AutoDB.Mongo

Nuget Package for .Net 6

Enable easy connection with MongoDb

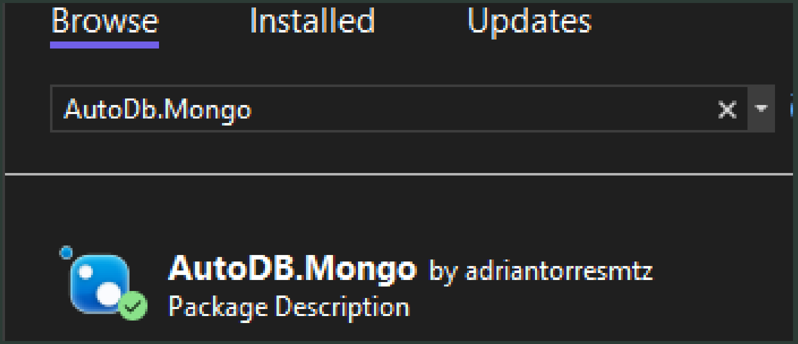
How to implement

Autor

Adrian Torres Mtz

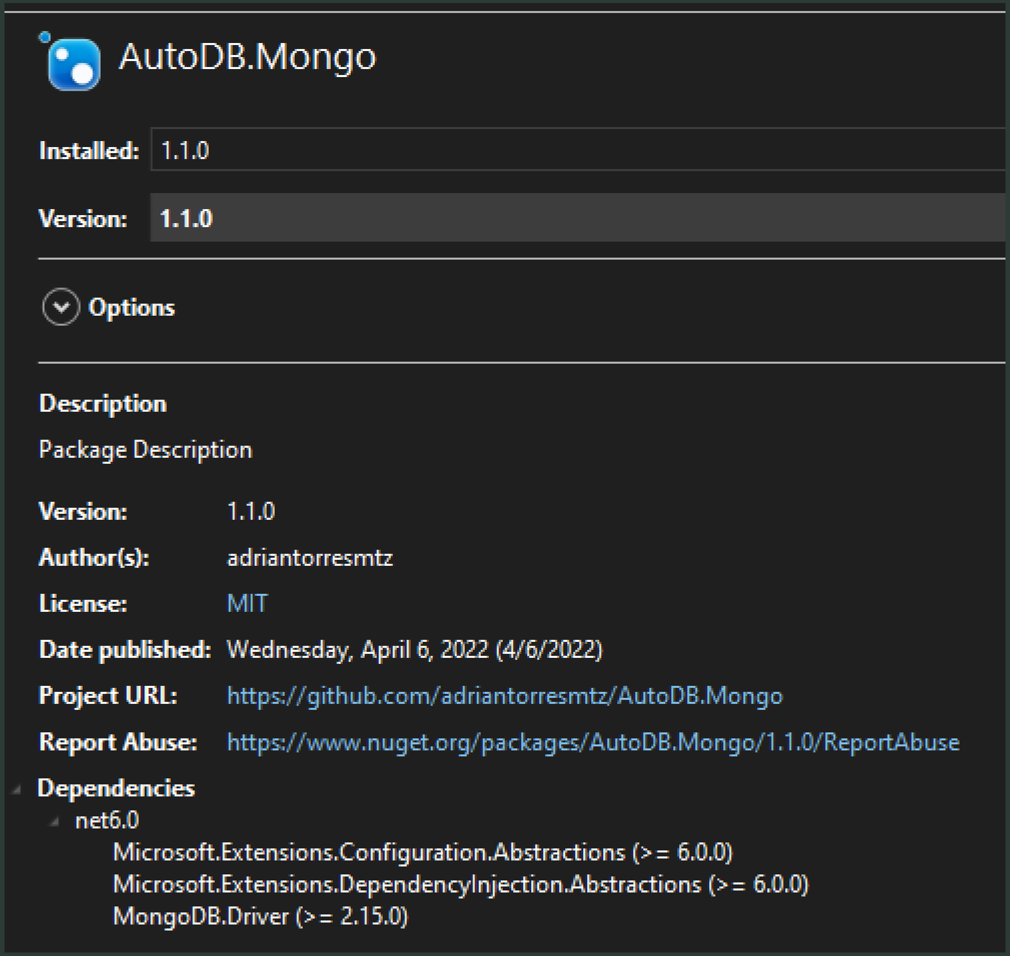
How Install AutoDB.Mongo

From NuGet Package Management



AutoDb.Mongo

AutoDB.Mongo by adriantorresmtz
Package Description



AutoDB.Mongo

Installed: 1.1.0

Version: 1.1.0

Description

Package Description

Version:	1.1.0
Author(s):	adriantorresmtz
License:	MIT
Date published:	Wednesday, April 6, 2022 (4/6/2022)
