# EmployeeController.cs

using Microsoft.AspNetCore.Mvc;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using OrchardAngular.Data.Models;

using OrchardAngular.Data.Models.Repository;

// For more information on enabling Web API for empty projects, visit https://go.microsoft.com/fwlink/?LinkID=397860

namespace OrchardAngular.WebAPI.Controllers

{

[Route("api/[controller]")]

[ApiController]

public class EmployeeController : Controller

{

private IEmployeeRepository \_repository;

public EmployeeController(IEmployeeRepository repository)

{

this.\_repository = repository;

}

// GET: api/<EmployeeController>

[HttpGet]

public IEnumerable<Employee> Get()

{

return \_repository.GetEmployee();

}

// GET api/<EmployeeController>/5

[HttpGet("{id:int}")]

public Employee Get(int id)

{

return \_repository.GetEmployeeById(id);

}

[HttpGet("{id:int}/name")]

public string GetNameById(int id)

{

return \_repository.GetEmployeeNameById(id);

}

// POST api/<EmployeeController>

[HttpPost]

public int Post(int deptId, [FromBody] Employee employee)

{

return \_repository.AddEmployee(deptId, employee);

}

// PUT api/<EmployeeController>/5

[HttpPut("{id:int}")]

public int Put(int id,Employee employee)

{

return \_repository.UpdateEmployee(id,employee);

}

// DELETE api/<EmployeeController>/5

[HttpDelete("{id:int}")]

public int Delete(int id)

{

return \_repository.DeleteEmployee(id);

}

}

}

# IEmployeeRepository.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace OrchardAngular.Data.Models

{

public interface IEmployeeRepository

{

IList<Employee> GetEmployee();

Employee GetEmployeeById(int id);

string GetEmployeeNameById(int id);

int AddEmployee(int deptId, Employee employee);

int UpdateEmployee(int id, Employee employee);

int DeleteEmployee(int id);

}

}

# EmployeeRepository.cs

using Microsoft.EntityFrameworkCore;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace OrchardAngular.Data.Models.Repository

{

public class EmployeeRepository : IEmployeeRepository

{

private OrchardAngularContext \_context;

public EmployeeRepository(OrchardAngularContext context)

{

this.\_context = context;

}

public int AddEmployee(int deptId,Employee employee)

{

employee.Department=\_context.Departments.FirstOrDefault(a => a.Id == deptId);

\_context.Add(employee);

return \_context.SaveChanges();

}

public int DeleteEmployee(int id)

{

Employee employee = \_context.Employees.FirstOrDefault(a => a.Id == id);

if (employee != null)

{

\_context.Remove(employee);

}

return \_context.SaveChanges();

}

public IList<Employee> GetEmployee()

{

List<Employee> employees = \_context.Employees.Include(a=>a.Department).ToList();

foreach (Employee emp in employees)

{

if (emp.Department!=null)

emp.Department.Employees = null;

}

return employees;

}

public Employee GetEmployeeById(int id)

{

Employee emp = \_context.Employees.Include(a=>a.Department).FirstOrDefault(a=>a.Id==id);

emp.Department.Employees = null;

return emp;

}

public string GetEmployeeNameById(int id)

{

Employee emp = \_context.Employees.Include(a => a.Department).FirstOrDefault(a => a.Id == id);

emp.Department.Employees = null;

return emp.Name;

}

public int UpdateEmployee(int id, Employee employee)

{

Employee oldEmployee = \_context.Employees.FirstOrDefault(a => a.Id == id);

if (oldEmployee != null)

{

employee.Id = oldEmployee.Id;

\_context.Update(employee);

}

return \_context.SaveChanges();

}

}

}

# UnitTest1.cs

using Microsoft.VisualStudio.TestTools.UnitTesting;

using Moq;

using OrchardAngular.Data.Models;

using OrchardAngular.WebAPI.Controllers;

namespace OrchardAngular.Test

{

[TestClass]

public class UnitTest1

{

[TestMethod]

public void TestMethod1()

{

var mock = new Mock<IEmployeeRepository>();

mock.Setup(p => p.GetEmployeeNameById(1)).Returns("Roshan");

EmployeeController home = new EmployeeController(mock.Object);

string result = home.GetNameById(1);

Assert.AreEqual("Roshan", result);

}

}

}