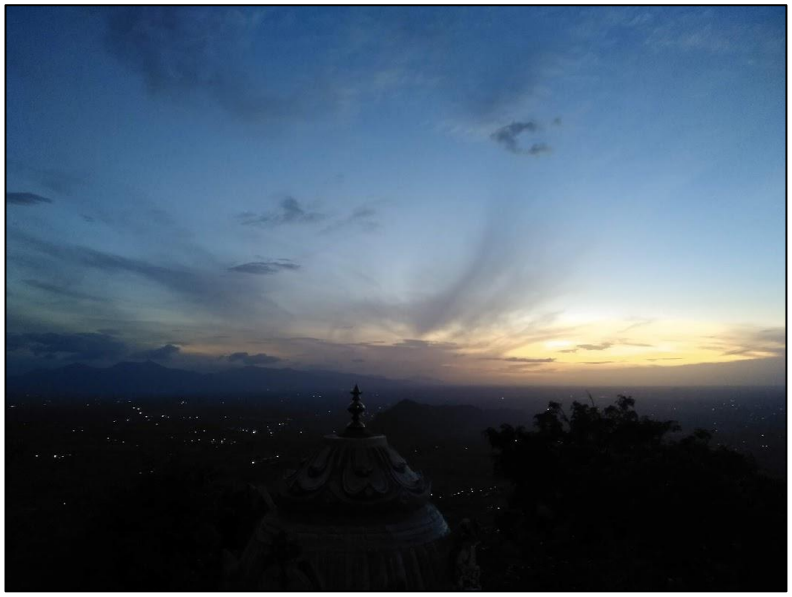


CREATE A WEB API WITH ASP.NET CORE AND MONGODB

DISCLAIMER

The material presented here is not original. It might point to several websites. This paper clarifies how MongoDB and the ASP.NET Core Web API interact. Original images extracted from the author's source code while running the project. All software's are community edition.

Vadivel Rangasamy
+91 9787778365

Table of Contents

About this Document

In this document help to Creates a web API that runs Create, Read, Update, and Delete (CRUD) operations on a MongoDB NoSQL database.

Introduction

MongoDB

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server-Side Public License which is deemed non-free by several distributions.[1]

MongoDB download link

https://www.mongodb.com/try/download/community?tck=docs_server

Install MongoDB Community Edition on Windows

https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-windows/

Visual Studio 2022

Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.[2]

Visual Studio download link

https://visualstudio.microsoft.com/vs/

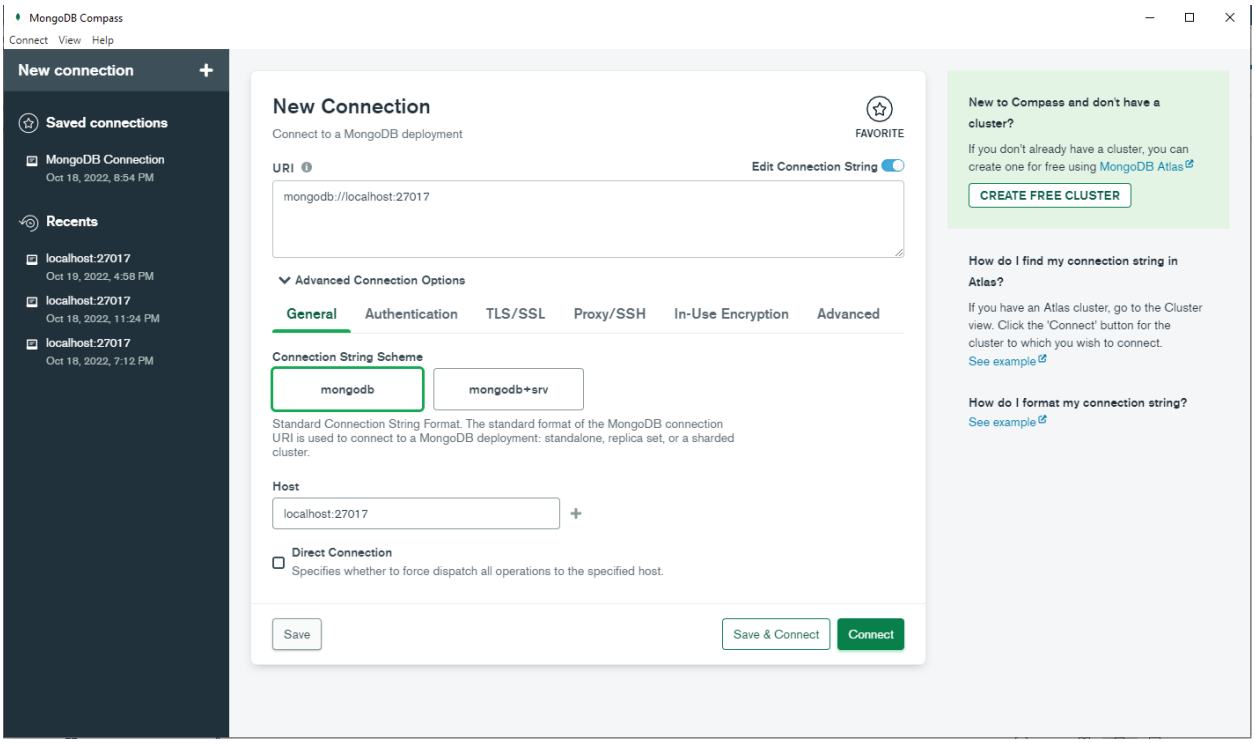
You can follow the above URLs instruction to install the software to your PC. The current experimental system has installed the MongoDB and VS2022 software's to Windows 10 operating system.

