Redis Summery By Mahmoud Badawy

-----------------------------------------------------

ملخص Redis

المراجع المستخدمه فى الشرح   
<https://www.codeproject.com/Articles/1120038/A-simple-Csharp-cache-component-using-Redis-as-pro?fbclid=IwAR04odm8tlRDj_mb5k_i-cI3eSHtEp9P8Jzm56N4z9KB0wFGMtY8_2SMSp4>

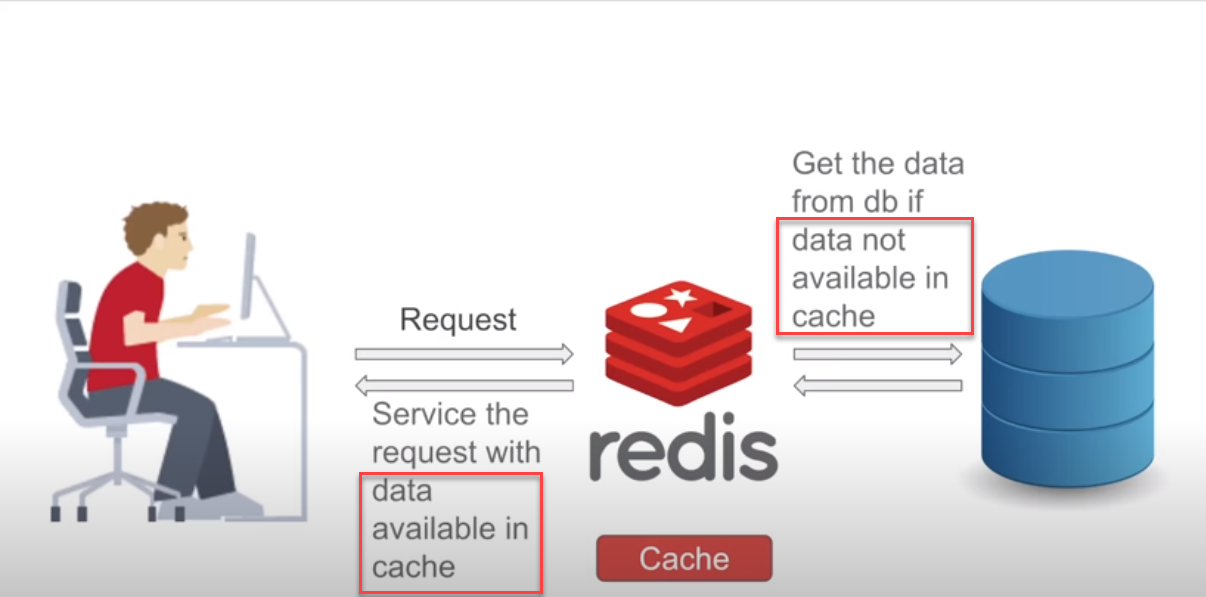
**و الفيدوهات المهمه دى جميله**   
<https://www.youtube.com/watch?v=HNDtcXVo5ow&list=PLhW3qG5bs-L8n1fsiT8z_VnDhnUk4vaVq&index=2&t=0s>

MB

########################################################################

Redis meaning and What is it ?

-----------------------

Remote dicrionary server = Redis   
ده ايه البتاع ده و ايه لازمته ؟؟ بص يا سيدى ده عباره عن حاجه بتعمل مكان فى الميمورى بخزن فيه الداتا الى انا عاوز اخزنها و بالتالى بتبقى عباره عن كاش من الداتا بيز و بالتالى لو عندى داتا حجمها كبير و جبتها اول مره فى وقت كبير فهيروح مخزنها فى الكاش بتاعه فلما اطلبها تانى هتبقى متسجله فى الميميورى و بالتالى هيجيبها من الكاش بسرعه كبيره جدا جدا بدل ما يروح يجيبها من الداتا بيز و عينه تطلع على ما يجيبها و بالتالى Redis عباره عن حاجه بخزن فيها البيانات الى بحتاجها بشكل متكرر و حجمها كبير عشان لما احتجها تيجى تانى تيجى بسرعه و ميروحش يجيبها من الداتا بيز يجيبلى شلل من البطء و ده هيوريك الميكانيزم الى الموضوع ده ماشى بيه حيث ان لو الداتا متوفر فى الكاش بيخدها من الكاش لو مش متوفره فى الكاش بيروح يجيبها من الداتا بيز   


