

How to create .NET8 CRUD WebAPI Azure Cache for Redis Microservice

You can find the source code for this example in this github repo:

https://github.com/luisco/MicroServices_dotNET8_CRUD_WebAPI-Azure-Cache-for-Redis

1. Create Azure Cache for Redis in Azure Portal

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the 'Microsoft Azure' logo, a search bar, and user information. The left sidebar lists various services and resources. The main content area displays the 'Azure Cache for Redis' service card, which is highlighted with a red box. The card includes a description, a 'Create' button, and a 'View' button. Below the card, there is a table of training resources from Microsoft.

Free training from Microsoft	Last Viewed
Work with mutable and partial data in Azur... 8 units · 1 hr 5 min	57 minutes ago
Optimize your web applications by caching ... 7 units · 55 min	2 weeks ago
Accelerate and scale a Spring Boot applicati... 9 units · 41 min	3 weeks ago

Microsoft Azure

Search resources, services, and docs (G+)

Home >

Azure Cache for Redis

particular (luiscocoenriquez@hotmail.onmicrosoft.com)

+ Create Manage view Refresh Export to CSV Open query Assign tags

Filter for any Create Subscription equals all Resource group equals all Location equals all Add filter

Showing 0 to 0 of 0 records.

No grouping List view

Name ↑↓	Location ↑↓	Status ↑↓	Size ↑↓	SKU ↑↓	Subscription ↑↓
---------	-------------	-----------	---------	--------	-----------------

[Home](#) > [Azure Cache for Redis](#) >

New Redis Cache ...

[Basics](#) [Networking](#) [Advanced](#) [Tags](#) [Review + create](#)

Azure Cache for Redis helps your application stay responsive even as user load increases. It does so by leveraging the low latency, high-throughput capabilities of the Redis engine. [Learn more](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Subscription 1

Resource group *

myRG

[Create new](#)

Instance Details

DNS name *

myrediscache1974

.redis.cache.windows.net

Location *

West Europe

Cache type ([View full pricing details](#)) *

Basic C0 (250 MB Cache, No SLA)

See which features are available on each Azure Cache for Redis tier [Learn more](#)

[Review + create](#)

[< Previous](#)

[Next : Networking >](#)

[+ Create a resource](#)

[Home](#)

[Dashboard](#)

[All services](#)

[FAVORITES](#)

[All resources](#)

[Resource groups](#)

[Quickstart Center](#)

[App Services](#)

[Function App](#)

[SQL databases](#)

[Azure Cosmos DB](#)

[Virtual machines](#)

[Load balancers](#)

[Storage accounts](#)

[Virtual networks](#)

[Microsoft Entra ID](#)

[Monitor](#)

[Advisor](#)

[Microsoft Defender for Cloud](#)

[Cost Management +](#)

The screenshot displays the Microsoft Azure portal interface. On the left, the 'All resources' list shows 'myrediscache1974' selected. The main pane shows the 'Overview' tab for this Azure Cache for Redis instance. Key details include:

- Resource group:** myRG
- Status:** Running - Basic 250 MB
- Location:** West Europe
- Subscription ID:** 99888cc6-c635-4ebd-b0ac-1be1dace0089
- Host name:** myrediscache1974.redis.cache.windows.net
- Ports:** Non-SSL port (6379) disabled
- Keys:** [Show access keys...](#)
- *Best practices:** <https://aka.ms/redis/p/bestpractices>
- *New features:** <https://aka.ms/newfeatures>

The 'Memory Usage' section features a line graph showing memory usage over time. The y-axis ranges from 0B to 450kB. The x-axis shows time from 12 PM to 6 AM UTC. A blue line indicates the 'Used Memory (Max)' for 'myrediscache1974', which is currently at 413.35 kB.

We copy the connection string in the appsettings.json file

Microsoft Azure

Search resources, services, and docs (G+)

Home > All resources > myrediscache1974

All resources

particular (luiscocoenriquezhotmail.onmicrosoft.c...

+ Create Manage view ...

Filter for any field...

