autoverseny.Models;

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using Microsoft.EntityFrameworkCore;

namespace autoverseny.Controllers

{

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

[ApiController]

public class PalyaController : ControllerBase

{

//Lekérdezés a Rendelések táblára a kapcsolt táblázatokkal együtt.

//Kategóriák táblára lekérdezés azonosító alapján

[HttpGet("id")]

public IActionResult Get(int id)

{

var contex = new VersenyContext();

try

{

return StatusCode(StatusCodes.Status200OK,contex.Palyas.Where(f=>f.Id==id).ToList());

}

catch (Exception ex)

{

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

}

}

//Kategóriák táblára új kategória felvétele

[HttpPost]

public IActionResult Post(Palya palya)

{

var context = new VersenyContext();

try

{

context.Palyas.Add(palya);

context.SaveChanges();

return StatusCode(StatusCodes.Status201Created, "Új pálya sikeresen felvéve.");

}

catch (Exception ex)

{

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

}

}

// Palya táblára létező pálya adatainak módosítása

[HttpPut]

public IActionResult Put(Palya palya)

{

var context = new VersenyContext();

try

{

context.Palyas.Update(palya);

context.SaveChanges();

return StatusCode(StatusCodes.Status202Accepted, "Pálya adatai módosítva.");

}

catch (Exception ex)

{

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

}

}

// Palya táblából törlés azonosító alapján

[HttpDelete("{id}")]

public IActionResult Delete(int id)

{

var context=new VersenyContext();

try

{

Palya palya = new Palya();

palya.Id = id;

context.Palyas.Remove(palya);

context.SaveChanges();

return StatusCode(StatusCodes.Status203NonAuthoritative, "Pálya sikeresen törölve.");

}

catch (Exception ex)

{

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

}

}

}

}

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using autoverseny.Models;

using Microsoft.EntityFrameworkCore;

namespace autoverseny.Controllers

{

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

[ApiController]

public class KoridoController : ControllerBase

{

[HttpGet]

public IActionResult Get()

{

var contex = new VersenyContext();

try

{

return StatusCode(StatusCodes.Status200OK, contex.Koridos.Include(f => f.Palya).Include(f=>f.Versenyzo.Csapat).ToList());

}

catch (Exception ex)

{

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

}

}

}

}

1. Lekérdezés a Korido táblára a kapcsolt táblázatokkal együtt.  
  
2. Palya táblára lekérdezés azonosító alapján  
3. Palya táblára új pálya felvétele  
4. Palya táblára létező pálya adatainak módosítása  
5. Palya táblából törlés azonosító alapján  
  
  
Scaffold-DbContext "SERVER=localhost;PORT=3306;DATABASE=???????;USER=root;PASSWORD=;SSL MODE=none;" mysql.entityframeworkcore -outputdir Models -f