**Update Category Http Put -API**

**IcategoryRepository**

Task<Category?> UpdateAsync(Category category);

**CategoryRepository**

public async Task<Category?> UpdateAsync(Category category)

        {

           var existingCategory = await dbContext.Categories.FirstOrDefaultAsync(x => x.Id == category.Id);

            if(existingCategory != null)

            {

                dbContext.Entry(existingCategory).CurrentValues.SetValues(category);

                await dbContext.SaveChangesAsync();

                return category;

            }

            return null;

        }

**CategoriesController**

[HttpPut]

        [Route("{id:int}")]

        public async Task<IActionResult> EditCategory([FromRoute] int id,UpdateCategoryRequestDto request)

        {

            var category= new Category

            {

                Id=id,

                Name = request.Name,

                UrlHandle = request.UrlHandle

            };

            category = await categoryRepository.UpdateAsync(category);

            if (category == null)

            {

                return NotFound();

            }

            var response = new CategoryDto

            {

              Id = category.Id,

              Name = category.Name,

              UrlHandle = category.UrlHandle

            };

            return Ok(response);

        }

**Category.services**

updateCategory( id: string, updateCategoryRequest: UpdateCategoryRequest): Observable<Category> {

   return this.http.put<Category>('http://localhost:5125/api/categories/'+id , updateCategoryRequest);

  }

**Edit-Category.component.ts**

import { Component, OnDestroy, OnInit } from '@angular/core';

import { ActivatedRoute, Router } from '@angular/router';

import { Subscription } from 'rxjs';

import { CategoryService } from '../services/category.service';

import { Category } from '../models/category.model';

import { UpdateCategoryRequest } from '../models/update-category-request.model';

@Component({

  selector: 'app-edit-category',

  templateUrl: './edit-category.component.html',

  styleUrls: ['./edit-category.component.css']

})

export class EditCategoryComponent implements OnInit, OnDestroy {

  id: string | null =null;

  paramsSubscription?: Subscription;

  editCategorySubscription?: Subscription;

  category?: Category;

  constructor(private route: ActivatedRoute,

    private categoryService: CategoryService,

    private router:Router) {

  }

  ngOnInit(): void {

    this.paramsSubscription = this.route.paramMap.subscribe({

      next: (params) => {

        this.id = params.get('id');

        if (this.id) {

          this.categoryService.getCategoryById(this.id)

          .subscribe({

            next: (response) => {

              this.category =  response;

            }

          });

        }

      }

    });

  }

  onFormSubmit():void{

    const UpdateCategoryRequest: UpdateCategoryRequest = {

      name: this.category?.name ?? '',

      urlHandle: this.category?.urlHandle ?? ''

    };

    if(this.id){

     this.editCategorySubscription = this.categoryService.updateCategory(this.id, UpdateCategoryRequest)

    .subscribe({

      next: (response) => {

        this.router.navigateByUrl('/admin/categories');

      }

    });

    }

  }

  ngOnDestroy(): void {

    this.paramsSubscription?.unsubscribe();

    this.editCategorySubscription?.unsubscribe();

  }

}

**Edit-category.component.html**

<div class="container">

    <h1 class="mt-3"> Edit Category </h1>

    <ng-container \*ngIf="category">

        <form>

            <div class="mt-2">

                <div class="mt-3">

                    <label class="form-label" for="categoryId"> Category Id</label>

                    <input type ="text" disabled

                    id="categoryId"

                    class="form-control"

                    name="categoryId"

                    value="{{category.id}}">

                </div>

                <div class="mt-3">

                    <label class="form-label" for="categoryName"> Category Name</label>

                    <input type ="text"

                    id="categoryName"

                    class="form-control"

                    name="categoryName"

                    value="{{category.name}}">

                </div>

                <div class="mt-3">

                    <label class="form-label" for="categoryUrlHandle"> Url Handle</label>

                    <input type ="text"

                    id="categoryUrlHandle"

                    class="form-control"

                    name="categoryUrlHandle"

                    value="{{category.urlHandle}}">

                </div>

            </div>

        </form>

        <div class="mt-3 d-flex">

            <button type="submit" class="btn btn-primary"> Save </button>

            <button type="button" class="btn btn-danger ms-3" (click)="onDelete()"> Delete </button>

          </div>

    </ng-container>

    <ng-template #notFound>

        <div class="alert alert-warning" role="alert">

            Category Not Found!