[1] https://www.mongodb.com/what-is-mongodb
[2] https://visualstudio.microsoft.com/vs/

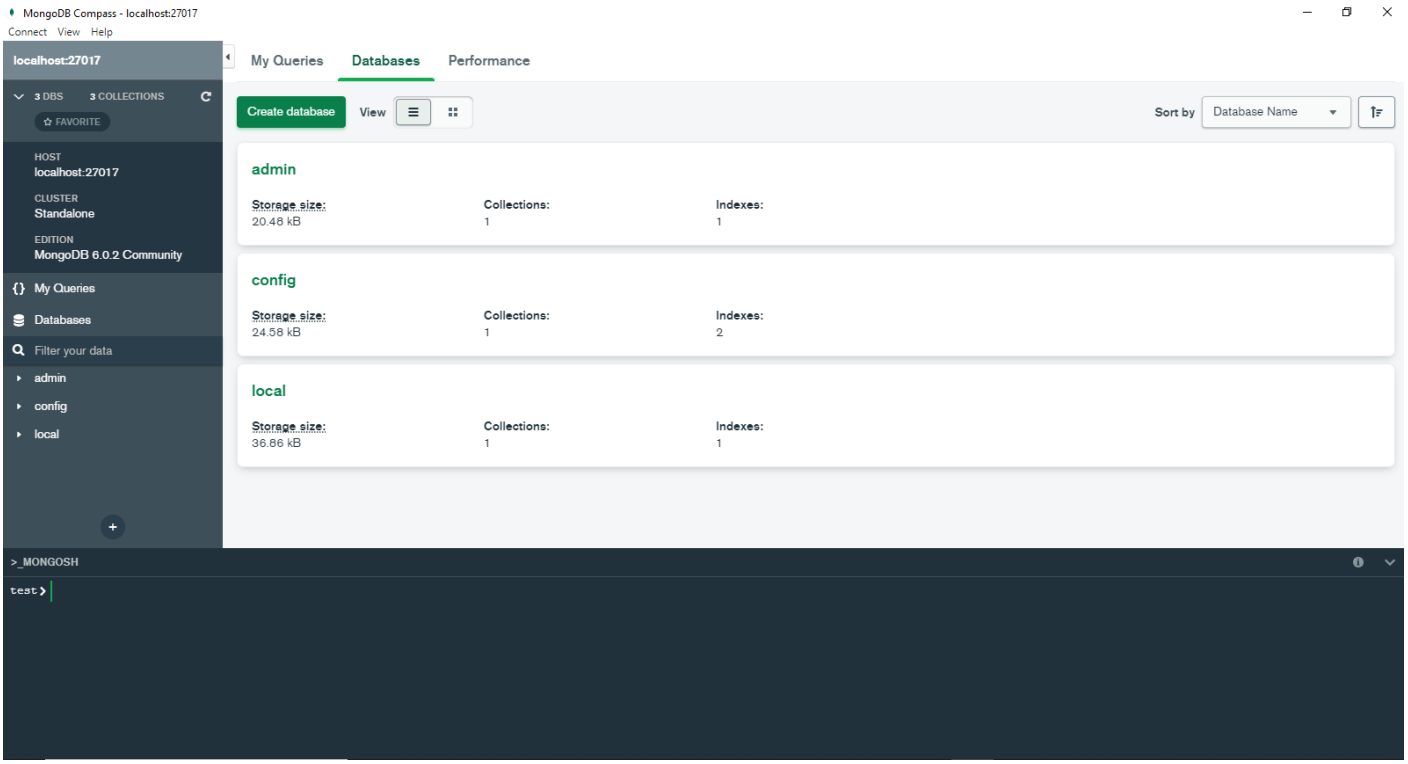
Create a database to MongoDB

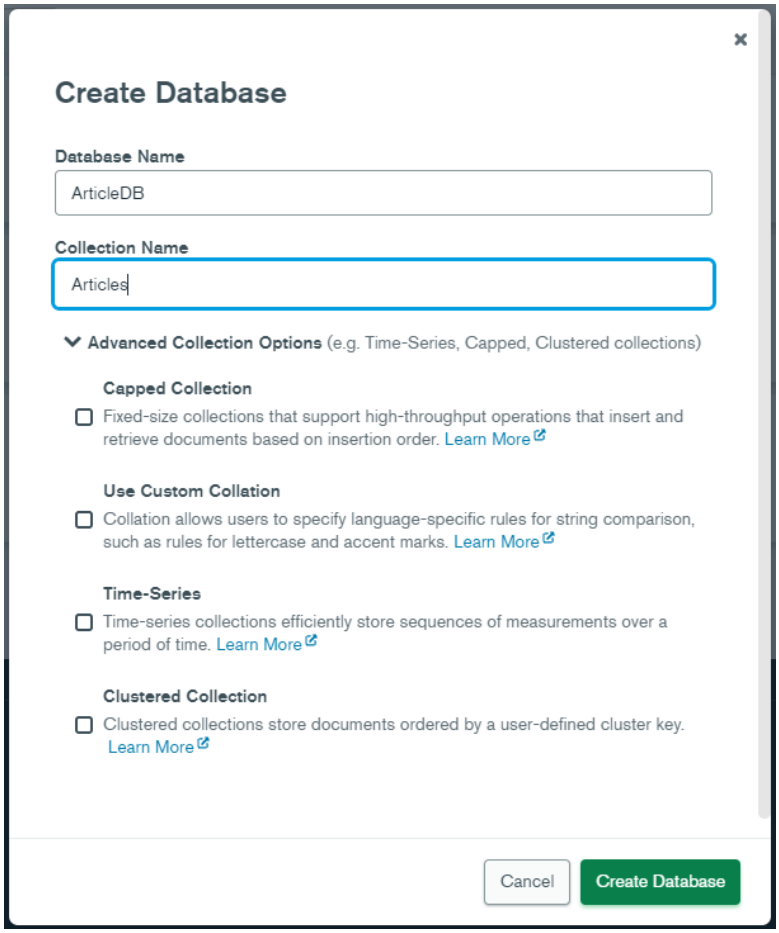
Most of the MongoDB commands are recommended to use Ms-DOS prompt. You can install additionally MongoDB Compass IDE to visually execute the MongoDB commands.

The MongoDB default logical port is 27017 (This logical port may be differed system to system). Use “Connect” button to connect the MongoDB.

