Harmony IT Solution

Angular

Tahaluf Training Center 2022

1. Create Admin Dashboard.
2. Hits API (Post).
3. Upload And Retrieve Image.

Harmony IT Solution

Objective

The Objective of this lecture

- Generate the admin dashboard and set all responsibilities for the admin.
- Understand the HTTP protocol and how to deal with HTTP posts and send the data from the body.
- Woking with the image and knowing how to upload and retrieve the image from API.

Harmony IT Solution

Create Admin Dashboard

Harmony IT Solution

Overview of Dashboard

The Dashboard provides a visual representation of your company's reports in a very easy way.

Charts display real-time information (e.g. technician productivity by work type, resource usage by a number of work orders, etc.)

How to create a dashboard for our project

Step one: Create a new module called Admin .

```
PS C:\Users\d.kanaan.ext\Desktop\EduTech> ng g m admin --routing
CREATE src/app/admin/admin-routing.module.ts (248 bytes)
CREATE src/app/admin/admin.module.ts (276 bytes)
PS C:\Users\d.kanaan.ext\Desktop\EduTech>
```

Steps to download the Toastr Library in the project

Step two: add the route of the admin module in the root module.

In app-routing.module.ts

```
{
  path:'admin',
  loadChildren:()=>AdminModule
}
```

How to create a dashboard for our project

Step three: Create a new component called sidebar inside the Admin module.

How to create a dashboard for our project

Note: The purpose of this component is to add a sidebar template for each component that the admin is responsible for.

How to create a dashboard for our project

Step four: Add the routing for the sidebar inside the admin module.

Steps to download the Toastr Library in the project

Step five: Use this link to add the sidebar style to the sidebar component.

https://codepen.io/michaelmcshinsky/pen/vYMdrb

Note: Modify the sidebar to fit the admin's responsibilities according to the project.

Hits API (Post)

Overview Of Http Post

Makes HTTP requests. This service is provided as an injectable class that includes methods for making HTTP requests.

During form submission, apps often use POST requests to transmit data to the server.

Example HTTP Post

1. Create a new component in the admin module called manage course to manage all operations that occur on the course table.

```
PS C:\Users\d.kanaan.ext\Desktop\EduTech> ng g c admin/managecourse
CREATE src/app/admin/managecourse/managecourse.component.html (27 bytes)
CREATE src/app/admin/managecourse/managecourse.component.spec.ts (668 bytes)
CREATE src/app/admin/managecourse/managecourse.component.ts (299 bytes)
CREATE src/app/admin/managecourse/managecourse.component.css (0 bytes)
UPDATE src/app/admin/admin.module.ts (466 bytes)
```

Example HTTP Post

2. Add the route for manage the course component for its module (admin module)

In admin-routing.module.ts

```
{
  path:'managecourse',
  component:ManagecourseComponent
}
```

Example HTTP Post

3. Create a new component to contain the template of the form to create a new course.

```
PS C:\Users\d.kanaan.ext\Desktop\EduTech> ng g c admin/create
CREATE src/app/admin/create/create.component.html (21 bytes)
CREATE src/app/admin/create/create.component.spec.ts (626 bytes)
CREATE src/app/admin/create/create.component.ts (275 bytes)
CREATE src/app/admin/create/create.component.css (0 bytes)
UPDATE src/app/admin/admin.module.ts (360 bytes)
PS C:\Users\d.kanaan.ext\Desktop\EduTech>
```

Example HTTP Post

In our project, we'll use an angular material dialog box in the Create component.

