The Learning Hub

The Learning Hub © Copyright
2022- All Right Reserved

Angular

Tahaluf Training Center 2023

The Learning Hub © Copyright
2022- All Right Reserved

1. API Calls (Update)
2. API Calls (Upload & Retrieve Image).
3. Spinner.
4. Toastr.


API Calls (Update)

API Calls (Update)

Step one: Add a new function in the home service that calls API.

```
updateCourse(body: any) {
    //hits Api (create function)
    debugger
    this.http.put('https://localhost:44382/api/course/',
body).subscribe((resp: any) =>
     {
      alert('Updated Sucessfully');
    }, err => {
      alert('Something wont wrong');
      console.log(err);     })
  }
```

API Calls (Update)

Step two: Add a form group in an HTML file in the Manage course component to receive the new data for a specific row.

```
updateForm : FormGroup= new FormGroup({
  course_Id:new FormControl(),
  course_Name :new FormControl(''),
  price :new FormControl(),
  startdate :new FormControl(''),
  enddate :new FormControl(''),
  imagename :new FormControl('')
})
```

API Calls (Update)

Step three: Add a new button in the HTML file inside the table in the Mange course component and send the previous data as a parameter.

```
<button type="button"(click)=“openUpdateDailog(course)"
class="btn btn-primary m-1">Update</button>
```

API Calls (Update)

Step four: define an object to receive previous data.

```
previousData:any={} ;
openUpdateDailog(courseObj:any){
  this.previousData={
    course_Id:courseObj.course_Id,
    course_Name:courseObj.course_Name,
    price:courseObj.price,
    startdate:courseObj.startdate,
    enddate:courseObj.enddate }
    console.log(this.previousData);
  this.updateForm.controls['course_Id'].setValue
   (this.previousData.course_Id); }
```

API Calls (Update)

Step five: Add the template of the update form in the HTML file of the Manage course component.

API Calls (Update)

```
<ng-template #callUpdateDailog>
  <h2 mat-dialog-title>Create New Course </h2>
  <mat-dialog-content class="mat-typography">
      <form class="example-form" [formGroup]="updateForm">
          <mat-form-field class="example-full-width" appearance="fill">
              <mat-label>Course Name </mat-label>
              <input type="text" matInput formControlName="course_Name"
              [(ngModel)]="previousData.course_Name" >
          </mat-form-field>
          <br>
          <mat-form-field class="example-full-width" appearance="fill">
              <mat-label>Price</mat-label>
              <input type="number" matInput formControlName="price"
              [(ngModel)]="previousData.price">
          </mat-form-field>
          <br>
          <mat-form-field class="example-full-width" appearance="fill">
              <mat-label>Start Date </mat-label>
              <input type="date" matInput formControlName="startdate"
              [(ngModel)]="previousData.startdate">
          </mat-form-field>
          <br>
```

API Calls (Update)

```
        <mat-form-field class="example-full-width" appearance="fill">
            <mat-label>End Date </mat-label>
            <input type="date" matInput formControlName="enddate"
            [(ngModel)]="previousData.enddate">
        </mat-form-field>

        <div class="example-container">
          <input type="file" #file formControlName="imagename"
          (change)="uploadImage(file.files)"  >
          </div>
        <br>
    </form>
```

API Calls (Update)

In the managecourse.component.ts:

```
@ViewChild('callUpdateDailog') callUpdateDailog!: TemplateRef<any>
```

API Calls (Update)

Then complete the OpenUpdateDailog.

```
openUpdateDailog(courseObj: any) {
  this.previousData = {
    course_Id: courseObj.course_Id,
    course_Name: courseObj.course_Name,
    price: courseObj.price,
    startdate: courseObj.startdate,
    enddate: courseObj.enddate
  }
  console.log(this.previousData);
  this.updateForm.controls['course_Id'].setValue(this.previousData.course_Id);
  this.dialog.open(this.callUpdateDailog);
}
```

API Calls (Update)

Step six: Add a new button in the update form.

```
<button mat-button (click)="updateCourse()"
  [mat-dialog-close]="true“ cdkFocusInitial>Update</button>
```

API Calls (Update)

Step seven: Implement the update course method in the typescript file.

```
updateCourse() {
  this.home.updateCourse(this.updateForm.value);
}
```

The Learning Hub © Copyright
2023- All Right Reserved

Spinner

The Learning Hub
The Learning Hub © Copyright
2023- All Right Reserved

Overview of Spinner

An Angular library for loading spinners, ngx spinners shows that data is loading in real-time to the user and can be used to give the user a notification.

