**Asp.NetCore API With Caching**

**Notes: -**

**1-with using Caching you don’t need to access to database, only you want to use caching like Radius (after the first load from database store in the cache)**

**2-the access to the database take some times and resources**

**3-Redius is based on key value pairs so its provide high performance**

**Steps: -**

**1-create new project application Blazor server called RedisDemo**

**2-with using Docker is tool that you can manipulate docker images and containers**

**Which you can remove and restore**

**//it will run and download image redis on local docker machine and run on port 5002**

**Docker run --name my-redis -p 5002:6379 -d redis**

**//to show all the containers and the image built from it**

**docker ps -a**

**//to access to the redis command shell**

**docker exec -it my-redis sh**

**//to access to redis command line interface**

**redis-cli**

**127.0.0.1:6379>ping //to test redis cache**

**127.0.0.1:6379> select 0 //to get first element on the redis cache**

**127.0.0.1:6379> dbsize //to show database size**

**127.0.0.1:6379> scan 0 //to get all keys on the redis database**

**//to access to the item based on key as below**

**127.0.0.1:6379> hgetall RedisDemo\_WeatherForeCast\_20210502\_1026**

**3-we install the following NuGet packages as below (the boths working on hand by hand)**

**//to allow caching on my application**

**Microsoft.Extensions.Caching.StackExchangeRedis**

**//provide Redis provider caching**

**StackExchange.Redis**

**4-make the caching life time short as possible to make its performance**

**(most of times we work with AbsoluteExpirationRelativeToNow)**