Using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace ProjectStudInfowebapi.Model

{

public class Student

{

public int ID { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public string Address { get; set; }

public string Location { get; set; }

public int PhoneNumber { get; set; }

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using Microsoft.EntityFrameworkCore;

using ProjectStudInfowebapi.Data;

using ProjectStudInfowebapi.Model;

namespace ProjectStudInfowebapi.Controllers

{

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

[ApiController]

public class StudentsController : ControllerBase

{

private readonly ProjectStudInfowebapiContext \_context;

public StudentsController(ProjectStudInfowebapiContext context)

{

\_context = context;

}

// GET: api/Students

[HttpGet]

public async Task<ActionResult<IEnumerable<Student>>> GetStudent()

{

return await \_context.Student.ToListAsync();

}

// GET: api/Students/5

[HttpGet("{id}")]

public async Task<ActionResult<Student>> GetStudent(int id)

{

var student = await \_context.Student.FindAsync(id);

if (student == null)

{

return NotFound();

}

return student;

}

// PUT: api/Students/5

// To protect from overposting attacks, enable the specific properties you want to bind to, for

// more details, see https://go.microsoft.com/fwlink/?linkid=2123754.

[HttpPut("{id}")]

public async Task<IActionResult> PutStudent(int id, Student student)

{

if (id != student.ID)

{

return BadRequest();

}

\_context.Entry(student).State = EntityState.Modified;

try

{

await \_context.SaveChangesAsync();

}

catch (DbUpdateConcurrencyException)

{

if (!StudentExists(id))

{

return NotFound();

}

else

{

throw;

}

}

return NoContent();

}

// POST: api/Students

// To protect from overposting attacks, enable the specific properties you want to bind to, for

// more details, see https://go.microsoft.com/fwlink/?linkid=2123754.

[HttpPost]

public async Task<ActionResult<Student>> PostStudent(Student student)

{

\_context.Student.Add(student);

await \_context.SaveChangesAsync();

return CreatedAtAction("GetStudent", new { id = student.ID }, student);

}

// DELETE: api/Students/5

[HttpDelete("{id}")]

public async Task<ActionResult<Student>> DeleteStudent(int id)

{

var student = await \_context.Student.FindAsync(id);

if (student == null)

{

return NotFound();

}

\_context.Student.Remove(student);

await \_context.SaveChangesAsync();

return student;

}

private bool StudentExists(int id)

{

return \_context.Student.Any(e => e.ID == id);

}

}

}

