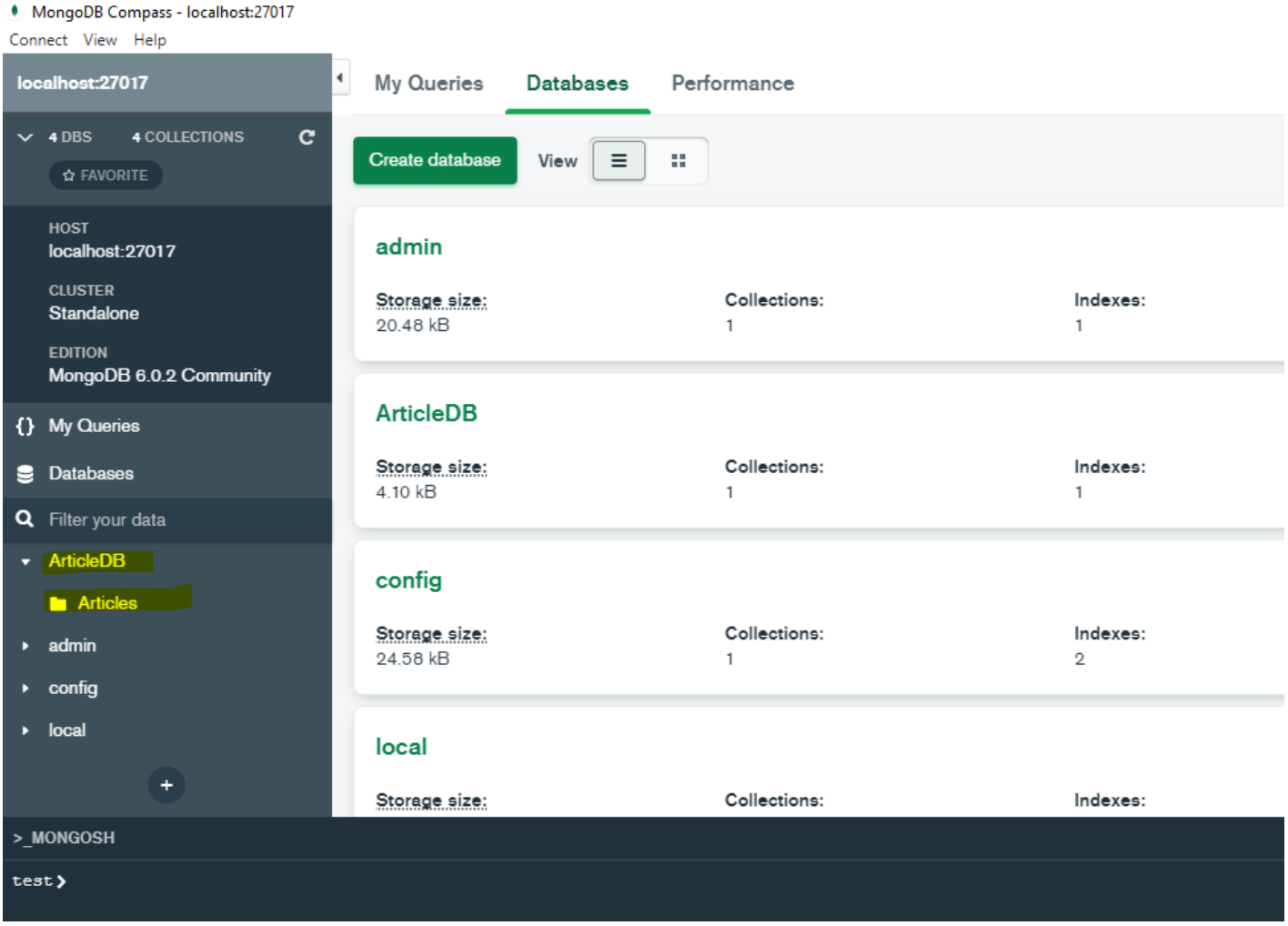


Once you successfully connected then you landed on the “Database” tab. Use “Create database” button or Shell prompt to create MongoDB database and collection. In this example “ArticleDB” is database and “Articles” is collection.





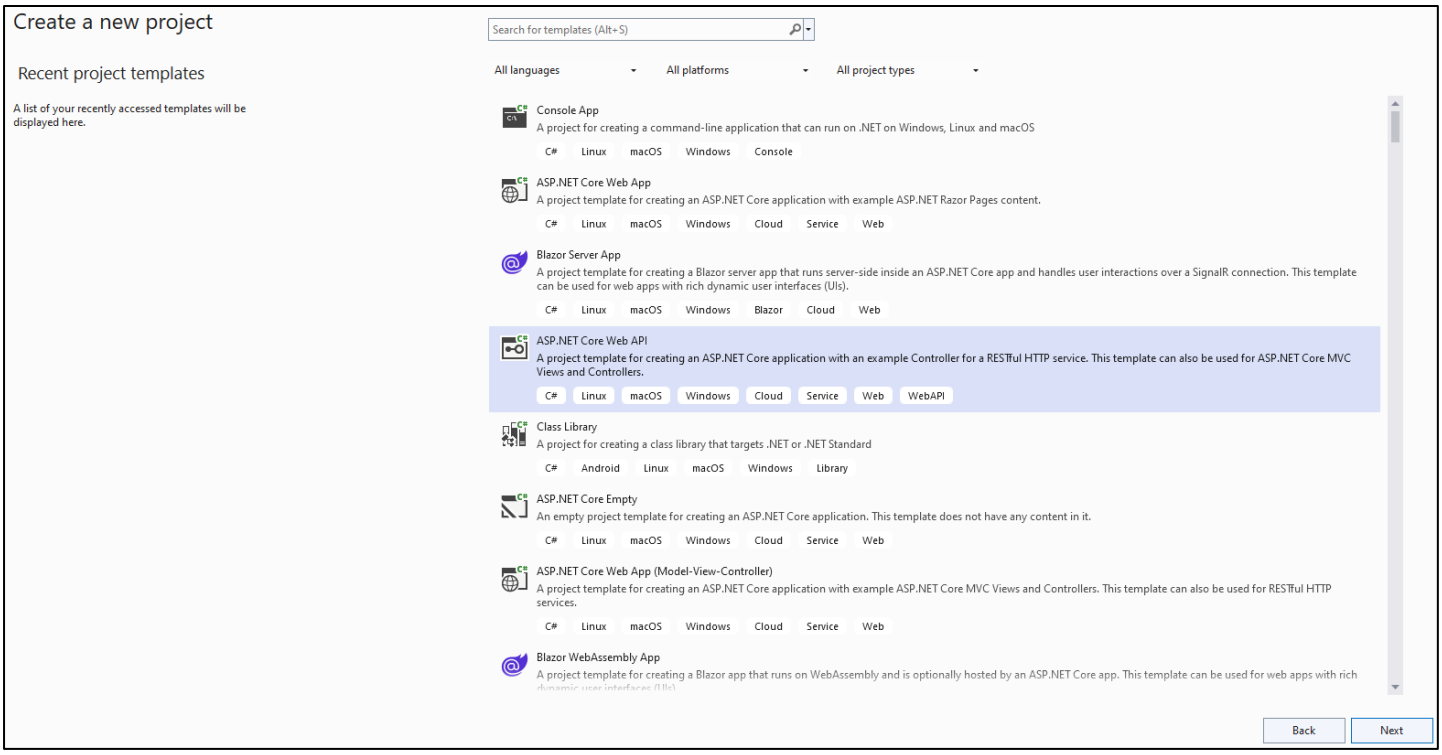
Database and collection have been successfully created