Name ↑↓

myrediscache1974

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Events

Settings

Access keys

Advanced settings

myrediscache1974 | Access keys

Azure Cache for Redis

Search

Regenerate Primary Regenerate Secondary

Primary

Secondary

Primary connection string (StackExchange.Redis)

Secondary connection string (StackExchange.Redis)

For information on other clients see: <https://aka.ms/redisclients>

2. appsettings.json

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "RedisCache": "myrediscache1974.redis.cache.windows.net:6380,password=vFuzCF5i81hLvJmmvCPTBnaoN17HMYJCyAzCaHzbkb0=,ssl=True,a"
  }
}
```

3. We set the Middleware (Program.cs)

```
using Microsoft.AspNetCore.Builder;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using StackExchange.Redis;
using AzureCacheforRedis.Services;
using Microsoft.OpenApi.Models;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddControllers();

// Add Swagger service
builder.Services.AddSwaggerGen(c =>
{
    c.SwaggerDoc("v1", new OpenApiInfo { Title = "Azure Cache for Redis API", Version = "v1" });
});

builder.Services.AddSingleton<IConnectionMultiplexer>(
    ConnectionMultiplexer.Connect(builder.Configuration.GetConnectionString("RedisCache")));
builder.Services.AddScoped<RedisCacheService>();

var app = builder.Build();

// Configure the HTTP request pipeline.
if (app.Environment.IsDevelopment())
{
    app.UseDeveloperExceptionPage();

    // Enable middleware to serve generated Swagger as a JSON endpoint
    app.UseSwagger();

    // Enable middleware to serve swagger-ui (HTML, JS, CSS, etc.)
    app.UseSwaggerUI(c => c.SwaggerEndpoint("/swagger/v1/swagger.json", "Azure Cache for Redis API V1"));
}
```

```
app.UseAuthorization();
```

```
app.MapControllers();
```

```
app.Run();
```

4. We create the Service (RedisCacheService.cs)

```
using StackExchange.Redis;  
using System.Threading.Tasks;
```

```
namespace AzureCacheforRedis.Services  
{  
    public class RedisCacheService  
    {  
        private readonly IConnectionMultiplexer _connectionMultiplexer;  
  
        public RedisCacheService(IConnectionMultiplexer connectionMultiplexer)  
        {  
            _connectionMultiplexer = connectionMultiplexer;  
        }  
  
        public async Task SetAsync(string key, string value)  
        {  
            var db = _connectionMultiplexer.GetDatabase();  
            await db.StringSetAsync(key, value);  
        }  
  
        public async Task<string> GetAsync(string key)  
        {  
            var db = _connectionMultiplexer.GetDatabase();  
            return await db.StringGetAsync(key);  
        }  
    }  
}
```

```
public async Task DeleteAsync(string key)
{
    var db = _connectionMultiplexer.GetDatabase();
    await db.KeyDeleteAsync(key);
}

// Other CRUD operations as needed

}
```

5. We create the Controller (CacheController.cs)

```
using AzureCacheforRedis.Services;
using Microsoft.AspNetCore.Mvc;
using System.Threading.Tasks;

namespace AzureCacheforRedis.Controllers
{
    [ApiController]
    [Route("[controller]")]

    public class CacheController : ControllerBase
    {
        private readonly RedisCacheService _cacheService;

        public CacheController(RedisCacheService cacheService)
        {
            _cacheService = cacheService;
        }

        [HttpPost]
        public async Task<IActionResult> Set(string key, string value)
        {

```



```
        await _cacheService.SetAsync(key, value);
        return Ok();
    }

    [HttpGet("{key}")]
    public async Task<IActionResult> Get(string key)
    {
        var value = await _cacheService.GetAsync(key);
        return Ok(value);
    }

