

Evidencias Pruebas unitarias y con Swagger Proyecto Prueba backend con .NET Core Web API

Autor: Reynol González Carballosa

Fecha: 22 de octubre 2024

Pruebas con Swagger

localhost:5266/swagger/index.html

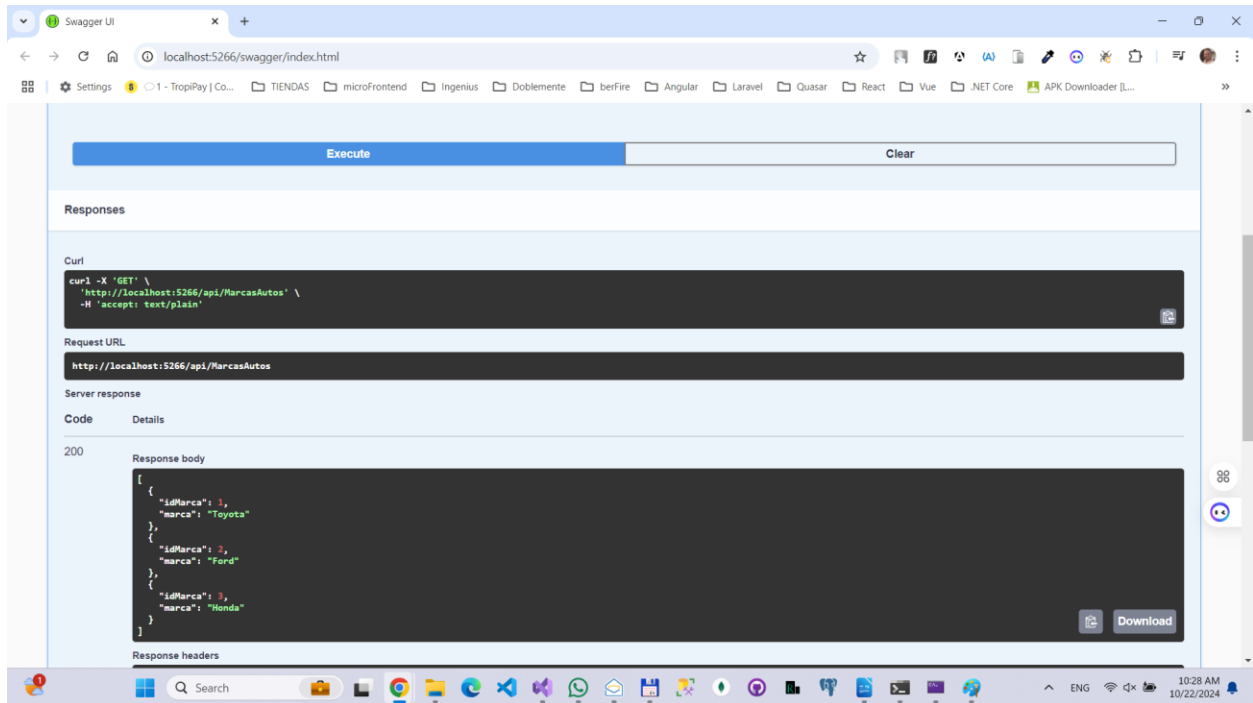
Swagger
Supported by SMARTBEAR

WebApplication1 1.0 OAS 3.0
<http://localhost:5266/swagger/v1/swagger.json>

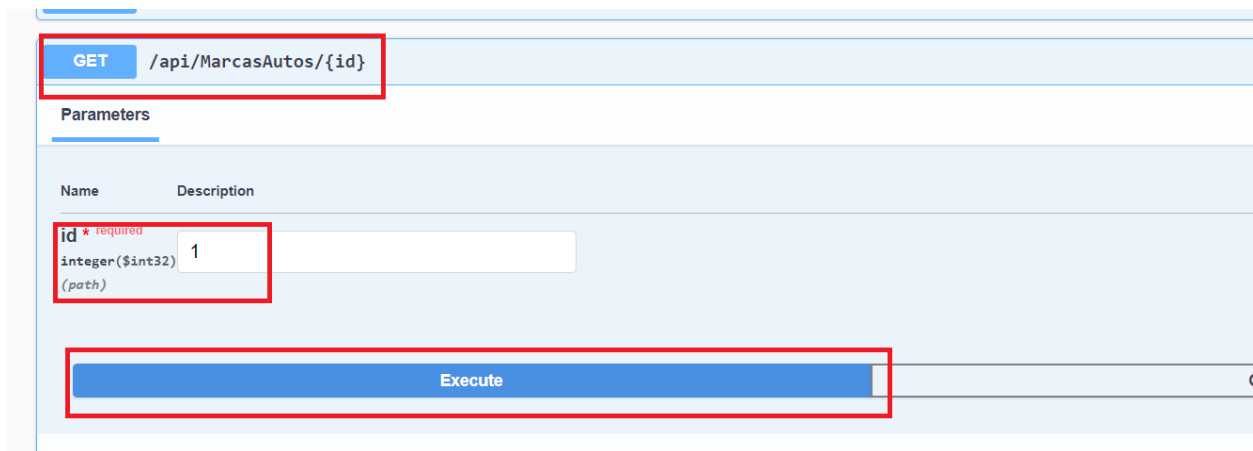
MarcasAutos

GET	/api/MarcasAutos
GET	/api/MarcasAutos/{id}

ENDPOINTS



Prueba de GET de una marca en específico



Respuesta de la Base de Datos

Server response	
Code	Details
200	<div>Response body</div> <pre>{ "idMarca": 1, "marca": "Toyota" }</pre>
Response headers	

Test Unitario, se preparó un entorno de prueba con una Base de Datos en memoria y se instancia el context con este ambiente para poder realizar los tests

0 references | 0 changes | 0 authors, 0 changes

```
public MarcasAutosControllerTests()
{
    var options = new DbContextOptionsBuilder<pruebaContext>()
        .UseInMemoryDatabase(databaseName: "TestDatabase")
        .Options;

    _context = new pruebaContext(options);

    //Inicializa datos de prueba
    var existe = _context.marcasAutos.FirstOrDefaultAsync().GetAwaiter().GetResult();
    //Si en la BD en memoria esta vacia la tabla marcasAuto
    //Entonces inserto un registro
    if (existe == null)
    {
        _context.marcasAutos.Add(new MarcasAutos { IdMarca = 1, Marca = "Toyota" });
        _context.SaveChanges();