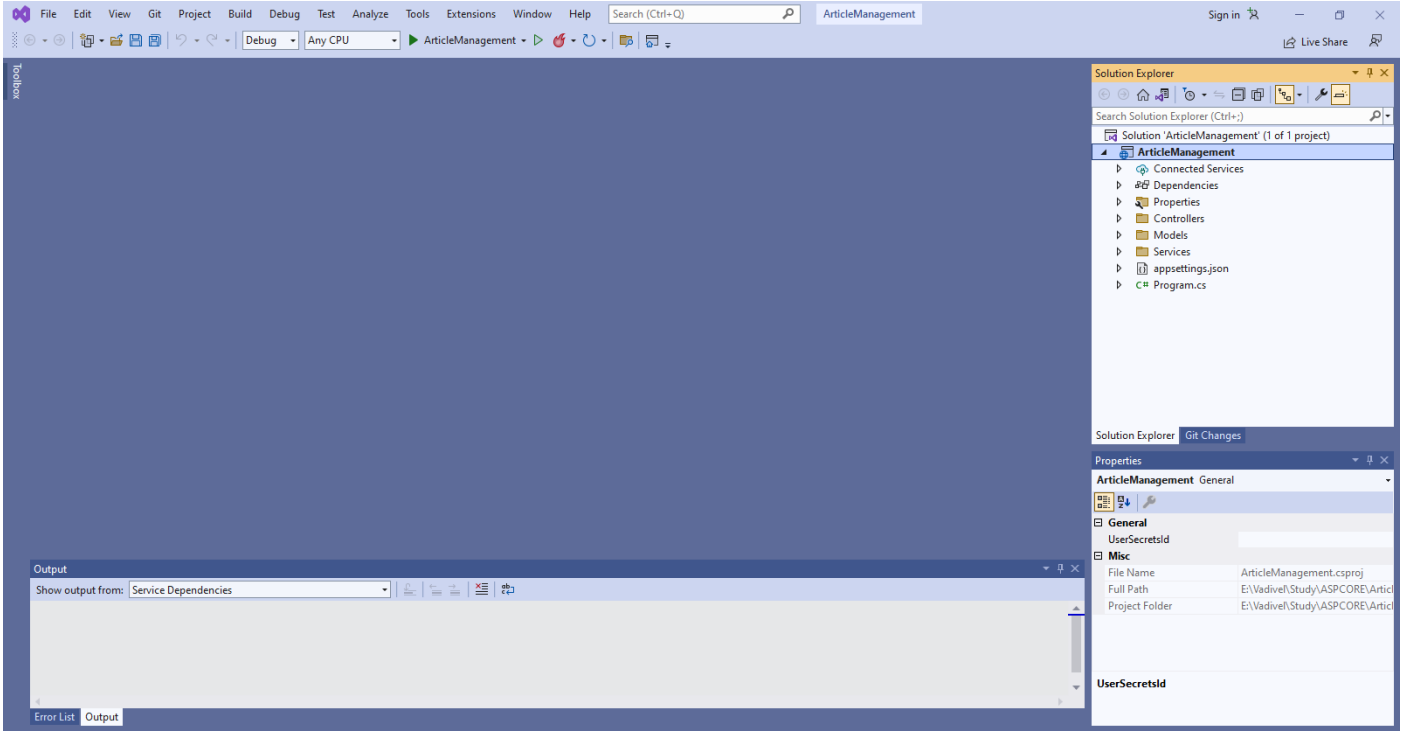


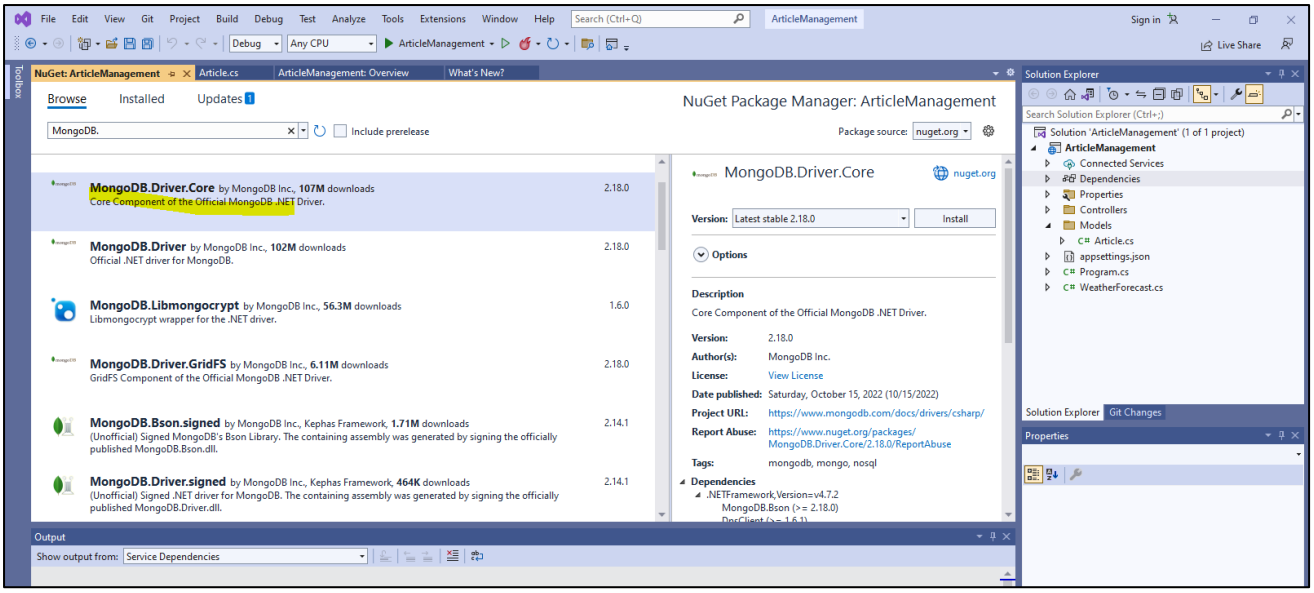
Create the ASP.NET Core web API project

Once you successfully installed the Visual Studio 2022 then you can see the below screenshot you select the **"ASP.NET Core Web API"**

1. Go to File > New > Project.
2. Select the ASP.NET Core Web API project type and select Next.
3. Name the project "ArticleManagement" and select Next.
4. Select the .NET 6.0 (Long-term support) framework and select Create.
5. Install the .NET driver for MongoDB to Visual studio 2022 project with help of "NuGet"







Add an entity model

1. Add a Models directory to the project root
2. Add a "Article" class to the Models directory with the following code

```
using MongoDB.Bson;
using MongoDB.Bson.Serialization.Attributes;
namespace ArticleManagement.Models
{
    10 references
    public class Article
    {
        [BsonId]
        [BsonRepresentation(BsonType.ObjectId)]
        6 references
        public string? ArticleId { get; set; }

        [BsonElement("ArticleName")]
        0 references
        public string ArticleName { get; set; } = null!;
        0 references
        public string Description { get; set; } = null!;
        0 references
        public string Author { get; set; } = null!;
    }
}
```

In the preceding class, the ArticleId property is:

- Required for mapping the Common Language Runtime (CLR) object to the MongoDB collection.
- Annotated with [BsonId] to make this property the document's primary key.
- Annotated with [BsonRepresentation(BsonType.ObjectId)] to allow passing the parameter as type string instead of an ObjectId structure. Mongo handles the conversion from string to ObjectId.

The ArticleName property is annotated with the [BsonElement] attribute. The attribute's value of Name represents the property name in the MongoDB collection.

Add a configuration model

1. Add the following database configuration values to appsettings.json file instead of “web.config” in traditional ASP.NET.

```
appsettings.json
Schema: https://json.schemastore.org/appsettings.json
{
  "ArticleDatabase": {
    "ConnectionString": "mongodb://localhost:27017",
    "DatabaseName": "AricleDB",
    "ArticleCollectionName": "Articles"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*"
}
```

2. Add a “ArticleDatabaseSettings” class to the Models directory with the following code

```
namespace ArticleManagement.Models
{
    2 references
    public class ArticleDatabaseSettings
    {
        0 references
        public string ConnectionString { get; set; } = null!;

        0 references
        public string DatabaseName { get; set; } = null!;

        0 references
