Dưới đây là hướng dẫn từng bước để **tích hợp Redis với .NET Core** bằng **StackExchange.Redis** – thư viện Redis phổ biến nhất cho .NET.

**Bước 1: Cài đặt Redis**

Nếu bạn chưa cài đặt Redis, hãy làm theo hướng dẫn sau:

**Cài đặt Redis trên Windows**

1. **Dùng Memurai (Redis cho Windows)**:
   * Tải từ <https://www.memurai.com/>
   * Cài đặt và chạy:

sh

redis-server.exe

1. **Chạy Redis trong WSL (Windows Subsystem for Linux)**
   * Cài đặt Redis:

sh

sudo apt update

sudo apt install redis-server

* + Chạy Redis:

sh

redis-server

* + Kiểm tra Redis:

sh

redis-cli ping

Nếu trả về "PONG" là Redis đã chạy thành công.

**Bước 2: Tạo dự án .NET Core**

Mở terminal và tạo một dự án **ASP.NET Core Web API**:

sh

dotnet new webapi -n RedisDemo

cd RedisDemo

Cài đặt **StackExchange.Redis** – thư viện chính thức cho Redis trong .NET:

sh

dotnet add package StackExchange.Redis

**Bước 3: Cấu hình Redis trong .NET Core**

Mở file **appsettings.json** và thêm cấu hình Redis:

json

{

"Redis": {

"ConnectionString": "localhost:6379"

}

}

Sau đó, mở file **Program.cs** và đăng ký Redis vào DI container:

csharp

using StackExchange.Redis;

var builder = WebApplication.CreateBuilder(args);

// Lấy connection string từ appsettings.json

var redisConnectionString = builder.Configuration.GetValue<string>("Redis:ConnectionString");

// Đăng ký Redis Connection Multiplexer vào DI container

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

builder.Services.AddControllers();

var app = builder.Build();

app.MapControllers();

app.Run();

**Bước 4: Tạo Service để làm việc với Redis**

Tạo một thư mục **Services** và thêm file **RedisService.cs**:

csharp

using StackExchange.Redis;

public class RedisService

{

private readonly IConnectionMultiplexer \_redis;

private readonly IDatabase \_db;

public RedisService(IConnectionMultiplexer redis)

{

\_redis = redis;

\_db = \_redis.GetDatabase();

}

// Lưu dữ liệu vào Redis

public async Task SetValueAsync(string key, string value, TimeSpan? expiry = null)

{

await \_db.StringSetAsync(key, value, expiry);

}

// Lấy dữ liệu từ Redis

public async Task<string?> GetValueAsync(string key)

{

return await \_db.StringGetAsync(key);

}

}

Sau đó, **đăng ký RedisService vào DI container** trong **Program.cs**:

csharp

builder.Services.AddSingleton<RedisService>();

**Bước 5: Tạo API để lưu và lấy dữ liệu từ Redis**

Mở **Controllers**, thêm **RedisController.cs**:

csharp

using Microsoft.AspNetCore.Mvc;

[Route("api/redis")]

[ApiController]

public class RedisController : ControllerBase

{

private readonly RedisService \_redisService;

public RedisController(RedisService redisService)

{

\_redisService = redisService;

}

[HttpPost("set")]

public async Task<IActionResult> SetValue([FromQuery] string key, [FromQuery] string value)

{

await \_redisService.SetValueAsync(key, value, TimeSpan.FromMinutes(5));

return Ok($"Đã lưu {key}: {value} vào Redis");

}

[HttpGet("get")]

public async Task<IActionResult> GetValue([FromQuery] string key)

{

var value = await \_redisService.GetValueAsync(key);

return value != null ? Ok(value) : NotFound("Key không tồn tại trong Redis");

}

}

**Bước 6: Chạy và kiểm tra API**

Chạy ứng dụng:

sh

dotnet run

API sẽ chạy trên http://localhost:5000.

**Gửi dữ liệu vào Redis**

Dùng Postman hoặc trình duyệt để gọi API:

sh

GET http://localhost:5000/api/redis/set?key=username&value=chatgpt

📌 Redis sẽ lưu **username = "chatgpt"** trong 5 phút.

**Lấy dữ liệu từ Redis**

Gọi API để lấy dữ liệu:

sh

GET http://localhost:5000/api/redis/get?key=username

✅ Kết quả: "chatgpt"

**Bước 7: Kiểm tra Redis bằng Redis CLI**

Bạn có thể kiểm tra dữ liệu trực tiếp trong Redis bằng Redis CLI:

sh

redis-cli

GET username

Nếu thấy "chatgpt", Redis đã hoạt động đúng!

**Tóm tắt**

✅ Cài đặt Redis trên Windows/Linux  
✅ Sử dụng **StackExchange.Redis** để kết nối Redis với .NET Core  
✅ Tạo API lưu và lấy dữ liệu từ Redis  
✅ Kiểm tra Redis bằng Postman hoặc Redis CLI