

**Controllers:**

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using Fogado.Models;

using Microsoft.EntityFrameworkCore;

namespace Fogado.Controllers

{

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

[ApiController]

public class foglalasok : ControllerBase

{

[HttpGet]

public IActionResult Get()

{

var contex = new FoglalasokContext();

try

{

return StatusCode(StatusCodes.Status200OK, contex.Foglalasoks.Include(f => f.VendegNavigation).Include(f => f.SzobaNavigation).ToList());

}

catch (Exception ex)

{

return StatusCode(StatusCodes.Status400BadRequest, ex.Message);

}

}

}

}

using Fogado.Models;

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

namespace Fogado.Controllers

{

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

[ApiController]

public class szoba : ControllerBase

{

[HttpGet("id")]

public IActionResult Get(int id)

{

var contex = new FoglalasokContext();

try

{

return StatusCode(StatusCodes.Status201Created, contex.Szobaks.Where(f => f.Szazon == id).ToList());

}

catch (Exception ex)

{

return StatusCode(StatusCodes.Status401Unauthorized, ex.Message);

}

}

[HttpPost]

public IActionResult Post(Szobak szoba)

{

var context = new FoglalasokContext();

try

{

context.Szobaks.Add(szoba);

context.SaveChanges();

return StatusCode(StatusCodes.Status202Accepted, "Új szoba sikeresen felvéve.");

}

catch (Exception ex)

{

return StatusCode(StatusCodes.Status402PaymentRequired, ex.Message);

}

}

[HttpPut]

public IActionResult Put(Szobak szoba)

{

var context = new FoglalasokContext();

try

{

context.Szobaks.Update(szoba);

context.SaveChanges();

return StatusCode(StatusCodes.Status203NonAuthoritative, "A szoba adatai módosítva.");

}

catch (Exception ex)

{

return StatusCode(StatusCodes.Status403Forbidden, ex.Message);

}

}

[HttpDelete("{id}")]

public IActionResult Delete(int id)

{

var context = new FoglalasokContext();

try

{

Szobak szoba = new Szobak();

szoba.Szazon = id;

context.Szobaks.Remove(szoba);

context.SaveChanges();

return StatusCode(StatusCodes.Status204NoContent, "Szoba sikeresen törölve.");

}

catch (Exception ex)

{

return StatusCode(StatusCodes.Status404NotFound, ex.Message);

}

}

}

}

1. Lekérdezés a Foglalasok táblára a kapcsolt táblázatokkal együtt.   
     
   Jó -200 Hiba -400  
     
   2. Szoba táblára lekérdezés azonosító alapján Jó -201 Hiba -401  
   3. Szoba táblára új szoba felvétele   
     
   Jó -202 Hiba -402  
   4. Szoba táblára létező szoba adatainak módosítása   
     
   Jó -203 Hiba -403  
   5. Szoba táblából törlés azonosító alapján   
     
   Jó -204 Hiba -404  
     
     
   Scaffold-DbContext "SERVER=localhost;PORT=3306;DATABASE=???????;USER=root;PASSWORD=;SSL MODE=none;" mysql.entityframeworkcore -outputdir Models -f

using System;

using System.Collections.Generic;

using System.Text.Json.Serialization;

namespace Fogado.Models;

public partial class Szobak

{

public int Szazon { get; set; }

public string Sznev { get; set; } = null!;

public int Agy { get; set; }

public int Potagy { get; set; }

[JsonIgnore]

public virtual ICollection<Foglalasok> Foglalasoks { get; set; } = new List<Foglalasok>();

}

using System;

using System.Collections.Generic;

namespace Fogado.Models;

public partial class Foglalasok

{

public int Fsorsz { get; set; }

public int Vendeg { get; set; }

public int Szoba { get; set; }

public DateTime Erk { get; set; }

public DateTime Tav { get; set; }

public int Fo { get; set; }

public virtual Szobak SzobaNavigation { get; set; } = null!;

public virtual Vendegek VendegNavigation { get; set; } = null!;

}

using System;

using System.Collections.Generic;

using System.Text.Json.Serialization;

namespace Fogado.Models;

public partial class Vendegek

{

public int Vsorsz { get; set; }

public string Vnev { get; set; } = null!;

public int Irsz { get; set; }

[JsonIgnore]

public virtual ICollection<Foglalasok> Foglalasoks { get; set; } = new List<Foglalasok>();

}