        public string ArticleCollectionName { get; set; } = null!;
    }
}
```

The preceding "ArticleDatabaseSettings" class is used to store the appsettings.json file's "ArticleDatabase" property values. The JSON and C# property names are named identically to ease the mapping process.

3. Add the following highlighted code to Program.cs

```
ArticleManagement
using ArticleManagement.Models;
using ArticleManagement.Services;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.Configure<ArticleDatabaseSettings>(
    builder.Configuration.GetSection("ArticleDatabase"));
```

In the preceding code, the configuration instance to which the appsettings.json file's "ArticleDatabase" section binds is registered in the Dependency Injection (DI) container. For example, the "ArticleDatabaseSettings" object's ConnectionString property is populated with the ArticleDatabase:ConnectionString property in appsettings.json.

4. Add the following code to the top of Program.cs to resolve the ArticleDatabaseSettings. In Visual studio 2022 automatically add the namespace.

```
ArticleManagement
1   using ArticleManagement.Models;
```

Add a CRUD operations service

1. Add a Services directory to the project root.
2. Add a “ArticleService” class to the Services directory with the following code

```
ArticleManagement | ArticleManagement.Services.ArticleService | ArticleService(IOptions<ArticleDatabaseSettings> articleDatabaseSetting
4   using MongoDB.Driver;
5   namespace ArticleManagement.Services
6   {
        4 references
7       public class ArticleService
8       {
9           private readonly IMongoCollection<Article> _articlesCollection;
10
            0 references
11          public ArticleService(
12          IOptions<ArticleDatabaseSettings> articleDatabaseSettings)
13          {
14              bool isMongoLive = MongoDBConnectionProbe("mongodb://localhost:27017", "ArticleDB"); //database.RunCommandAsync((Command<BsonDocument>)"{ping:1}").Wait(1000);
15              if (isMongoLive)
16              {
17                  var mongoClient = new MongoClient("mongodb://localhost:27017"); //(articleDatabaseSettings.Value.ConnectionString);
18                  var mongoDatabase = mongoClient.GetDatabase("ArticleDB"); //(articleDatabaseSettings.Value.DatabaseName);
19                  _articlesCollection = mongoDatabase.GetCollection<Article>("Articles"); //(articleDatabaseSettings.Value.ArticleCollectionName);
20              }
21              else
22              {
23                  // couldn't connect
24              }
25          }
26
            1 reference
27          public async Task<List<Article>> GetAsync() =>
28          await _articlesCollection.Find(_ => true).ToListAsync();
```