Project URL:	https://github.com/adriantorresmtz/AutoDB.Mongo
Report Abuse:	https://www.nuget.org/packages/AutoDB.Mongo/1.1.0/ReportAbuse

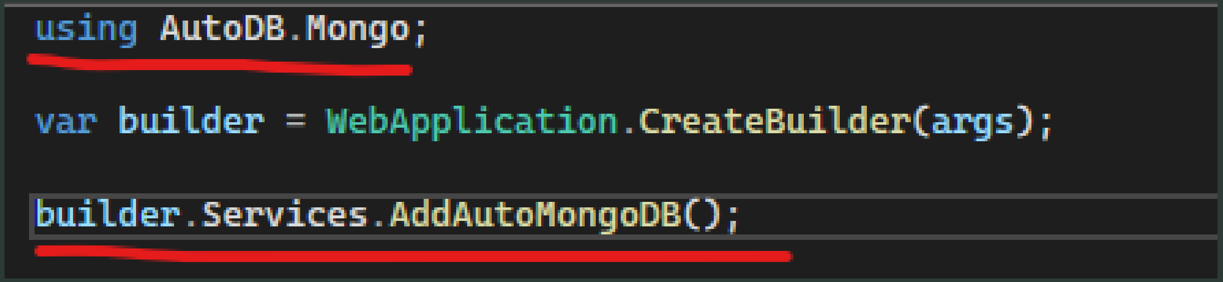
Dependencies

- net6.0
  - Microsoft.Extensions.Configuration.Abstractions (>= 6.0.0)
  - Microsoft.Extensions.DependencyInjection.Abstractions (>= 6.0.0)
  - MongoDB.Driver (>= 2.15.0)

AutoDB.Mongo

Add Dependency Injection

In Program.cs add these lines



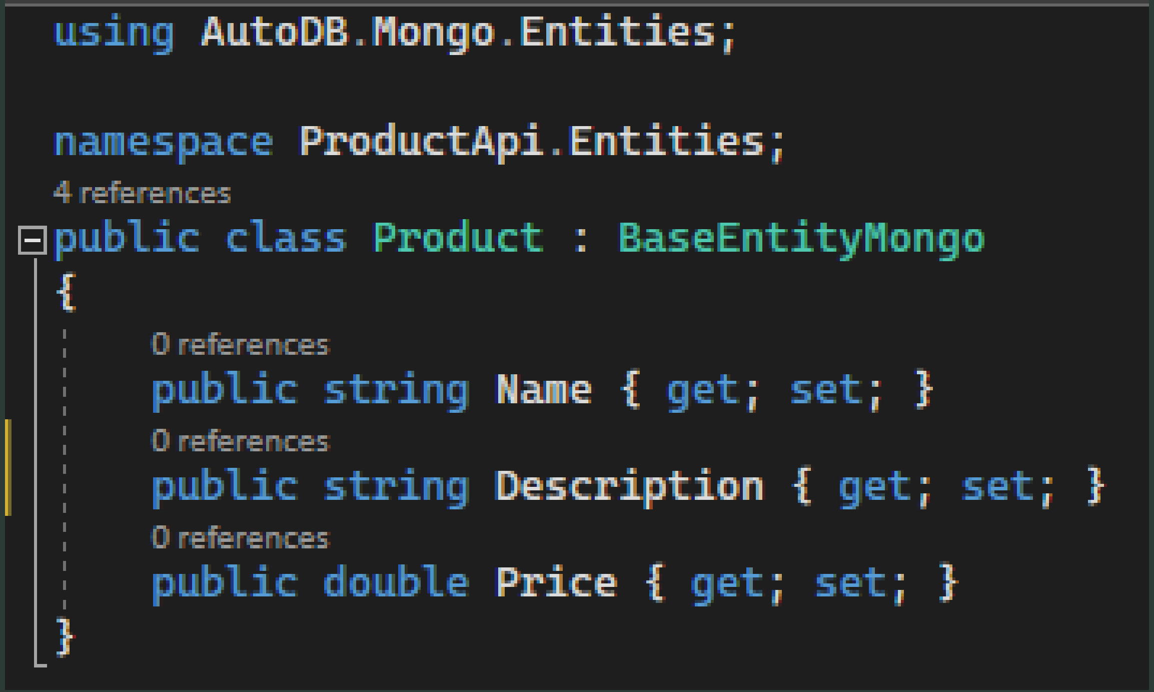
```
using AutoDB.Mongo;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddAutoMongoDB();
```