هى عباره عن in-Memory DB او NoSQL DB و بالتالى مفيهاش جداول ولا سليكت ولا انسيرت ولا الكلام ده و كمان هى فى التخزين مبتسجلش فى جداول و الكلام ده لا بتستخدم ال data structure و بالتالى هى بتعتمد فى التخزين على ال Key and Value ذى   
Name = “Mahmoud Badawy” هنا ال Key هى ال Name و ال Value هى Mahmoud Badawy

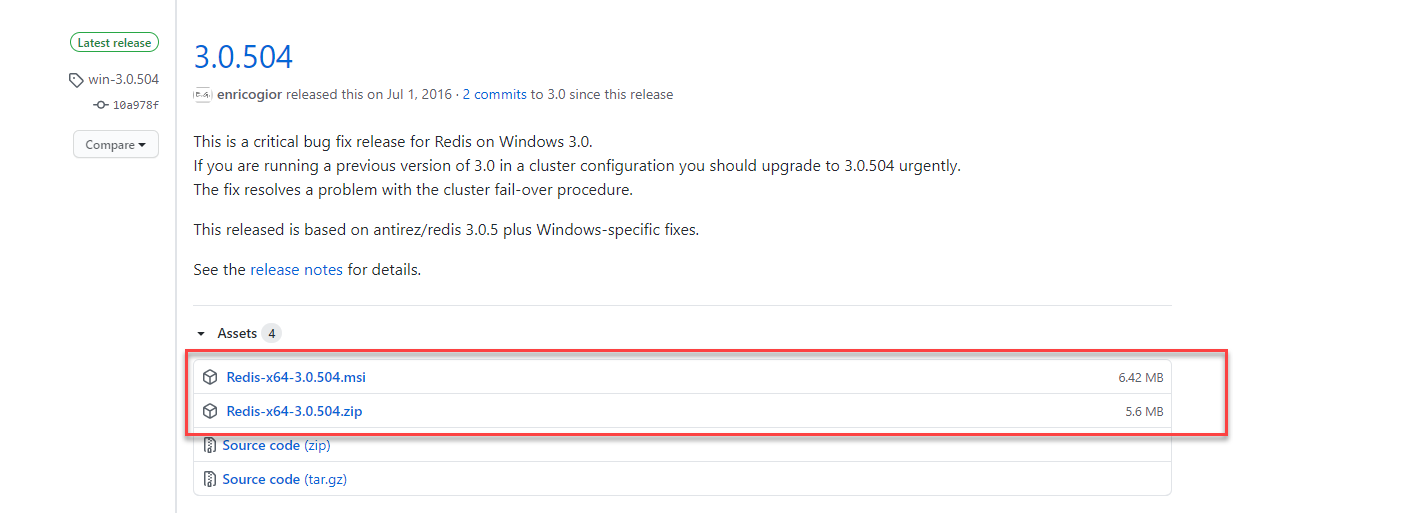
و ممكن تبقى ال Key اسم جدول و ال Value اوبجيكت كامل فيه بيانات الجدول ده بس فى الاخر التخزين مش فى جداول لا فى Key , Value

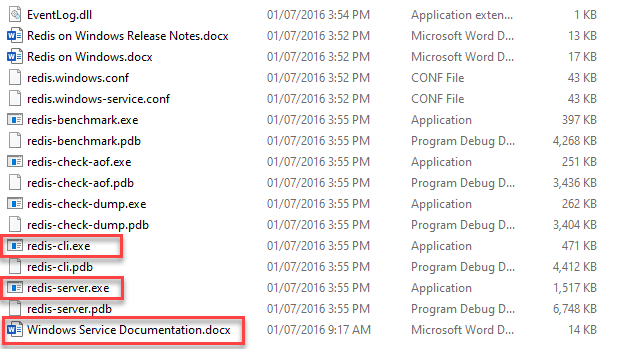
MB

########################################################################

How to install Redis in Windows ?

------------------------------------------

أول حاجه هتحمل البرنامج من هنا   
<https://github.com/microsoftarchive/redis/releases>  


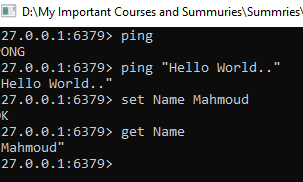
يفضل تحمل نسخه ال MSI عشان دى تسطيب و بتعمل كل حاجه لوحدها و بتظبط الدنيا مع نفسها بس حمل الاتنين عشان الى مضغوط ده فيه ملفات مهمه بيشرح فيها هتشتغل اذاى   
  