4. So, add the API for the dialog in the Shared module.

```
import {MatDialogModule} from '@angular/material/dialog';
```

Example HTTP Post

5. Add the name of the dialog module in the import and export array.

```
imports: [
  CommonModule,
  RouterModule,
  FormsModule,
  ReactiveFormsModule,
  MatFormFieldModule,
  MatInputModule,
  MatDialogModule
],
```

```
exports:[
  FormsModule,
  ReactiveFormsModule,
  MatFormFieldModule,
  MatInputModule,
  NavbarComponent,
  FooterComponent,
  MatDialogModule
]
```

Example HTTP Post

6. Define a form group in the Create component, and use it to send data to the database.

```
createform:FormGroup=new FormGroup({
coursename: new FormControl('',Validators.required),
price:new FormControl('',Validators.required),
startdate:new FormControl('',Validators.required),
enddate:new FormControl('',Validators.required),
imagename:new FormControl()
})
```

Example HTTP Post

7. Add the template form in the HTML file for Create component.

```
<h2 mat-dialog-title>Create New Course </h2>
<mat-dialog-content class="mat-typography">
<form class="example-form" [formGroup]="createform" >
        <mat-form-field class="example-full-width" appearance="fill">
          <mat-label>Course Name </mat-label>
           <input type="text" matInput formControlName="coursename">
            <mat-error *ngIf="createform.controls['coursename'].hasError('required')">
            course name is <strong>required</strong>
          </mat-error>
        </mat-form-field>
        <br>
```

Example HTTP Post

```
<mat-form-field class="example-full-width" appearance="fill">
 <mat-label>Price</mat-label>
 <input type="number" matInput formControlName="price">
  <mat-error *ngIf="createform.controls['price'].hasError('required')">
    price is <strong>required</strong>
    </mat-error>
 </mat-form-field>
 <br>
```

Example HTTP Post

```
<mat-form-field class="example-full-width" appearance="fill">
  <mat-label>Start Date </mat-label>
  <input type="date" matInput formControlName="startdate">
  <mat-error *ngIf="createform.controls['startdate'].hasError('required')">
    Start Date  is <strong>required</strong>
  </mat-error>
  </mat-form-field>
   <br>
  <mat-form-field class="example-full-width" appearance="fill">
  <mat-label>End Date </mat-label>
  <input type="date" matInput formControlName="enddate">
  <mat-error *ngIf="createform.controls['enddate'].hasError('required')">
  End Date is <strong>required</strong>
  </mat-error>
</mat-form-field>
<br>
</form>
```

Example HTTP Post

8. Create a function in the home service.

```
createCourse(body:any){
  //show spinner
  this.spinner.show();
  //hits Api (create function)
  debugger
  this.http.post('https://localhost:44320/api/course',body).subscribe((resp:any)=>{
    //hide spinner
    this.spinner.hide();
    //resp --> toastr
    this.toastr.success('Created Successfully ');
  },err=>{
    //hide spinner
    this.spinner.hide;
    //resp --> toastr
    this.toastr.error(err.message , err.status)
  })}
```

Example HTTP Post

9. Create a function in the Create component to call a created course function from home services.

```
saveCourse(){
  debugger
  this.home.createCourse(this.createform.value);
}
```

Example HTTP Post

10. Add a button in the HTML file of the Create component to call a seveCourse function.

```
<mat-dialog-actions align="end">
  <button mat-button mat-dialog-close>Cancel</button>
  <button mat-button (click)="saveCourse()"[mat-dialog-close]="true"
   cdkFocusInitial>Save</button>
</mat-dialog-actions>
```

Example HTTP Post

11. Add a button in the HTML file of the Manage course component to open the create dialog .

```
<button class="btn btn-success" (click)="openDialog()">
Create new Courses </button>
```

And in the typescript file for this component create an object of MatDialog Service to use an open method from this service.