Create an AutoDB.Mongo Entity

To declare an entity as AutoDB.Mongo Entity we need to inheritance from `AutoDB.Mongo.Entities.BaseEntityMongo`

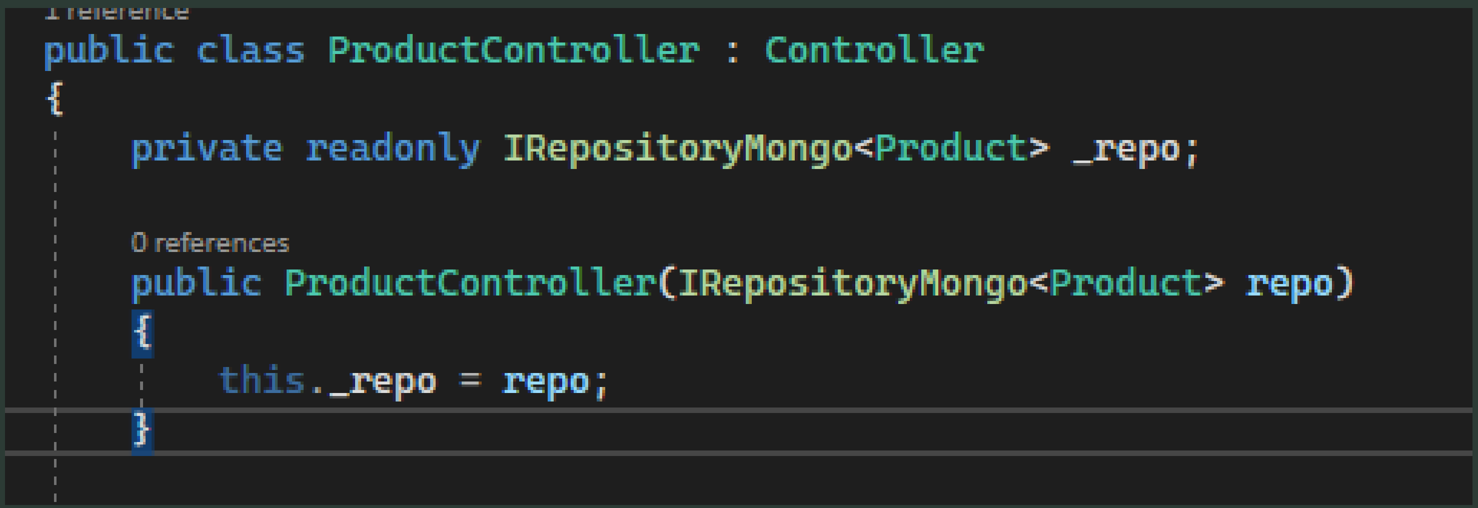


```
using AutoDB.Mongo.Entities;

namespace ProductApi.Entities;
4 references
public class Product : BaseEntityMongo
{
    0 references
    public string Name { get; set; }
    0 references
    public string Description { get; set; }
    0 references
    public double Price { get; set; }
}
```