A loader is used to display the loading state of data in an application.

The Learning Hub
The Learning Hub © Copyright
2023- All Right Reserved

How to add spinner

Step one: from the npm website (node package manager) search for the ngx-spinner library.

Step two: Install the library using the terminal.

This is the installation command.

→ npm i ngx-spinner

The Learning Hub © Copyright
2023- All Right Reserved

How to add spinner

Step three: import the NgxSpinnerModule in the root module (App Module).

```
import { NgxSpinnerModule } from "ngx-spinner";
```

Step four: Adding the template you want in the App Component.

Step five: Add CSS animation files to angular.json config.

The Learning Hub
The Learning Hub © Copyright
2023- All Right Reserved

How to add spinner

Step six: Add NgxSpinnerService service wherever you want to use the ngx-spinner and create an object of this service in the constructor.

```
import { NgxSpinnerService } from "ngx-spinner";

constructor(private spinner: NgxSpinnerService){}
```

By using this object, you can show and hide the spinner using the show and hide.

The Learning Hub © Copyright
2023- All Right Reserved

Example

Add the spinner in the register component when the user clicks on submit button.

In the app.module.ts:

```
import { NgxSpinnerModule } from "ngx-spinner";
```

And add the NgxSpinnerModule in the import array.

The Learning Hub © Copyright
2023- All Right Reserved

Example

2. Add the template in the app component:

```
<ngx-spinner bdColor = "rgba(0, 0, 0, 0.8)" size = "medium"
color = "#fff"
type = "ball-fussion"
[fullScreen] = "true">
<p style="color: white" > Loading... </p>
</ngx-spinner>
```

The Learning Hub © Copyright
2023- All Right Reserved

Example

3. Add CSS animation files to angular.json config.

```
"node_modules/ngx-spinner/animations/ball-scale-multiple.css"

"styles": [
./node_modules/@angular/material/prebuilt-themes/
indigo-pink.css,
"src/styles.css",
"node_modules/ngx-spinner/animations/
ball-scale-multiple.css"
  ],
```

The Learning Hub © Copyright
2023- All Right Reserved

Example

4. Add NgxSpinnerService service in the register component

In register.component.ts

```
import { NgxSpinnerService } from "ngx-spinner";
```

In the constructor :

```
constructor(private router:Router,
private spinner: NgxSpinnerService)
```

The Learning Hub © Copyright
2023- All Right Reserved

Example

5. Use the show and hide functions in the submit method.

```
submit(){
      this.spinner.show();
      setTimeout(() => {
        /** spinner ends after 5 seconds */
        this.spinner.hide();
      }, 5000);
    }
```

The Learning Hub © Copyright
2023- All Right Reserved

Example

5. Use the show and hide functions in the submit method.

```
submit(){
      this.spinner.show();
      setTimeout(() => {
        /** spinner ends after 5 seconds */
        this.spinner.hide();
      }, 5000);
    }
```

The Learning Hub © Copyright
2023- All Right Reserved

Toastr

The Learning Hub © Copyright
2023- All Right Reserved

Overview of Toastr

Toastr is a JavaScript library that creates pop-up notifications.

It is simple, easy to use, and can be extended.

It allows you to create simple toasts using HTML5 and JavaScript.