بعد كده هتسطب ذى اى برنامج هتسيب كل حاجه على ال Default بتعها   
طيب عاوز اتأكد ان الدنيا شغاله فى Redis هتعمل ايه ؟؟  
حمل النسخه ال Zip و فكها هتلاقى ملف مكتوبه عليه Redis-server و كمان ملف اسمه redis-cli دول افتحهم و جرب بقه شوف البورتات اتخدت صح و بيانات السيرفر بتاع redis و جرب فى ال redis-cli تعمل set , get لاى قيمه كده   


هتدوس على redis-server.exe المفروض هتجيبلك بيانات السيرفر ان السيرفر شغال و على البورت الفلانى و هكذا

بعد كده هتدوس على ال redis-cli.exe و تكتب فيه ping هيرد عليك ب PONG هتروح كاتب

ping "Hello World.."

هيرد عليك Hello world كده بقه اشتغل عادى و اعمل Set , get يعنى اكتب   
set Name Mahmoud و بعد كده جرب تقوله Get Name هتلاقيه جابلك Mahmoud يبقى كده الدنيا شغاله و ذى الفل



كده يا باشا سيرفر redis شغال و ذى الفل كده بقه نروح نسطب ال DeskTop بتاعه عشان نراقب منه الداتا الى متخزنه فى ال Cache بتاع الميمورى

MB

########################################################################

How to use Redis Desktop Manager?

------------------------------------------

هنخش هنا و نحمل البرنامج و نسطبه و نتبع التعليمات ذى ما مكتوب هناك

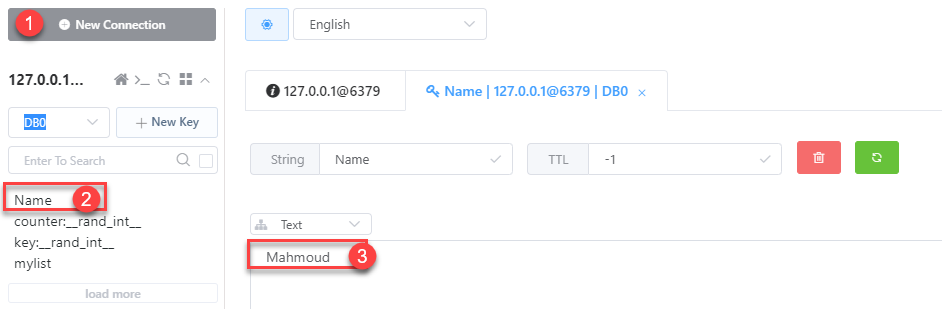
<https://redisdesktopmanager.readthedocs.io/en/latest/install/>

بس دى بقت بفلوس هات امو بلاش بقه

<https://www.electronjs.org/apps/anotherredisdesktopmanager>

<https://github.com/qishibo/AnotherRedisDesktopManager>

هتسطبه و تفتحه و تكتب فيه اللوكال هوست و البورت و غالبا بيبقوا مكتوبين اصلا و هتشوف لو عاوز تشفر الداتا و كده قشطه المهم انك اول ما هتفتح هتلاقى الدنيا كده و بما انك لقيت فى داتا بيز فيها Key اسمه Name و قيمته محمود يبقى ذى الفل كل حاجه شغاله عنب



كده بقه الى باقى انى اعمل set و Get بالكود و ده بقه الى جاى يا برنس

MB

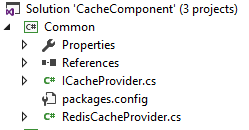
########################################################################

Redis in .Net

------------------------------------------

هنعمل بروجيكت من نوع Class Library و نسميه Common و نسطب الباكيدج دى بقه فى البروجيكت ده بقه Install-Package ServiceStack.Redis

دى الباكيدج الى بشغل بيها الاكواد الى بتتعامل مع redis عشان اخزن بقه كاشات و اروق على حالى



بعد كده هعمل Interface فى مشروع ال Common و هسميه ICacheProvider

using System;

namespace Common

{

public interface ICacheProvider

{

void Set<T>(string key, T value);

void Set<T>(string key, T value, TimeSpan timeout);

T Get<T>(string key);

bool Remove(string key);

bool IsInCache(string key);

}

}

بعد كده هعمل ال implemention بتاع الانترفيس ده بقه فى كلاس ال RedisCacheProvider

using ServiceStack.Redis;

using System;

namespace Common

{

public class RedisCacheProvider : ICacheProvider

{

RedisEndpoint \_endPoint;

public RedisCacheProvider()

{

\_endPoint = new RedisEndpoint("localhost", 6379, null,0 );

}

public void Set<T>(string key, T value)

{

this.Set(key, value, TimeSpan.Zero);