Example HTTP Post

So, In managecourse.component.ts

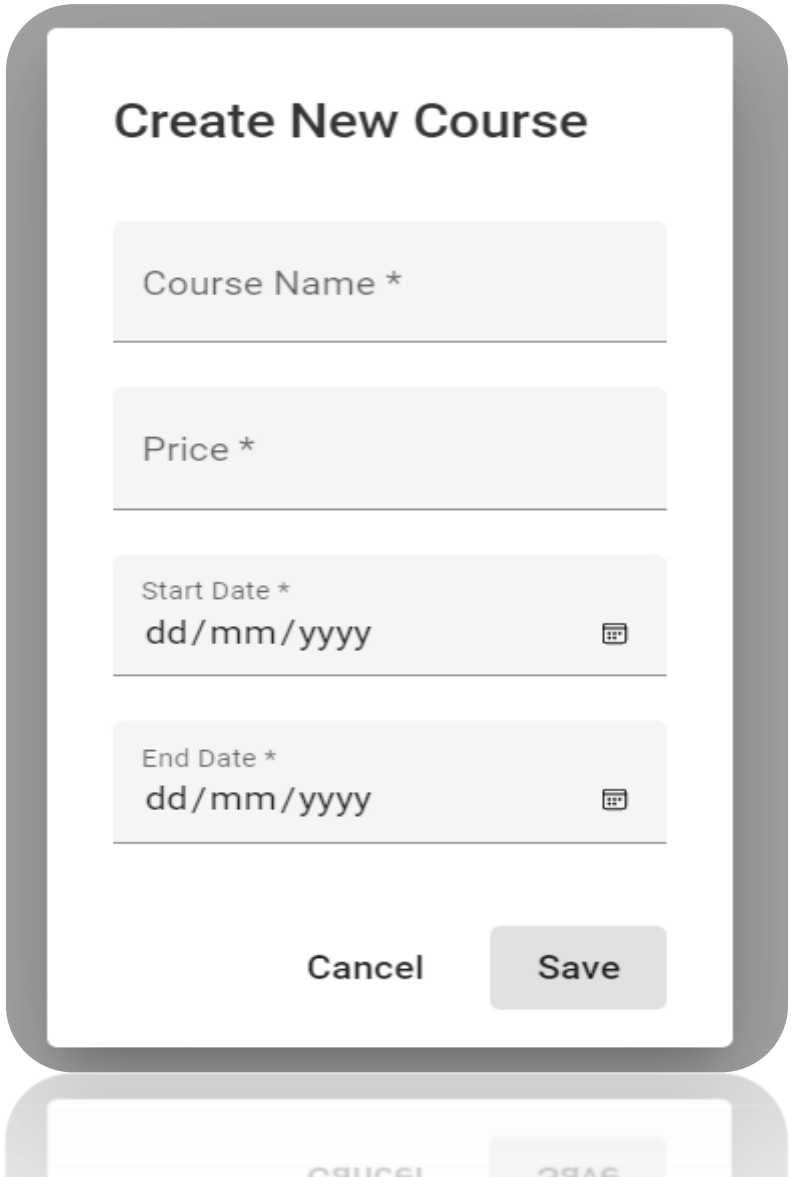
```
constructor(private dialog :MatDialog) { }

openDialog() {
    const dialogRef = this.dialog.open(CreateComponent);
  }
```