The Learning Hub © Copyright
2023- All Right Reserved

Step to download the toastr package

Step One: Install the toastr package using this command:

```
npm install ngx-toastr --save
```

```
PS C:\Users\d.kanaan.ext\Desktop\TheLearningHub> npm i ngx-spinner

added 1 package, and audited 1015 packages in 11s

107 packages are looking for funding
  run `npm fund` for details

2 high severity vulnerabilities

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
```

The Learning Hub © Copyright
2023- All Right Reserved

Step to download the toastr package

Step Two: Add the style of the toastr in the angular.json file.

```
"styles": [
  "./node_modules/@angular/material/prebuilt-themes/indigo-pink.css",
  "src/styles.css",
  "node_modules/ngx-spinner/animations/ball-fussion.css",
  "node_modules/ngx-toastr/toastr.css"
],
```

The Learning Hub © Copyright
2023- All Right Reserved

Step to download the toastr package

Step Three: Import the ToasterModule and ToastrNoAnimationModule in the root module(app.module.ts).

```
import {ToastrModule, ToastNoAnimation,
ToastNoAnimationModule}
from 'ngx-toastr'
```

Note: Don't forget to add the module's name in the import array.

```
imports: [
  BrowserModule,
  AppRoutingModule,
  BrowserAnimationsModule,
  NgxSpinnerModule,
  SharedModule,
  ToastNoAnimationModule.forRoot(),
  ToastrModule.forRoot()
],
```

The Learning Hub © Copyright
2023- All Right Reserved

Example

To add the toastr in the home component

1. Import the ToastrService in the typescript of the component.

So, In home.component.ts.

```
import {ToastrService} from 'ngx-toastr';
```

The Learning Hub © Copyright
2023- All Right Reserved

Example

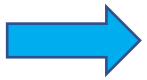
2. Create an instance of toastr service in the constructor of the component and then use this object to access the toastr method like success, warning, error, and info.

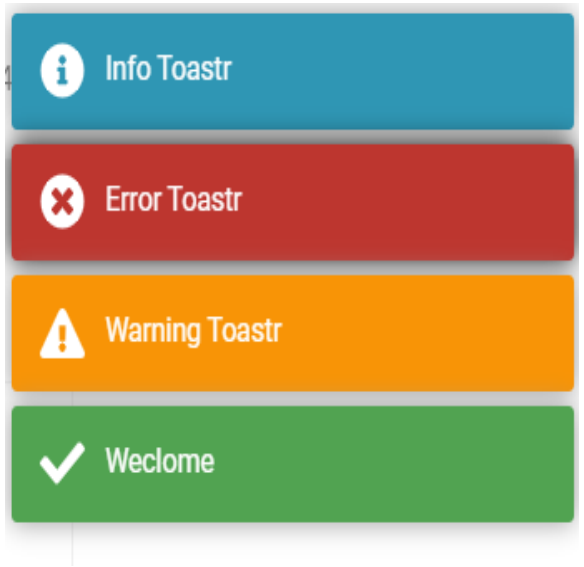
```
export class HomeComponent implements OnInit {

  constructor(private toastr:ToastrService) { }

  ngOnInit(): void {
    this.toastr.success('Weclome');
    this.toastr.warning('Warning Toastr');
    this.toastr.error('Error Toastr');
    this.toastr.info('Info Toastr');

  }
}
```





The Learning Hub © Copyright
2023- All Right Reserved

Exercise

Update the API functionality for whole home services as follows:

1. Show the spinner.
2. Connect to the API.
3. Display a toastr as per the response statutes.
4. Hide the spinner.

The Learning Hub © Copyright
2022- All Right Reserved

Break

9:00 - 10:00

References :

1. Angular, “Angular,” *Angular.io*, 2019. https://angular.io/

2. “Complete Angular Tutorial For Beginners,” *TekTutorialsHub*. https://www.tektutorialshub.com/angular-tutorial/

3. “npm | build amazing things,” *Npmjs.com*, 2019. https://www.npmjs.com/

4. “Angular Tutorial for Beginners | Simplilearn,” *Simplilearn.com*. https://www.simplilearn.com/tutorials/angular-tutorial (accessed Aug. 19, 2022).

The Learning Hub © Copyright
2022- All Right Reserved

Thank You