Inject the IRepositoryMongo

To make a call to our Repository MongoDB we have to inject in the Constructor IRepositoryMongo<TEntity>



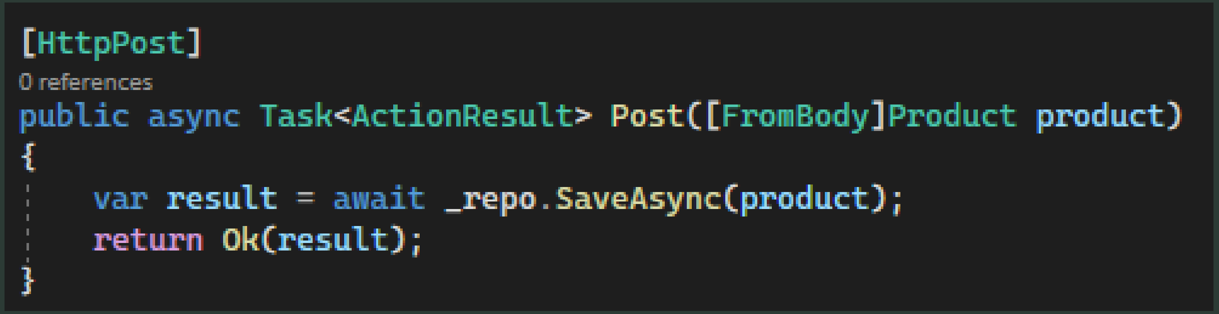
```
1 reference
public class ProductController : Controller
{
    private readonly IRepositoryMongo<Product> _repo;

    0 references
    public ProductController(IRepositoryMongo<Product> repo)
    {
        this._repo = repo;
    }
```

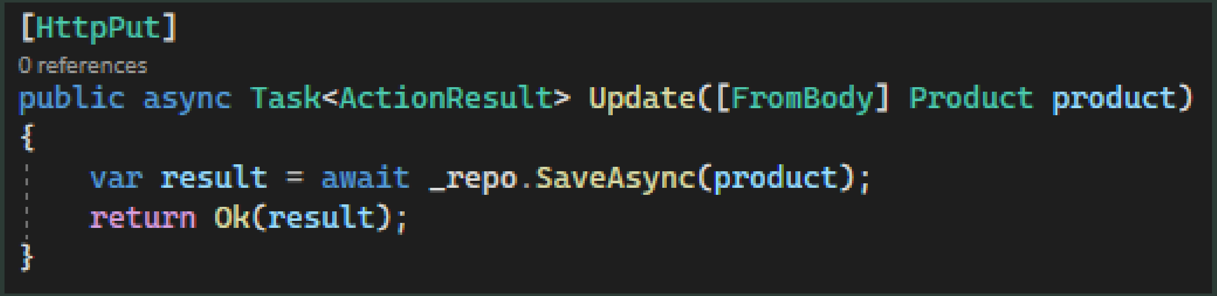
How works SaveAsync (Post / Put)

AutoDB.Mongo

To create a new document in our collection <TEntity> we need to make a call to SaveAsync with an object of <TEntity>



```
[HttpPost]
0 references
public async Task<ActionResult> Post([FromBody]Product product)
{
    var result = await _repo.SaveAsync(product);
    return Ok(result);
}
```



```
[HttpPut]
0 references
public async Task<ActionResult> Update([FromBody] Product product)
{
    var result = await _repo.SaveAsync(product);
    return Ok(result);
}
```