Note: The open function receives in the Parmenter the name of loaded

Harmony IT Solution

Example HTTP Post

The Result

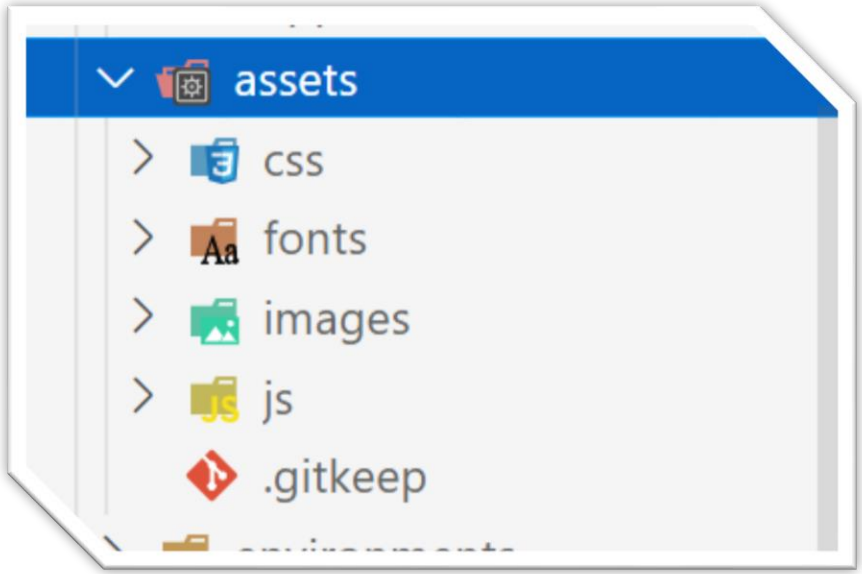


Harmony IT Solution

Upload and Retrieve Image

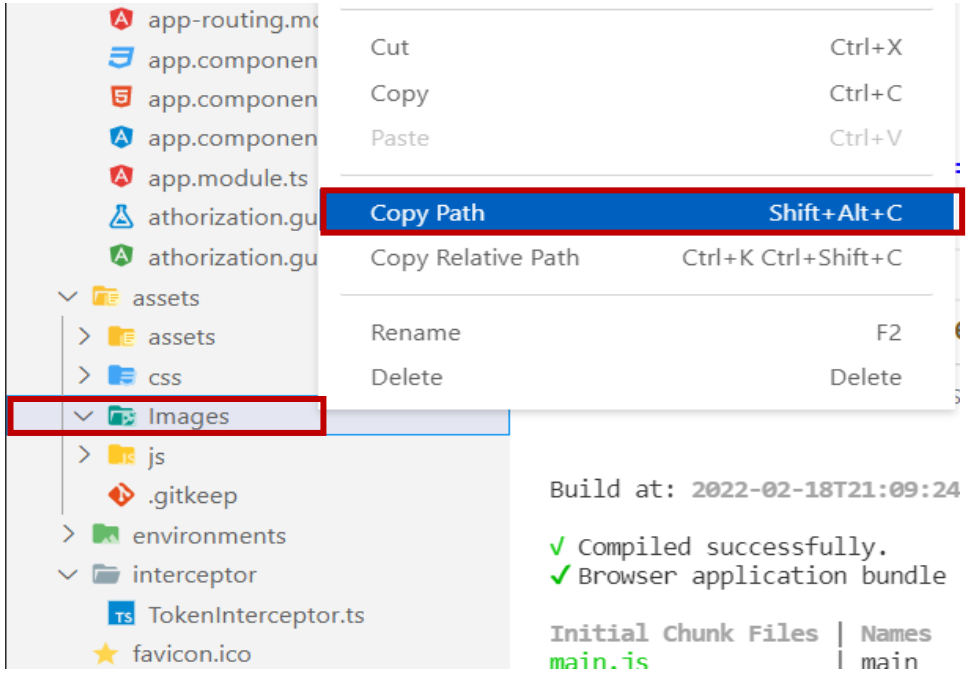
Steps to upload and retrieve images in angular

Step one: Create a folder called images in the assets folder.



Steps to upload and retrieve images in angular

Step two: Copy the path of the image folder.



Steps to upload and retrieve images in angular

Step three: Update the API project => Course Controller => UploadImage method

Paste the copied path as the following:

```
var file = Request.Form.Files[0];
var fileName = Guid.NewGuid().ToString() + "_" + file.FileName;
var fullPath = Path.Combine("C:\\Users\\d.kanaan.ext\\Desktop\\EduTech\\src\\assets\\images", fileName);
```

Steps to upload and retrieve images in angular

Step four: Define property display_image and Create a function in the home service to hits API for the upload function.

```
display_image:any
```

```
uploadAttachment(file: FormData) {
  this.http.post('https://localhost:44320/api/course/uploadImage', file)
  .subscribe(
    (data: any) => {
      this.display_image = data.imagename;
      debugger
      if (data) {
        console.log(data);
      }
    }, err => { console.log(err); })
}
```