    }
    //Creo la instancia del context con la Base de Datos en Memoria
    _controller = new MarcasAutosController(_context);
}
```

Se programaron 2 pruebas, la primera pasando un valor que no existe en la Base de Datos en memoria con valor IdMarca = 5. La segunda prueba es pasando un valor de IdMarca = 1 para validar la respuesta

```

[Fact]
0 references | 0 changes | 0 authors, 0 changes
public async Task GetMarcaAuto_ReturnsNotFound_WhenMarcaDoesNotExist()
{
    // Act
    var result = await _controller.GetMarcaAuto(5); // ID que no existe

    // Assert
    Assert.IsType<NotFoundResult>(result.Result);
}

```

```

[Fact]
0 references | 0 changes | 0 authors, 0 changes
public async Task GetMarcaAuto_ReturnsOkResult_WhenMarcaExists()
{
    //Act
    //Se ejecuta el método con el
    //controller instanciado a un
    //entorno de prueba con base de Datos en Memoria
    var result = await _controller.GetMarcaAuto(1); // ID que existe en la BD en memoria

    // Se comprueba el tipo devuelto por el método GetMarcaAuto
    Assert.IsType<ActionResult<MarcasAutos>>(result);
    //Se comprueba si el valor devuelto en la propiedad IdMarca es igual al esperado, o sea, "1"
    Assert.Equal("1",result?.Value?.IdMarca.ToString());
}

```

Starting test execution, please wait...

A total of 1 test files matched the specified pattern.

Passed! - Failed: 0, Passed: 2, Skipped: 0, Total: 2, Duration: 7 ms - TestProject1.dll (net6.0)
PM>