Scaffold-DbContext "Data Source = mssql; Initial Catalog = ispp2111; User ID = ispp2111; Password = 2111; Trust Server Certificate=True" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models -ContextDir Data -Context GameContext -Tables Game, Category

builder.Services.AddDbContext<GameContext>();

var app = builder.Build();

Meth(ods):

1) GET

//endpoints eptah:

/api/users – список

/api/users/1 – оюъект

/api/users?param1=value1&param2=value&… - передача параметров

200 ok

404 not found

400 bad request

2) POST – создает объект, данные объекта передаются в теле запроса

POST можно использовать для авторизации

/api/users

201 created

Location /api/users/123

400

409 conflict

3) PUT / PATCH (PUT – замена ресурса(обновление всех полей); PATCH – заплатка (замена части полей))

/api/users

/api/users/1

204 no content – данные обновлены, ничего не возвращает

200 ok – для возврата замененного ресурса

400 – параметр говна

404 – не найдено

409 – отстань

4) DELETE

/api/users/1

204 no content

400

404

В C# доступны минимальные app (MapGet/MapPost/MapPut/MapPatch/MapDelete)

Ну или же можно создать контроллеры

Для проверки работы api можно использовать 1) swagger (до Dot.Net 9 встроенный, после нужно подключать) 2) postman, 3) консоль диспетчера пакетов

3) Invoke-RestMethod hhttp://localhost:5147/swagger/

Получение списка

app.MapGet("/categories", (GameContext context) =>

context.Categories.ToList()

);

Получение объекта

app.MapGet("/categories/{id:int}", (int id, GameContext context) =>

context.Categories.Find(id) is Category category ?

Results.Ok(category) : Results.NotFound()

);

Объявление объекта

app.MapPost("/categories", (Category category, GameContext context) =>

{

try

{

context.Categories.Add(category);

context.SaveChanges();

return Results.Created($"/categories/{category.CategoryId}", category);

}

catch

{

return Results.BadRequest();

}

}

);

Удаление объекта

app.MapDelete("/categories/{id:int}", (int id, GameContext context) =>

{

var category = context.Categories.Find(id);

if (category == null)

return Results.NotFound();

try

{

context.Categories.Remove(category);

context.SaveChanges();

return Results.NoContent();

}

catch

{

return Results.BadRequest();

}

}

);

Обновление объекта

app.MapPut("/categories/{id:int}", (int id, Category category, GameContext context) =>

{

/\*if (context.Categories.Count(c => c.CategoryId == id) == 0)

return Results.NotFound();\*/

if (category.CategoryId != id)

return Results.NotFound();

try

{

context.Categories.Update(category);

//context.Entry(category).State = Microsoft.EntityFrameworkCore.EntityState.Modified;

context.SaveChanges();

return Results.NoContent();

}

catch

{

return Results.BadRequest();

}

}

[JsonIgnore] –для того чтобы не передавать данные о теле запросов

Передача параметров адресной строке