}

public void Set<T>(string key, T value, TimeSpan timeout)

{

using (RedisClient client = new RedisClient(\_endPoint))

{

client.As<T>().SetValue(key, value, timeout);

}

}

public T Get<T>(string key)

{

T result = default(T);

using (RedisClient client = new RedisClient(\_endPoint))

{

var wrapper = client.As<T>();

result = wrapper.GetValue(key);

}

return result;

}

public bool Remove(string key)

{

bool removed = false;

using (RedisClient client = new RedisClient(\_endPoint))

{

removed = client.Remove(key);

}

return removed;

}

public bool IsInCache(string key)

{

bool isInCache = false;

using (RedisClient client = new RedisClient(\_endPoint))

{

isInCache = client.ContainsKey(key);

}

return isInCache;

}

}

}

اول حاجه خد reference من بروجيكت ال common داخل بروجيكت ال web.Api الى فيه شغل ال Controllers

جوه مشروع ال web.API

بعد كده هتسجل سيرفيس ال Redis دى جوه ال StartUp بقه عشان تشتغل

public void ConfigureServices(IServiceCollection services)

{

var options = new DbContextOptionsBuilder<SPSDBContext>().

UseInMemoryDatabase(databaseName: "SPSDB").Options;

using (var context = new SPSDBContext(options))

{

// add service here

services.AddScoped<SPSDBContext>();

services.AddTransient<ICitiesService, CitiesService>();

services.AddScoped<ICacheProvider, RedisCacheProvider>();

services.AddScoped(typeof(IRepository<>), typeof(Repository<>));

services.AddAutoMapper(AppDomain.CurrentDomain.GetAssemblies());

services.AddControllers();

}

}

بعد كده تقدر تستخدم اكواد redis بقه فى الكونترولار كده

بعد كده هروح اكتب كود فى الكنترولار بقه

فمثلا انا عاوز اخزن و ليكن الداتا الى راجعه من GetAllCities فى كنترولار ال Cities

تعالى نجرب نشغلها و نشوف تأثيرها هيبقى عامل اذاى ؟؟

namespace Web.API.Controllers

{

[ApiController]

[Route("[controller]")]

public class CitiesController : ApiBaseController

{

private readonly ICacheProvider \_cacheProvider;

public CitiesController(

ICacheProvider cacheProvider) :base() {

\_cacheProvider = cacheProvider;

}

[HttpGet]

public IActionResult Get()

{

string UserName = \_cacheProvider.Get<string>("UserName");

return Ok(UserName);

}

[HttpPost]

public IActionResult Post(Cities VM)

{

\_cacheProvider.Set<string>("UserName", "Mahmoud Badawy");

return Ok();

}

}

}

انا عملت بقه حاجه كده لما اعمل Post يروح يسجلى اسم مستخدم Mahmoud Badawy فى الميمورى و لما اعوز اجيبه اجيبه من ال Get و ده معناه انه بيجيبه بعد ريكويست و بالتالى اقدر استخدم الكلام ده فى حفظ ال Token information ذى ال username و كل بيانات المستخدم المهمه خلال السيسشن بتاعه و اول ما يخرج امسح البيانات دى من الميميورى و يبقى كده انا مبخزنش اى حاجه فى ال client Side و كله متخزن فى ال Server side و بالتالى أمنت معلومات المستخدم بشكل كبير جدا

على فكره انت ممكن تحفظ أوبجيكت كامل ذى كده يا اسطى

namespace Web.API.Controllers

{

[ApiController]

[Route("[controller]")]

public class CitiesController : ApiBaseController

{

private readonly ICitiesService \_CTRL\_Service;

private readonly IMapper \_mapper;

private readonly ICacheProvider \_cacheProvider;

public CitiesController(ILogger<CitiesController> logger ,

ICitiesService CTRL\_Service , IMapper mapper ,

ICacheProvider cacheProvider)

:base(logger) {

\_CTRL\_Service = CTRL\_Service;

\_mapper = mapper;

\_cacheProvider = cacheProvider;

}

[HttpGet]

public IActionResult Get()

{

var RedisCities = \_cacheProvider.Get<List<CitiesVM>>("Cities");

return Ok(RedisCities);

}

[HttpPost]

public IActionResult Post()

{

List<Cities> Data = \_CTRL\_Service.GetAllCities().ToList();

List<CitiesVM> Item = \_mapper.Map<List<CitiesVM>>(Data);

\_cacheProvider.Set<List<CitiesVM>>("Cities", Item);

return Ok();

}

}

}