3. Add the following highlighted code to Program.cs

```
ArticleManagement
1   using ArticleManagement.Models;
2   using ArticleManagement.Services;
3
4   var builder = WebApplication.CreateBuilder(args);
5
6   // Add services to the container.
7   builder.Services.Configure<ArticleDatabaseSettings>(
8       builder.Configuration.GetSection("ArticleDatabase"));
9
10  builder.Services.AddSingleton<ArticleService>();
```

4. Add the following code to the top of Program.cs to resolve the “ArticleService” reference. In Visual studio 2022 automatically add the namespace.

```
ArticleManagement
1   using ArticleManagement.Models;
2   using ArticleManagement.Services;
3
```

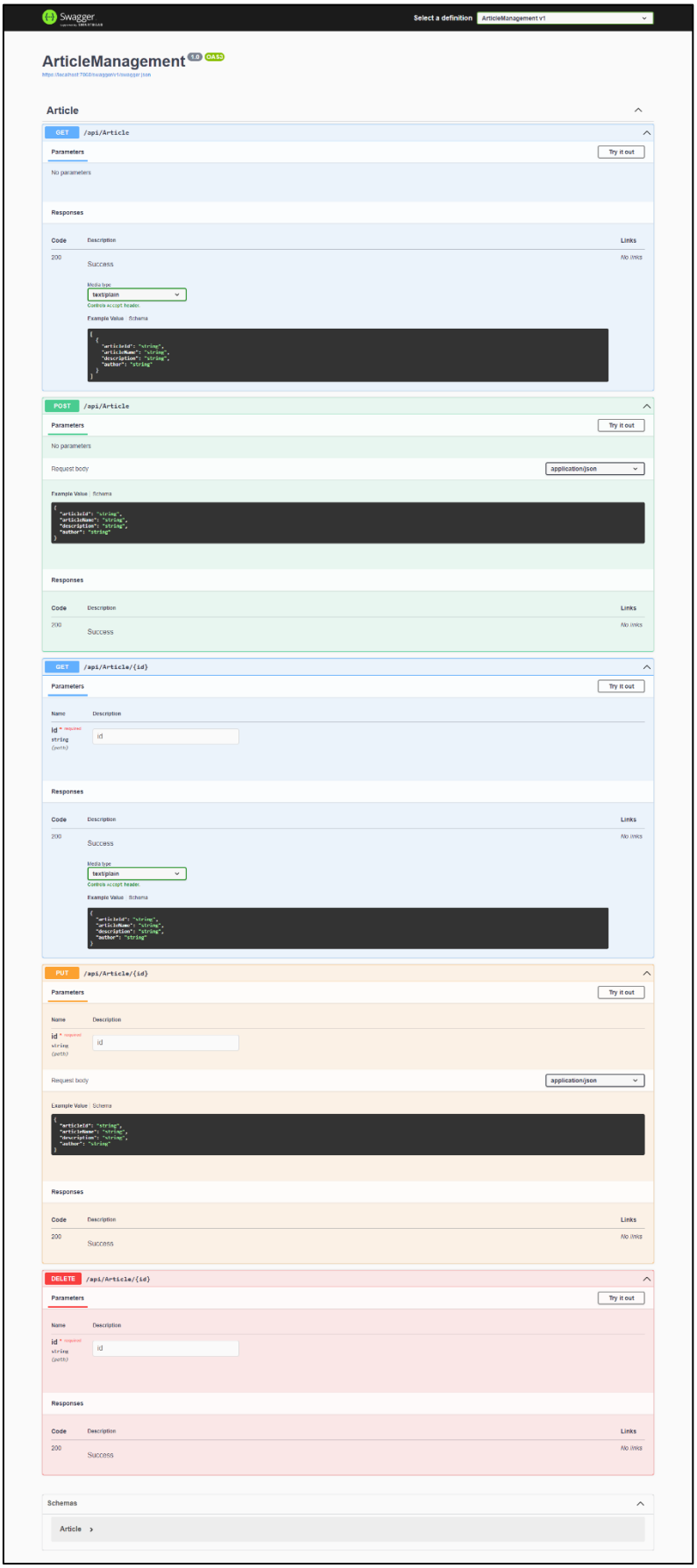
The "ArticleService" class uses the following MongoDB.Driver members to run CRUD operations against the database

Add a controller

1. Add a "ArticleController" class to the Controllers directory with the following code

```
ArticleManagement                                   ArticleManagement.Contro
1   using ArticleManagement.Models;
2   using ArticleManagement.Services;
3   using Microsoft.AspNetCore.Mvc;
4
5   namespace ArticleManagement.Controllers
6   {
7       [ApiController]
8       [Route("api/[controller]")]
        1 reference
9       public class ArticleController: ControllerBase
10      {
11          private readonly ArticleService _articleService;
12
            0 references
13          public ArticleController(ArticleService articleService) =>
14              _articleService = articleService;
15
16          [HttpGet]
            1 reference
17          public async Task<List<Article>> Get() =>
18
19              await _articleService.GetAsync();
20
21          [HttpGet("{id:length(24)}")]
            1 reference
22          public async Task<ActionResult<Article>> Get(string id)
23          {
24              var article = await _articleService.GetAsync(id);
25
26              if (article is null)
27              {
28                  return NotFound();
29              }
30
31              return article;
32          }
33
```

Test the web API



Create / Insert

URL: https://localhost:7068/api/article (The URL may be differed system to system)

POST Insert Article

APF Beta2 Mobile App API

Insert Article

Examples 0 BUILD

POST https://localhost:7068/api/article Send Save

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
{
    "articleId": "634edf085eeec043ecb385ae",
    "articleName": "Elcom",
    "description": "We help organisations build world-class intranet, portal, website, learning management and digital workplace solutions - giving
        staff the tools they need to deliver exceptional digital customer and employee experiences.",
    "author": "Vadivel"
}
```

Body Cookies Headers (5) Test Results Status: 201 Created Time: 1884 ms Size: 541 B Save Response

Pretty Raw Preview Visualize JSON