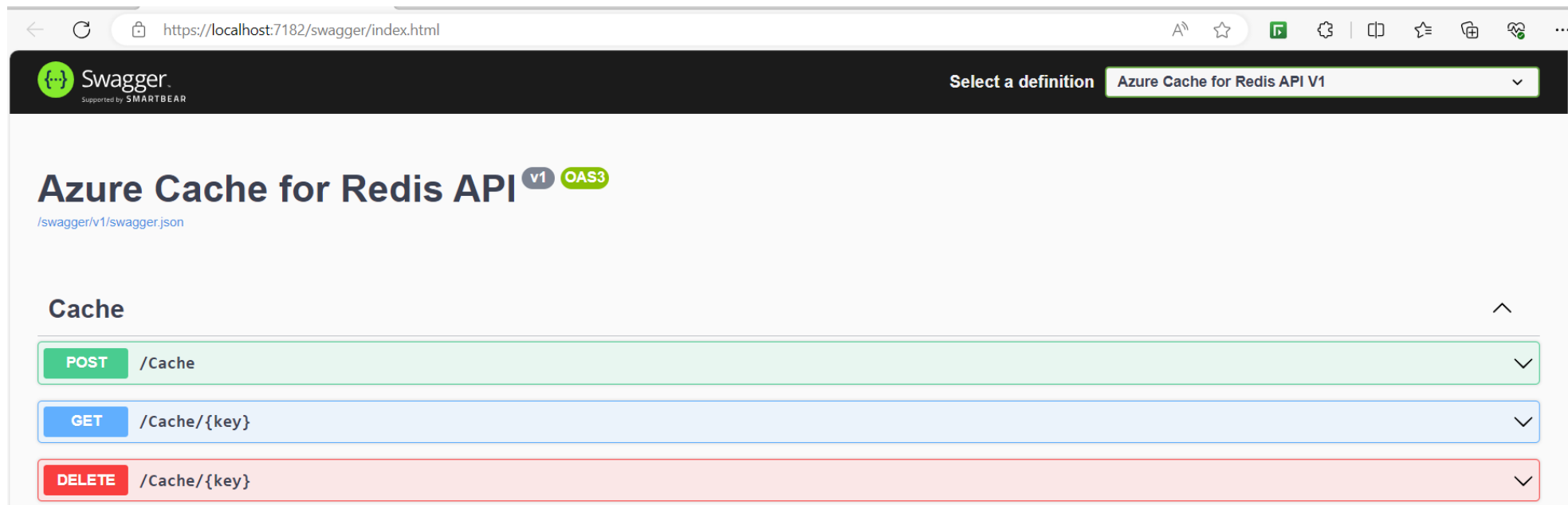
    [HttpDelete("{key}")]
    public async Task<IActionResult> Delete(string key)
    {
        await _cacheService.DeleteAsync(key);
        return Ok();
    }

