В контекст надо добавить сервисы

**GET**

// endpoints:

/api/users – список

/api/users/1 – объект

/api/users?param1=value1&param2=value2&… - передача параметров

200 ok

404 not found

400 bad request

**POST** – создает объект, данные объекта передаются в теле запроса, можно использовать для авторизации

/api/users

201 created

Location /api/users/123

400

409 confict

**PUT** – замена ресурса, то есть для обновления всех полей

**PATCH** – замена части полей

/api/users

/api/users/1

204 no content – данные обновили ничего не вернули

200 ok – для возврата обновленного ресурса

400

404

409

**DELETE**

/api/users/1

204 no content

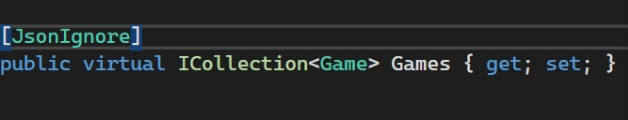
400

404

В C# доступен минимальный API

App: MapGet/MapPost/MapPut/MapPatch/MapDelete

[JsonIgnore] – для того что бы не передавать данные в теле запроса, указывать над полем в классе



Можно указывать над навигационными свойствами, что бы не было зацикливания

Для проверки работы API можно использовать:

1. Swagger
2. Postman

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

context.Categories.ToList()

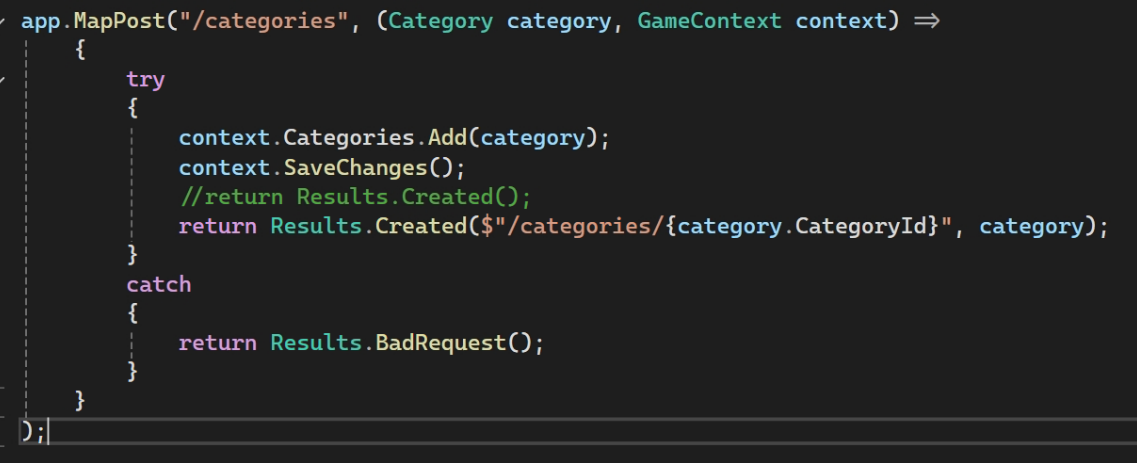
);

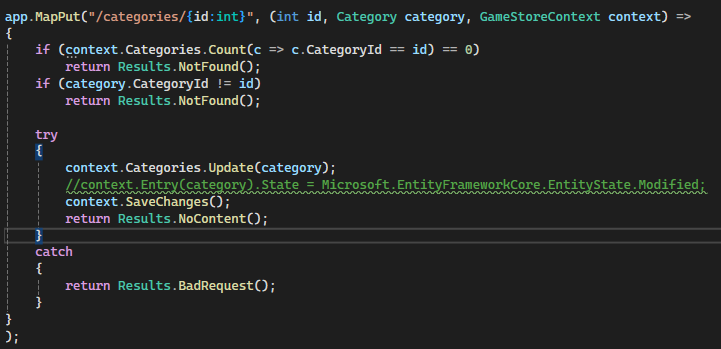
app.MapGet("/categories/{id:int}", (int id, GameStoreContext context) =>