```
{
    "articleId": "634edf085eeec043ecb385ae",
    "articleName": "Elcom",
    "description": "We help organisations build world-class intranet, portal, website, learning management and digital workplace solutions - giving
        staff the tools they need to deliver exceptional digital customer and employee experiences.",
    "author": "Vadivel"
}
```

MongoDB Compass - localhost:27017/ArticleDB.Articles

Connect View Collection Help

localhost:27017

4 DBS 4 COLLECTIONS

FAVORITE

HOST
localhost:27017

CLUSTER
Standalone

EDITION
MongoDB 6.0.2 Community

My Queries

Databases

Filter your data

ArticleDB

Articles

admin

config

local

Documents
ArticleDB.Articles

ArticleDB.Articles

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

ADD DATA VIEW

```
_id: ObjectId('634edf085eeec043ecb385ae')
ArticleName: "Elcom"
Description: "We help organisations build world-class intranet, portal, website, lea…"
Author: "Vadivel"

_id: ObjectId('634edf085eeec043ecb385aa')
ArticleName: "MongoDB"
Description: "MongoDB is a source-available cross-platform document-oriented databas…"
Author: "Vadivel Rangasamy"

_id: ObjectId('634edf085eeec043ecb385ab')
ArticleName: "VisualStudio"
Description: "Visual Studio is an integrated development environment (IDE) from Micr…"
Author: "Vadivel Rangasamy"
```

Read

URL: https://localhost:7068/api/article/634edf085eeec043ecb385aa (The URL may be differed system to system)

POST Insert Article | GET Read Article | APF Beta2 Mobile App API

Read Article — Examples 0 — BUILD

GET https://localhost:7068/api/article/634edf085eeec043ecb385aa — Send — Save

Params | Authorization | Headers (7) | Body | Pre-request Script | Tests | Settings — Cookies

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body | Cookies | Headers (4) | Test Results — Status: 200 OK — Time: 159 ms — Size: 572 B — Save Response

Pretty | Raw | Preview | Visualize | JSON


```
{
    "articleId": "634edf085eeec043ecb385aa",
    "articleName": "MongoDB",
    "description": "MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server-Side Public License which is deemed non-free by several distributions.",
    "author": "Vadivel Rangasamy"
}
```


Update

URL: https://localhost:7068/api/Article/634edf085eeec043ecb385aa (The URL may be differed system to system)

POST Insert Article | GET Read Article | PUT Update Article | APF Beta2 Mobile App API

Update Article — Examples 0 — BUILD

PUT https://localhost:7068/api/Article/634edf085eeec043ecb385aa — Send — Save

Params | Authorization | Headers (9) | Body | Pre-request Script | Tests | Settings — Cookies

none | form-data | x-www-form-urlencoded | raw | binary | GraphQL | JSON — Beautif


