using Microsoft.AspNetCore.Mvc;

using System.Reflection.Metadata;

using ASPNetCoreApp.Models;

using Microsoft.AspNetCore.Mvc;

using Microsoft.EntityFrameworkCore;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace AspNetCoreApp.Controllers

{

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

[ApiController]

public class DogovorsController : ControllerBase

{

private readonly OperatorContext \_context;

public DogovorsController(OperatorContext context)

{

\_context = context;

if (\_context.Dogovor.Count() == 0)

{

\_context.Dogovor.Add(new Dogovor { Номер\_договора = 1, Дата\_заключения = DateTime.Today, Дата\_расторжения=DateTime.Today, Код\_тарифа\_FK= 1, Номер\_телефона="000", Номер\_клиента\_FK='1', Серийный\_номер\_сим\_карты="000"});

\_context.SaveChanges();

}

}

[HttpGet]

public IEnumerable<Dogovor> GetAll()

{

return \_context.Dogovor;

}

[HttpGet("{Номер\_договора}")]

public async Task<IActionResult> GetDogovor([FromRoute] int Номер\_договора)

{

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

var dogovor = await \_context.Dogovor.SingleOrDefaultAsync(m => m.Номер\_договора == Номер\_договора);

if (dogovor == null)

{

return NotFound();

}

return Ok(dogovor);

}

[HttpPost]

public async Task<IActionResult> Create([FromBody] Dogovor dogovor)

{

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

\_context.Dogovor.Add(dogovor);

await \_context.SaveChangesAsync();

return CreatedAtAction("GetDogovor", new { Номер\_договора = dogovor.Номер\_договора },

dogovor);

}

[HttpPut("{Номер\_договора}")]

public async Task<IActionResult> Update([FromRoute] int id,

[FromBody] Dogovor dogovor)

{

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

var item = \_context.Dogovor.Find(id);

if (item == null)

{

return NotFound();

}

item.Номер\_договора = dogovor.Номер\_договора;

\_context.Dogovor.Update(item);

await \_context.SaveChangesAsync();

return NoContent();

}

[HttpDelete("{id}")]

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

{

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

var item = \_context.Dogovor.Find(id);

if (item == null)

{

return NotFound();

}

\_context.Dogovor.Remove(item);

await \_context.SaveChangesAsync();

return NoContent();

}

}

}

using Microsoft.AspNetCore.Mvc;

using Microsoft.EntityFrameworkCore;

using System.Reflection.Metadata;

using ASPNetCoreApp.Models;

namespace WebAPI.Controllers

{

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

[ApiController]

public class DogovorsController : ControllerBase

{

private readonly OperatorContext \_context;

public DogovorsController(OperatorContext context)

{

\_context = context;

if (!\_context.Dogovor.Any())

{

\_context.Dogovor.Add(new Dogovor

{

Номер\_договора = 2,

Дата\_заключения = DateTime.Today,

Дата\_расторжения = DateTime.Today,

Код\_тарифа\_FK = 2,

Номер\_телефона = "000",

Номер\_клиента\_FK = '1',

Серийный\_номер\_сим\_карты = "000"

});

\_context.SaveChanges();

}

}

[HttpGet]

public IEnumerable<Dogovor> GetAll()

{

return \_context.Dogovor;

}

[HttpGet("{Номер\_договора}")]

public async Task<IActionResult> GetDogovor([FromRoute] int Номер\_договора)

{

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

var dogovor = await \_context.Dogovor.SingleOrDefaultAsync(m => m.Номер\_договора == Номер\_договора);

if (dogovor == null)

{

return NotFound();

}

return Ok(dogovor);

}

// POST: api/Dogovors

[HttpPost]

public async Task<ActionResult<Dogovor>> CreateDogovor(Dogovor dogovor)

{

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

\_context.Dogovor.Add(dogovor);

await \_context.SaveChangesAsync();

return CreatedAtAction("GetDogovor", new { id = dogovor.Номер\_договора }, dogovor);

}

[HttpPut("{Номер\_договора}")]

public async Task<IActionResult> Update([FromRoute] int id,

[FromBody] Dogovor dogovor)

{

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

var item = \_context.Dogovor.Find(id);

if (item == null)

{

return NotFound();

}

item.Номер\_договора = dogovor.Номер\_договора;

\_context.Dogovor.Update(item);

await \_context.SaveChangesAsync();

return NoContent();

}

[HttpDelete("{id}")]

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

{

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

var item = \_context.Dogovor.Find(id);

if (item == null)

{

return NotFound();

}

\_context.Dogovor.Remove(item);

await \_context.SaveChangesAsync();

return NoContent();

}

}

}