

Código

Aqui está o código completo:

```
using MongoDB.Driver;
using Microsoft.Extensions.DependencyInjection;
using StackExchange.Redis;
using System.Text.Json;
using MonitoramentoConsumo.Model;
using Microsoft.Extensions.Logging;

var builder = WebApplication.CreateBuilder(args);

var redisConnectionString = builder.Configuration.GetConnectionString("RedisConnection");
if (string.IsNullOrEmpty(redisConnectionString))
{
    throw new ArgumentNullException(nameof(redisConnectionString), "A configuração do Redis não pode ser nula.");
}

builder.Services.AddSingleton<IConnectionMultiplexer>(sp =>
    ConnectionMultiplexer.Connect(redisConnectionString));

builder.Services.AddSingleton<IDatabase>(sp =>
    sp.GetRequiredService<IConnectionMultiplexer>().GetDatabase());

builder.Services.Configure<MongoDbSettings>(
    builder.Configuration.GetSection("MongoDB")
);

builder.Services.AddSingleton<IMongoClient, MongoClient>(sp =>
    new MongoClient(builder.Configuration.GetValue<string>("MongoDB:ConnectionString"))
);
```

```

builder.Services.AddScoped(sp =>

sp.GetRequiredService<IMongoClient>().GetDatabase(builder.Configuration.GetValue<string>("MongoDB:DatabaseName")).GetCollection<EnergyConsumption>("EnergyConsumption")

);

builder.Services.AddControllers();
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();

builder.Services.AddLogging();

var app = builder.Build();

if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}

app.MapGet("/health", () => Results.Ok("Service is running"));

app.MapPost("/consumo", async (EnergyConsumption consumption, HttpContext context) =>
{
    var logger = context.RequestServices.GetRequiredService<ILogger<Program>>();
    try
    {
        var collection =
context.RequestServices.GetRequiredService<IMongoCollection<EnergyConsumption>>();
        await collection.InsertOneAsync(consumption);
        return Results.Created($" /consumo/{consumption.Id}", consumption);
    }
}

```

```

        catch (Exception ex)
        {
            logger.LogError(ex, "Erro ao salvar dados de consumo.");
            return Results.StatusCode(500);
        }
    });

app.MapGet("/consumo", async (HttpContext context) =>
{
    var logger = context.RequestServices.GetRequiredService<ILogger<Program>>();
    var cache = context.RequestServices.GetRequiredService<IDatabase>();
    var collection =
context.RequestServices.GetRequiredService<IMongoCollection<EnergyConsumption>>();

    try
    {
        var cachedData = await cache.StringGetAsync("consumoData");
        if (!cachedData.IsNullOrEmpty)
        {
            var consumoLista = JsonSerializer.Deserialize<List<EnergyConsumption>>(cachedData);
            logger.LogInformation("Dados recuperados do cache.");
            return Results.Ok(consumoLista);
        }

        logger.LogInformation("Dados não encontrados no cache, buscando no MongoDB.");
        var consumoList = await collection.Find(_ => true).ToListAsync();
        await cache.StringSetAsync("consumoData", JsonSerializer.Serialize(consumoList),
            TimeSpan.FromMinutes(5));
        logger.LogInformation("Dados recuperados do MongoDB e armazenados no cache.");
        return Results.Ok(consumoList);
    }
    catch (RedisConnectionException rex)

```

```
{
    logger.LogError(rex, "Erro ao conectar ao Redis.");
    return Results.StatusCode(500);
}
catch (MongoException mex)
{
    logger.LogError(mex, "Erro ao conectar ao MongoDB.");
    return Results.StatusCode(500);
}
catch (Exception ex)
{
    logger.LogError(ex, "Erro desconhecido.");
    return Results.StatusCode(500);
}
});
```

```
app.UseHttpsRedirection();
app.UseAuthorization();
app.MapControllers();
app.Run();
```

Prints:

Curl

```
curl -X 'GET' \
  'https://localhost:7058/health' \
  -H 'accept: */*'
```



Request URL

https://localhost:7058/health

Server response

Code Details

200

Response body

"Service is running"



Download

Response headers

content-type: application/json; charset=utf-8
date: Fri, 22 Nov 2024 01:39:45 GMT
server: Kestrel

Response

```
{  
  "id": "6",  
  "timestamp": "2024-11-22T01:40:10.722Z",  
  "consumption": 6345  
}
```

//

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'https://localhost:7058/consumo' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{  
  "id": "6",  
  "timestamp": "2024-11-22T01:40:10.722Z",  
  "consumption": 6345  
}'
```



Request URL

https://localhost:7058/consumo

Server response

Code Details

201

Undocumented

Response body

```
{  
  "id": "6",  
  "timestamp": "2024-11-22T01:40:10.722Z",  
  "consumption": 6345  
}
```



Download

Response headers

content-type: application/json; charset=utf-8
date: Fri, 22 Nov 2024 01:40:17 GMT
location: /consumo/6
server: Kestrel

Response

EnergyConsumptionDB > EnergyConsumptionDB > EnergyConsumption

Documents 2 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

Explain

Reset

Find

Options

ADD DATA

EXPORT DATA

UPDATE

DELETE

1 - 5 of 5

<div><div>_id: "2"</div><div>Timestamp : 2024-11-21T23:13:15.453+00:00</div><div>Consumption : 11231232120</div></div>
<div><div>_id: "3"</div><div>Timestamp : 2024-11-21T23:23:20.860+00:00</div><div>Consumption : 123140</div></div>
<div><div>_id: "4"</div><div>Timestamp : 2024-11-21T23:39:44.149+00:00</div><div>Consumption : 64543</div></div>
<div><div>_id: "5"</div><div>Timestamp : 2024-11-22T00:46:36.387+00:00</div><div>Consumption : 9873</div></div>

https://localhost:7058/consumo

Save

POST

https://localhost:7058/consumo

Send

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

JSON

Beautify

1

2

3

4

5

```
{
  "id": "7",
  "timestamp": "2024-11-22T01:40:10.720Z",
  "consumption": 2346345
}
```

Body

Cookies

Headers (5)

Test Results

Status: 201 Created

Time: 44 ms

Size: 245 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

```
{
  "id": "7",
  "timestamp": "2024-11-22T01:40:10.72Z",
  "consumption": 2346345
}
```

EnergyConsumptionDB > EnergyConsumptionDB > EnergyConsumption

Documents 2 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

Explain

Reset

Find

Options

ADD DATA

EXPORT DATA

UPDATE

DELETE

1 - 7 of 7

<div><div>_id: "4"</div><div>Timestamp : 2024-11-21T23:39:44.149+00:00</div><div>Consumption : 64543</div></div>
<div><div>_id: "5"</div><div>Timestamp : 2024-11-22T00:46:36.387+00:00</div><div>Consumption : 9873</div></div>
<div><div>_id: "6"</div><div>Timestamp : 2024-11-22T01:40:10.720+00:00</div><div>Consumption : 6345</div></div>
<div><div>_id: "7"</div><div>Timestamp : 2024-11-22T01:40:10.720+00:00</div><div>Consumption : 2346345</div></div>