```
{
    "articleName": "VisualStudio - Update",
    "description": "Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.",
    "author": "Vadivel Rangasamy-CBE"
}
```


Body | Cookies | Headers (2) | Test Results — Status: 204 No Content — Time: 23 ms — Size: 81 B — Save Response

Pretty | Raw | Preview | Visualize | Text

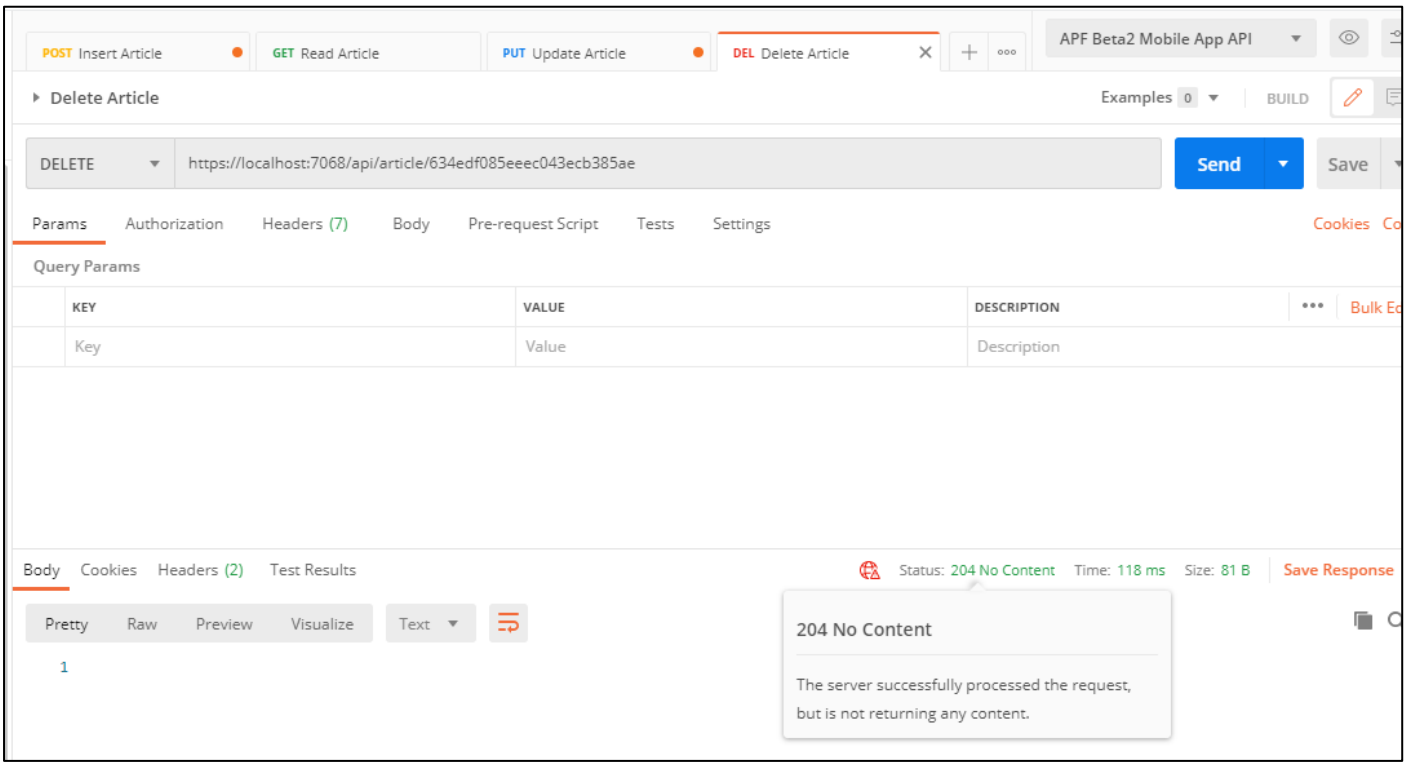
1

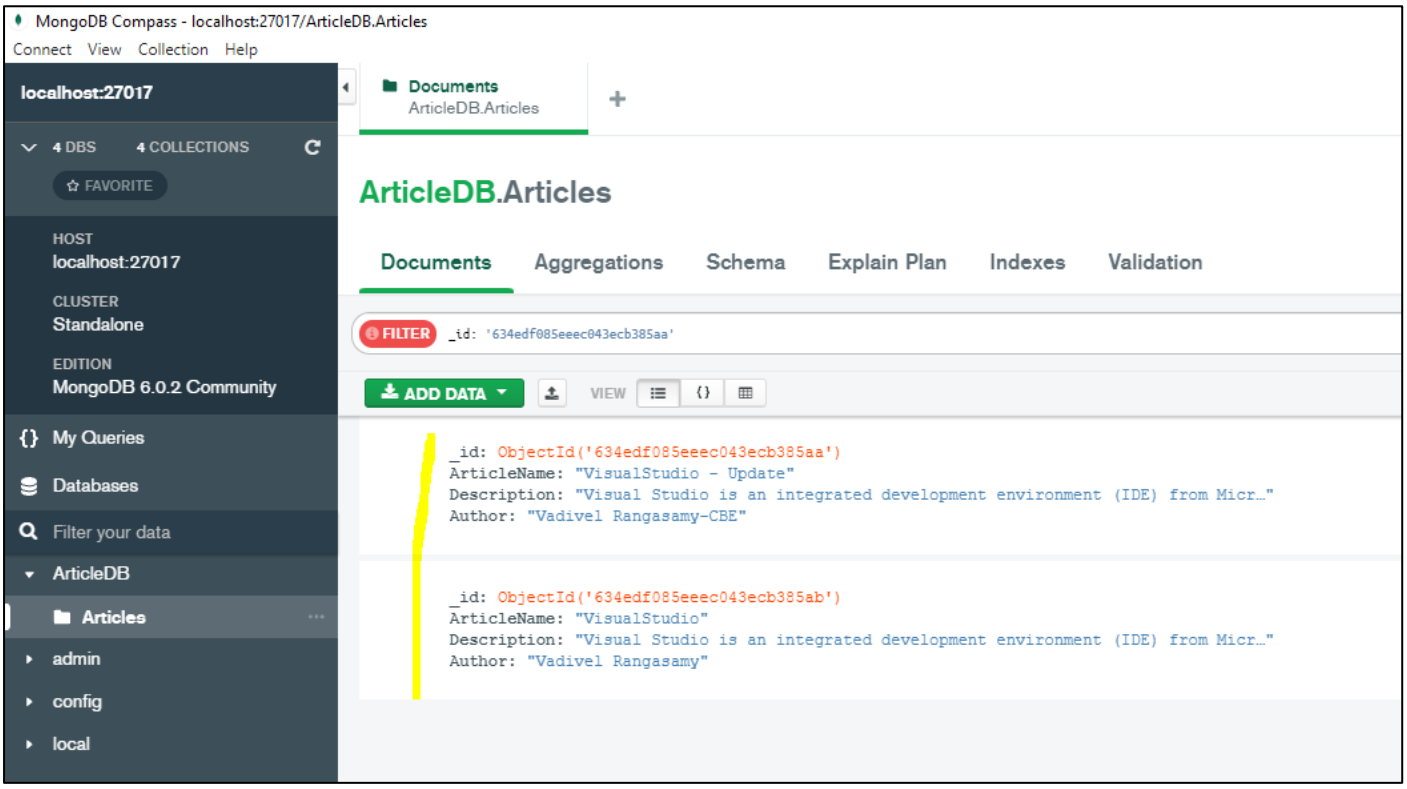
204 No Content

The server successfully processed the request, but is not returning any content.

Delete

URL: https://localhost:7068/api/Article/634edf085eeec043ecb385ae (The URL may be differed system to system)





Not yet to resolve

1. Directly used connection string *“var mongoClient = new MongoClient("mongodb://localhost:27017");”* instead of articleDatabaseSettings object *“var mongoClient = new MongoClient(articleDatabaseSettings.Value.ConnectionString);”.*

```
ArticleManagement | ArticleManagement.Services.ArticleService | ArticleService(IOptions<ArticleDatabaseSettings> articleDatabaseSetting
using MongoDB.Driver;
namespace ArticleManagement.Services
{
    4 references
    public class ArticleService
    {
        private readonly IMongoCollection<Article> _articlesCollection;

        0 references
        public ArticleService(
        IOptions<ArticleDatabaseSettings> articleDatabaseSettings)
        {
            bool isMongoLive = MongoDBConnectionProbe("mongodb://localhost:27017", "ArticleDB"); //database.RunCommandAsync((Command<BsonDocument>)"{ping:1}").Wait(1000);
            if (isMongoLive)
            {
                var mongoClient = new MongoClient("mongodb://localhost:27017"); //(articleDatabaseSettings.Value.ConnectionString);
                var mongoDatabase = mongoClient.GetDatabase("ArticleDB"); //(articleDatabaseSettings.Value.DatabaseName);
                _articlesCollection = mongoDatabase.GetCollection<Article>("Articles"); //(articleDatabaseSettings.Value.ArticleCollectionName);
            }
            else
            {
                // couldn't connect
            }
        }

        1 reference
        public async Task<List<Article>> GetAsync() =>
        await _articlesCollection.Find(_ => true).ToListAsync();
```

2. The web site is not published to a specific web server. I have directly run the website with auto generated URLs
   https://localhost:7068/ to https://testwebapi.com

Source code

URL: https://drive.google.com/file/d/1HJmpwRbUJZm8PKlysM1GPnPI55vCDbbT/view?usp=sharing