          </div>

    </ng-template>

</div>

**Delete category by id -API**

**IcategoryRepository**

Task<Category?> DeleteAsync(int id);

**CategoryRepository**

public async Task<Category?> DeleteAsync(int id)

        {

            var existingCategory = await dbContext.Categories.FirstOrDefaultAsync(x => x.Id == id);

            if (existingCategory is null)

            {

                return null;

            }

            dbContext.Categories.Remove(existingCategory);

            await dbContext.SaveChangesAsync();

            return existingCategory;

        }

**CategoriesController**

[HttpDelete]

        [Route("{id:int}")]

        public async Task<IActionResult> DeleteCategory([FromRoute] int id)

        {

            var category = await categoryRepository.DeleteAsync(id);

            if(category is null)

            {

                return NotFound();

            }

            var response = new CategoryDto

            {

                Id = category.Id,

                Name = category.Name,

                UrlHandle = category.UrlHandle

            };

            return Ok(response);

**Delete category by id -UI**

**Category.services**

deleteCategory(id:string): Observable<Category>{

   return this.http.delete<Category>('http://localhost:5125/api/categories/'+id)

  }

**Delete category component**

import { Component, OnDestroy, OnInit } from '@angular/core';

import { ActivatedRoute, Router } from '@angular/router';

import { Subscription } from 'rxjs';

import { CategoryService } from '../services/category.service';

import { Category } from '../models/category.model';

import { UpdateCategoryRequest } from '../models/update-category-request.model';

@Component({

  selector: 'app-edit-category',

  templateUrl: './edit-category.component.html',

  styleUrls: ['./edit-category.component.css']

})

export class EditCategoryComponent implements OnInit, OnDestroy {

  id: string | null =null;

  paramsSubscription?: Subscription;

  editCategorySubscription?: Subscription;

  category?: Category;

  constructor(private route: ActivatedRoute,

    private categoryService: CategoryService,

    private router:Router) {

  }

  ngOnInit(): void {

    this.paramsSubscription = this.route.paramMap.subscribe({

      next: (params) => {

        this.id = params.get('id');

        if (this.id) {

          this.categoryService.getCategoryById(this.id)

          .subscribe({

            next: (response) => {

              this.category =  response;

            }

          });

        }

      }

    });

  }

  OnFormSubmit():void{

    const UpdateCategoryRequest: UpdateCategoryRequest = {

      name: this.category?.name ?? '',

      urlHandle: this.category?.urlHandle ?? ''

    };

    if(this.id){

     this.editCategorySubscription = this.categoryService.updateCategory(this.id, UpdateCategoryRequest)

    .subscribe({

      next: (response) => {

        this.router.navigateByUrl('/admin/categories');

      }

    });

    }

  }

  onDelete(): void{

    if(this.id){

    this.categoryService.deleteCategory(this.id)

    .subscribe({

      next: (response) =>{

        this.router.navigateByUrl('/admin/categories');

      }

    })

    }

  }

  ngOnDestroy(): void {

    this.paramsSubscription?.unsubscribe();

    this.editCategorySubscription?.unsubscribe();

  }

}

**Edit(delete)-category.component.html**

<div class="container">

    <h1 class="mt-3"> Edit Category </h1>

    <ng-container \*ngIf="category">

        <form>

            <div class="mt-2">

                <div class="mt-3">

                    <label class="form-label" for="categoryId"> Category Id</label>

                    <input type ="text" disabled

                    id="categoryId"

                    class="form-control"

                    name="categoryId"

                    value="{{category.id}}">

                </div>

                <div class="mt-3">

                    <label class="form-label" for="categoryName"> Category Name</label>

                    <input type ="text"

                    id="categoryName"

                    class="form-control"

                    name="categoryName"

                    value="{{category.name}}">

                </div>

                <div class="mt-3">

                    <label class="form-label" for="categoryUrlHandle"> Url Handle</label>

                    <input type ="text"

                    id="categoryUrlHandle"

                    class="form-control"

                    name="categoryUrlHandle"

                    value="{{category.urlHandle}}">

                </div>

            </div>

        </form>

        <div class="mt-3 d-flex">

            <button type="submit" class="btn btn-primary"> Save </button>

            <button type="button" class="btn btn-danger ms-3" (click)="onDelete()"> Delete </button>

          </div>

    </ng-container>

    <ng-template #notFound>

        <div class="alert alert-warning" role="alert">

            Category Not Found!

          </div>

    </ng-template>

</div>