By default <TEntity> has Id as String Guid autogenerated handle by AutoDB.Mongo

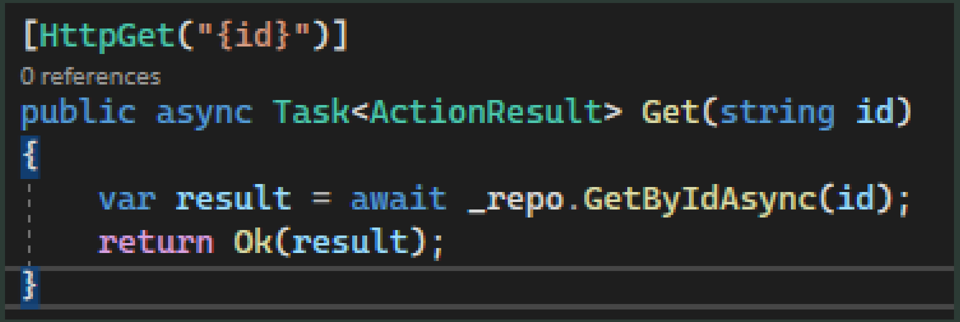
If Id in <TEntity> is null AutoDB.Mongo creates a new Guid then handles as new document

If Id in <TEntity> is Not null AutoDB.Mongo handles this object as an existing document

How works GetByIdAsync

AutoDB.Mongo

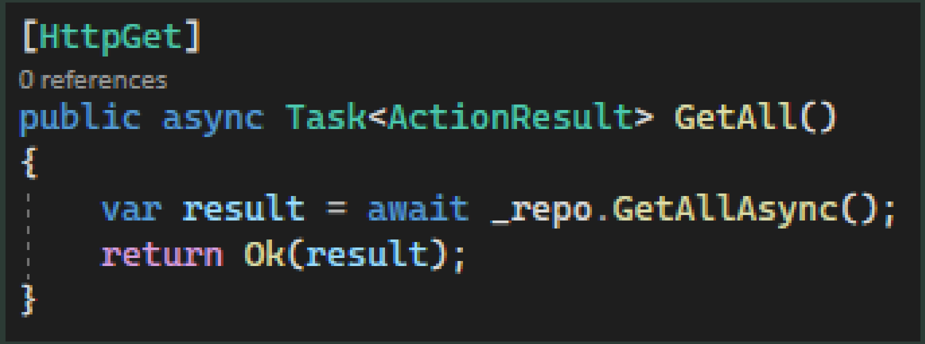
To get a document by id from our collection <TEntity> we need to make a call to GetByIdAsync with Id as string



```
[HttpGet("{id}")]
0 references
public async Task<ActionResult> Get(string id)
{
    var result = await _repo.GetByIdAsync(id);
    return Ok(result);
}
```

How works GetAllAsync

To get all documents from our collection <TEntity> we need to make a call to GetAllAsync

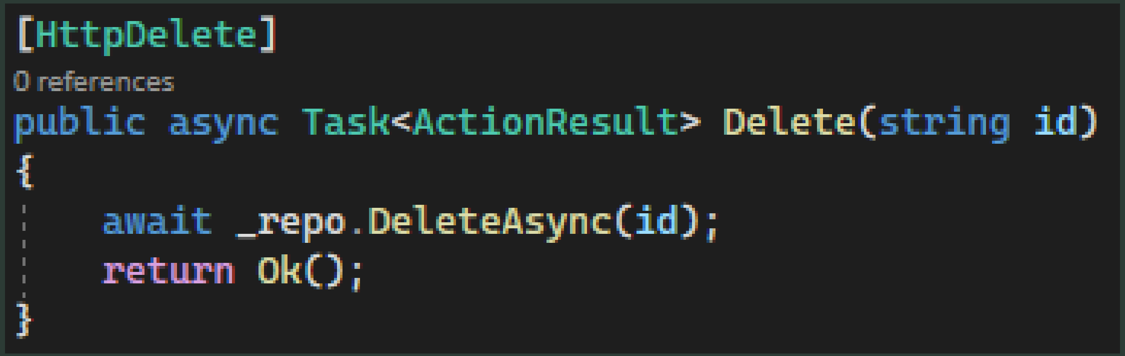


```
[HttpGet]
0 references
public async Task<ActionResult> GetAll()
{
    var result = await _repo.GetAllAsync();
    return Ok(result);
}
```

How works DeleteAsync

AutoDB.Mongo

To delete single document from our collection <TEntity> we need to make a call to DeleteAsync with Id as string



```
[HttpDelete]
0 references
public async Task<ActionResult> Delete(string id)
{
    await _repo.DeleteAsync(id);
    return Ok();
}
```