Steps to upload and retrieve images in angular

Step five: In the HTML file for the Create component add the following to create a form

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

Steps to upload and retrieve images in angular

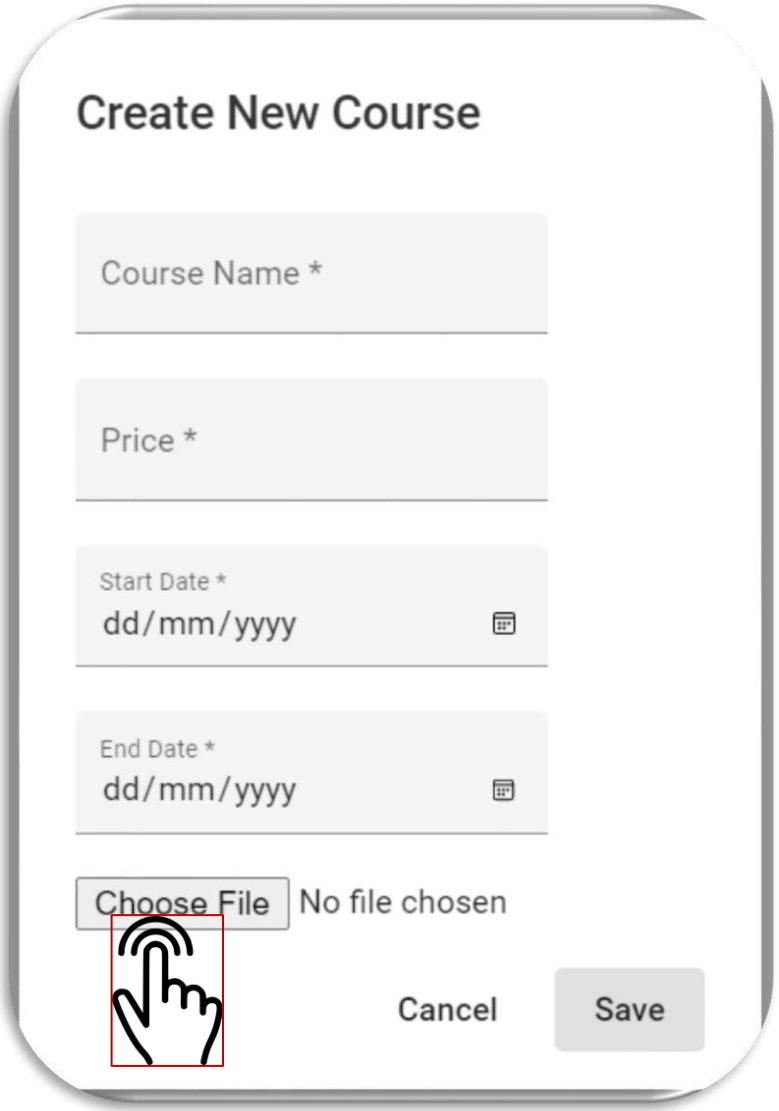
Step six: Implement the UploadFile function in the typescript of Create component.

```
uploadFile(files:any) {
  if (files.length === 0) {
  return;
  }
  let fileToUpload = <File>files[0];
  const formData = new FormData();
  formData.append('file', fileToUpload, fileToUpload.name);
  this.home.uploadAttachment(formData);
   }
```

Harmony IT Solution

Steps to upload and retrieve images in angular

The Result



Steps to upload and retrieve images in angular

To retrieve the image , In course-card component .html add this code.

```
<div class="courses-image">
    <img (click)="showPorfile()" src="../../../assets/images/{{imagename}}"
     style="width: 300px;height: 200px;"
     class="img-responsive" alt="">
</div>
```

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).

Harmony IT Solution