    // Add other endpoints for CRUD operations

    }
}
```

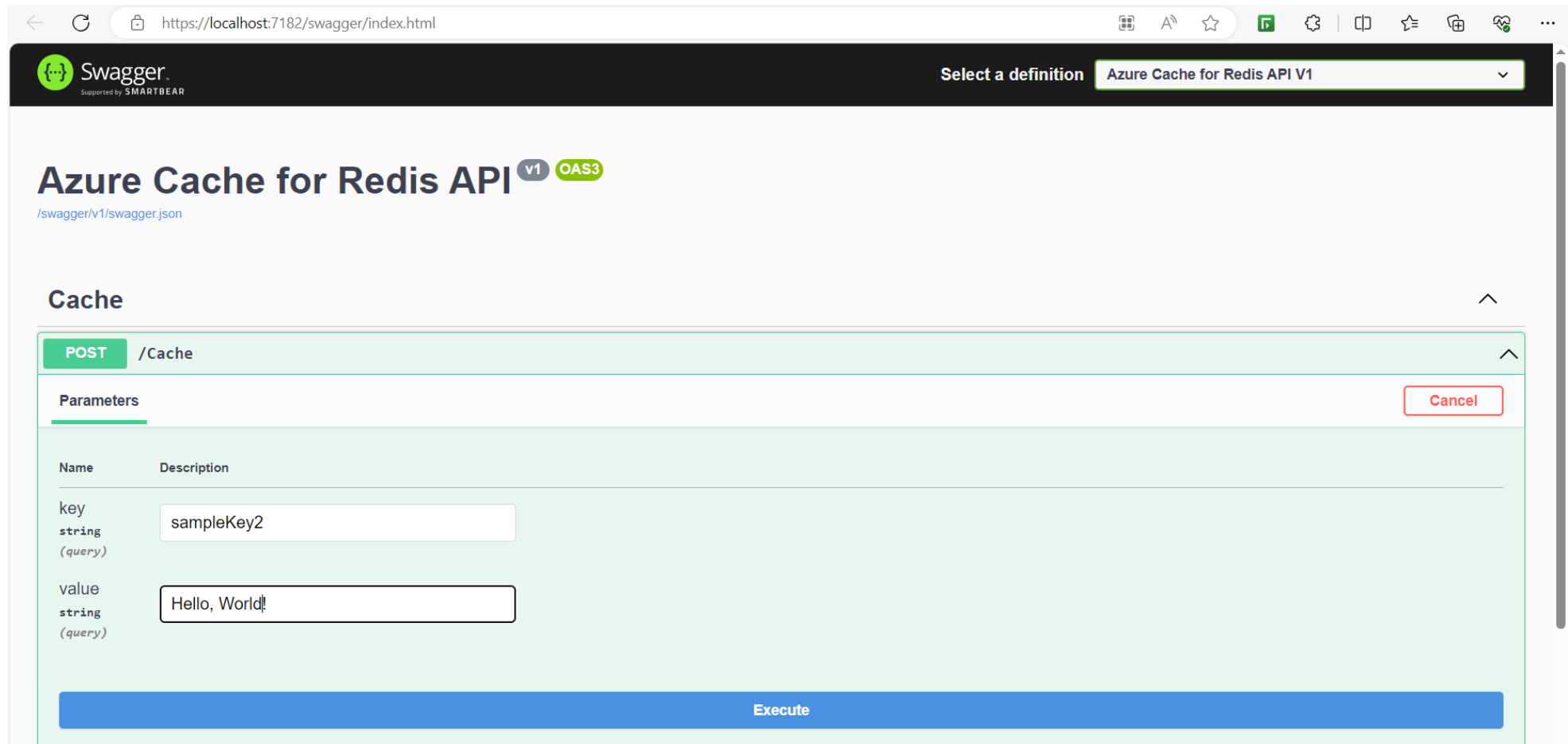
6. Verify the application

<https://localhost:7182/swagger/index.html>



The image shows a web browser displaying the Swagger UI for the Azure Cache for Redis API V1. The browser's address bar shows the URL `https://localhost:7182/swagger/index.html`. The Swagger UI header includes the Swagger logo, the text "Supported by SMARTBEAR", and a dropdown menu labeled "Select a definition" with "Azure Cache for Redis API V1" selected. The main title is "Azure Cache for Redis API" with "v1" and "OAS3" badges. Below the title is the link `/swagger/v1/swagger.json`. The "Cache" section is expanded, showing three endpoints: a POST endpoint at `/Cache`, a GET endpoint at `/Cache/{key}`, and a DELETE endpoint at `/Cache/{key}`. Each endpoint is represented by a colored bar (green for POST, blue for GET, red for DELETE) with a dropdown arrow on the right.

We send a POST request



The image shows a web browser window displaying the Swagger UI for the Azure Cache for Redis API V1. The browser's address bar shows the URL `https://localhost:7182/swagger/index.html`. The Swagger logo is in the top left, and a dropdown menu in the top right shows "Select a definition" with "Azure Cache for Redis API V1" selected. The main heading is "Azure Cache for Redis API" with "v1" and "OAS3" labels. Below the heading is the link `/swagger/v1/swagger.json`. The "Cache" section is expanded, showing a "POST /Cache" endpoint. Under the "Parameters" tab, there are two query parameters: "key" (string) with a value of "sampleKey2" and "value" (string) with a value of "Hello, World!". A red "Cancel" button is in the top right of the parameters section, and a blue "Execute" button is at the bottom.

Swagger
Supported by SMARTBEAR

Select a definition Azure Cache for Redis API V1

Azure Cache for Redis API ^{v1} OAS3

</swagger/v1/swagger.json>

Cache

POST /Cache

Parameters Cancel

Name	Description
key string (query)	<input type="text" value="sampleKey2"/>
value string (query)	<input type="text" value="Hello, World!"/>

Execute

We send a GET request

Swagger UI

https://localhost:7182/swagger/index.html

GET /Cache/{key}

Parameters Cancel

Name	Description
key * required string (path)	<input type="text" value="sampleKey2"/>

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7182/Cache/sampleKey2' \
  -H 'accept: */*'
```

Request URL

```
https://localhost:7182/Cache/sampleKey2
```

Server response

Code	Details
200	<p>Response body</p> <pre>Hello, World!</pre>