import { PreloadAllModules, RouterModule, Routes } from '@angular/router';
3
4  const routes: Routes = [
5      {
6          path: '',
7          loadChildren: () => import('./pages/tabs/tabs.module').then(m => m.TabsPageModule)
8      },
9      {
10         path: 'registro',
11         loadChildren: () => import('./registro/registro.module').then( m => m.RegistroPageModule)
12     }
13 ];
14
15 @NgModule({
16     imports: [
17         RouterModule.forRoot(routes, { preloadingStrategy: PreloadAllModules })
18     ],
19     exports: [RouterModule]
20 })
21 export class AppRoutingModule {}
22

```

En esta parte es donde se encuentran las rutas del proyecto, donde el tabs es la página que se creó al momento que hicimos el ionic start, la cual tiene las 3 tabs que muestran el get, el post y la estructura del login con unos cuantos componentes más que agregué. La pagina que se llama registro es una página que agregué yo la

cual únicamente estuve usando para ver componentes, ya que como tal no tiene función, más que un botón para volver a la pagina de los tabs.

GET y DELETE

```
10  },
11  export class Tab1Page implements OnInit {
12    public resp: TopLevel[] = [];
13
14    constructor(private newService: ApiService) {}
15
16    ngOnInit() {
17      this.newService.getTopHeadlines().subscribe((resp) => {
18        console.log(resp); // Imprime el objeto TopLevel o arreglo TopLevel en la consola
19        if (Array.isArray(resp)) {
20          this.resp = resp; // Si es un arreglo, asigna directamente
21        } else {
22          this.resp = [resp]; // Si es un objeto, envuélvelo en un arreglo antes de asignar
23        }
24      });
25    }
26
27    eliminarDato(id_mae: number) {
28      this.newService.eliminarDato(id_mae).subscribe(
29        () => {
30          console.log('Usuario eliminado con éxito');
31          // Filtrar la lista actual para quitar el elemento eliminado
32          this.resp = this.resp.filter((item) => item.id_mae !== id_mae);
33        },
34        (error) => {
35          console.error('Error al eliminar el usuario:', error);
36        }
37      );
38    }
39  }
```

Para hacer uso del get, utilicé el primer tab, por lo que dentro del tab1.ts es donde está la función del get para mostrarlo en el html, así como también una función para borrar los registros, el cual es mediante un botón que está puesto en el html, que es el siguiente. El siguiente es el html del tab 1:

```
Trabajo > src > app > pages > tab1 > tab1.page.html > ion-header > ion-toolbar > ion-title
14  </ion-header>
15
16  <ion-grid>
17    <ion-row>
18      <ion-col *ngFor="let item of resp">
19        <ion-card>
20          <ion-card-header>
21            <ion-card-title>{{ item.id_mae }}</ion-card-title>
22          </ion-card-header>
23          <ion-card-content>
24            <p>ID: {{ item.id_mae }}</p>
25            <p>Nombre: {{ item.nombre }}</p>
26            <p>Apodo: {{ item.apodo }}</p>
27            <p>Tel: {{ item.tel }}</p>
28            <p>Foto: {{ item.foto }}</p>
29          </ion-card-content>
30          <ion-button (click)="item.id_mae !== undefined && eliminarDato(item.id_mae)" shape="round-10">
31            </ion-button>
32        </ion-card>
33      </ion-col>
34    </ion-row>
35  </ion-grid>
36  </ion-content>
```

POST

```
Trabajo > src > app > pages > tab2 > TS tab2.page.ts > Tab2Page
9     styleUrls: ['tab2.page.scss'],
10  });
11  export class Tab2Page {
12    nombre?: string;
13    apodo?: string;
14    tel?: string;
15    foto?: string;
16
17    constructor(private apiService: ApiService) {}
18
19    enviarDatos() {
20      const datos: TopLevel = {
21        nombre: this.nombre,
22        apodo: this.apodo,
23        tel: this.tel,
24        foto: this.foto,
25      };
26
27      // Configura los encabezados para indicar que se está enviando JSON
28      const headers = new HttpHeaders({
29        'Content-Type': 'application/json',
30      });
31
32      // Llama al método postDatos() del servicio ApiService para enviar los datos al servidor
33      this.apiService.postDatos(datos).subscribe(
34        (resp) => {
35          console.log(resp);
36          // Aquí puedes manejar la respuesta del servidor como desees
37        },
38        (error) => {
39          console.error('Error al enviar los datos:', error);
40          // Aquí puedes manejar el error de manera más específica
41        }
42      );
43    }
44  }
```

El post lo usé dentro del segundo tab, por lo que la función para enviar los datos está dentro del tab2.ts, y así usando el api que ya hicimos manda los datos a la base de datos usando el method.php.

La siguiente imagen es la del html del tab 2, donde básicamente están los inputs de los campos que se van a insertar con el post, así como un botón para hacer el envío de los datos:

```

Trabajo > src > app > pages > tab2 > tab2.page.html > ...
1 <ion-header [translucent]="true">
2   <ion-toolbar>
3     <ion-title>
4       Post
5     </ion-title>
6   </ion-toolbar>
7 </ion-header>
8 <ion-content [fullscreen]="true">
9   <ion-header collapse="condense">
10    <ion-toolbar>
11      <ion-title size="large">Tab 2</ion-title>
12    </ion-toolbar>
13  </ion-header>
14  <ion-item>
15    <ion-label position="floating">nombre</ion-label>
16    <ion-input type="text" [(ngModel)]="nombre"></ion-input>
17  </ion-item>
18  <ion-item>
19    <ion-label position="floating">apodo</ion-label>
20    <ion-input type="text" [(ngModel)]="apodo"></ion-input>
21  </ion-item>
22  <ion-item>
23    <ion-label position="floating">tel</ion-label>
24    <ion-input type="text" [(ngModel)]="tel"></ion-input>
25  </ion-item>
26  <ion-item>
27    <ion-label position="floating">foto</ion-label>
28    <ion-input type="text" [(ngModel)]="foto"></ion-input>
29  </ion-item>
30  <ion-button expand="block" (click)="enviarDatos()">Enviar</ion-button>
31 </ion-content>
32

```

Tab 3

Para el tab 3 únicamente tengo la estructura del login, pero como solo se nos pidió hacer la estructura de este, aun no funciona. Debajo de lo anterior, que es el login, tengo puestos varios componentes que son nada mas para rellenar, cuando siga haciendo cambios pondré algo acorde a la pantalla.

```

Trabajo > src > app > pages > tab3 > tab3.page.html > ...
1 <ion-header>
2   <ion-toolbar>
3     <ion-title>
4       Inicio de sesión
5     </ion-title>
6   </ion-toolbar>
7 </ion-header>
8
9 <ion-content>
10  <ion-item>
11    <ion-label position="floating">Usuario</ion-label>
12    <ion-input type="text" [(ngModel)]="username"></ion-input>
13  </ion-item>
14
15  <ion-item>
16    <ion-label position="floating">Contraseña</ion-label>
17    <ion-input type="password" [(ngModel)]="password"></ion-input>
18  </ion-item>
19
20  <ion-button expand="block" (click)="login()">Iniciar sesión</ion-button>
21  <ion-button expand="block" (click)="nextpage()">Registro</ion-button>
22

```

También es en este tercer tab en donde coloqué un botón que nos cambie a otra página, la cual llamé registro debido a que está en el mismo sitio que el login, pero como tal en esa pantalla solo estuve probando componentes, los cuales dejé ahí.

```
nextpage() {  
  this.router.navigate(['/registro']);  
}
```

En el html puse que al hacer clic en el botón que dice registro mande llamar al método nextpage para hacer la redirección, y este de arriba es el método que use para que mediante las rutas te lleve a la página de registro.

Página “registro”

```
Trabajo > src > app > registro > registro.page.html > ion-content > ion-content  
1 <ion-header [translucent]="true">  
2   <ion-toolbar>  
3     <ion-title>Registro</ion-title>  
4   </ion-toolbar>  
5 </ion-header>  
6  
7 <ion-content [fullscreen]="true">  
8   <ion-header collapse="condense">  
9     <ion-toolbar>  
10       <ion-title size="large">Registro</ion-title>  
11     </ion-toolbar>  
12   </ion-header>  
13   <ion-button (click)="prevpage()"><ion-icon name="chevron-back-outline"></ion-icon>Volver</ion-button>  
14  
15   <ion-card color="dark">  
16     <ion-card-header>  
17       <ion-card-title>El registro no está disponible</ion-card-title>  
18       <ion-card-subtitle>Bueno, no está hecho xd</ion-card-subtitle>  
19     </ion-card-header>  
20  
21     <ion-card-content> </ion-card-content>  
22   </ion-card>  
23   <ion-content color="light">  
24     <ion-list [inset]="true">  
25       <ion-item>  
26         <ion-input label="First Name"></ion-input>  
27       </ion-item>  
28       <ion-item>  
29         <ion-input label="Last Name"></ion-input>  
30       </ion-item>  
31       <ion-item>  
32         <ion-toggle>  
33           <ion-label>Allow Notifications</ion-label>  
34           <ion-note color="medium">Unsubscribe at any time</ion-note>  
35         </ion-toggle>  
36       </ion-item>  
37     </ion-list>
```

Como ya mencioné, esta página es la que se nos pidió hacer fuera de los tabs, la cual no contiene nada útil en su contenido mas que varios componentes con los que estuve experimentando y el botón que uso para volver a la página de los tabs.

```
prevpage() {  
  this.route.navigate(['/tabs']);  
}
```

Para que el botón de volver que hay en esta pantalla lo que hice fue lo mismo que el botón de registro pero a la inversa, al hacer clic al botón manda llamar la función dentro del archivo ts en donde mediante las rutas te devuelve a los tabs.