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

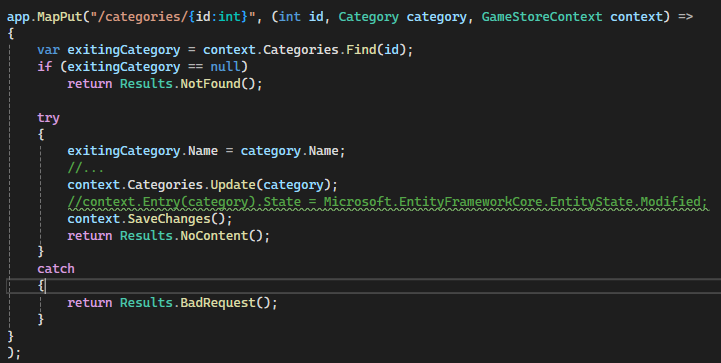
Results.Ok(category) : Results.NotFound()

);



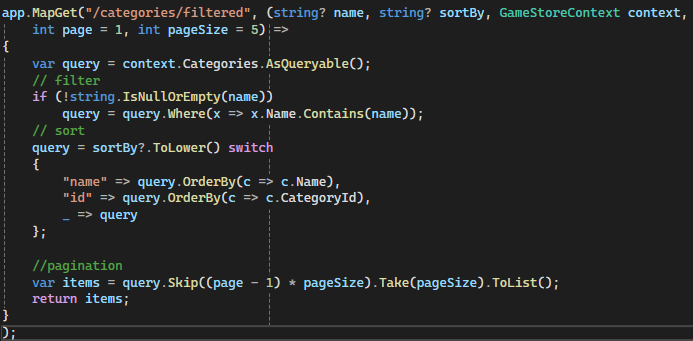


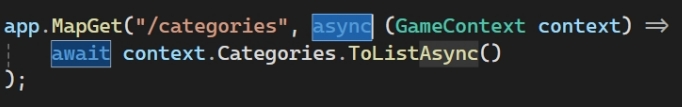
ИЛИ



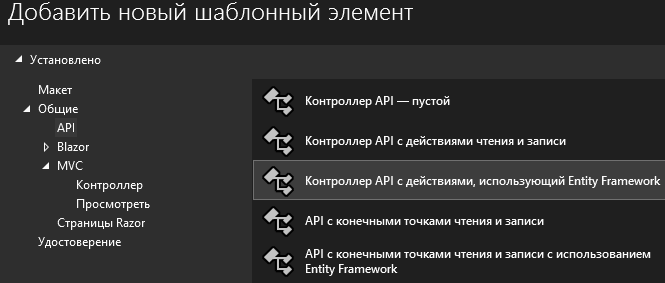
Передача параметров в адресной строке

/api/users?param1=value1&param2=value2&…



Async 

Для создания контроллеров

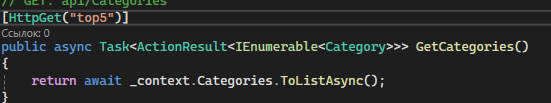
1. Создали проект с галочкой использовать контроллеры
2. Добавили классы в модели БД и контексты данных
3. Для генерации контроллера пкм по папке Controllers добавить контроллер
4. 

Методы контроллера могут вернуть

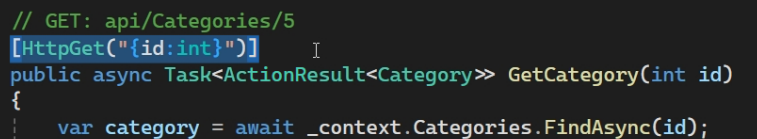
1. Объект простого типа
2. IActionResult – подходит для Delete и Put
3. ActionResult<T>/<Category> - GetById /<IEnumerable<Category>> - GetAll

Если тип возврата асинхронный все оборачивается в Task<…>

В маршруте API рекомендуется указывать версию 

Маршрут можно уточнить в описании типа метода 

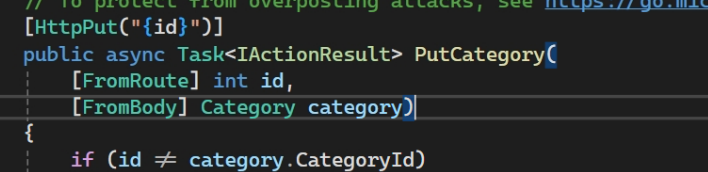
В маршрут можно передавать параметры, например id



Типы параметров методов для HTTP запросов

[FromRoute] int id – берется из Endpoint

[FromBody] Category category - берется из тела запроса, применяется для сложных объектов, данные обычно в JSON или XML



[FromQuery] string? sortBy, [FromQuery] int page=1 endpoint?sortBy=price&page=2 – берется из адресной строки после вопросительного знака