Redis is in memory cache

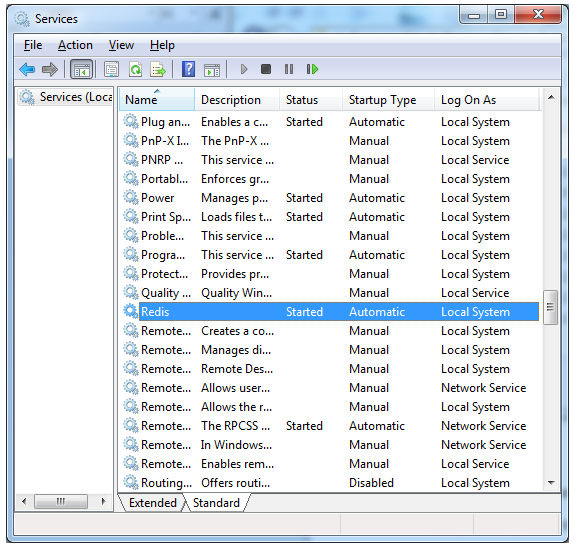
Redis is an open source, fast, feature rich, in-memory caching engine that can be used to store and retrieve data in your applications. ... Redis Cache is an open source, high-speed, NoSQL database. It's fast, and it runs entirely in memory with negligible performance overhead when reading and writing data.

Best practices for using Redis?

https://redis.io/

From above link we can download Redis cache exe and install server

After insolation we can verify running state of Redis server using following services windows

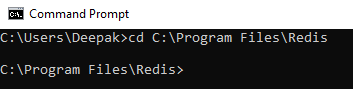


After Redis Server is installed in system we can check the installed files using following path

C:\Program Files\Redis

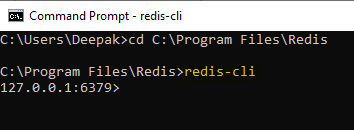
Note: all command needs to be executed from above directory

Using following command, we can go to this directory



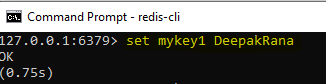
We can check following command to check running server IP and port

redis-cli

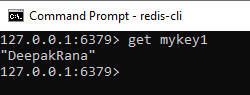


We can set key using following command

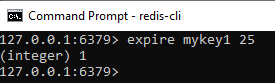
Note: Redis CLI gives us intelligence support also.



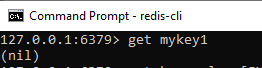
We can get key value using following command



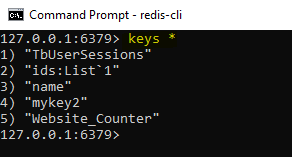
We can set expiry duration for our key even after creation



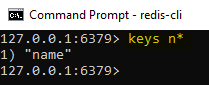
If I try after expiration time we will not get value



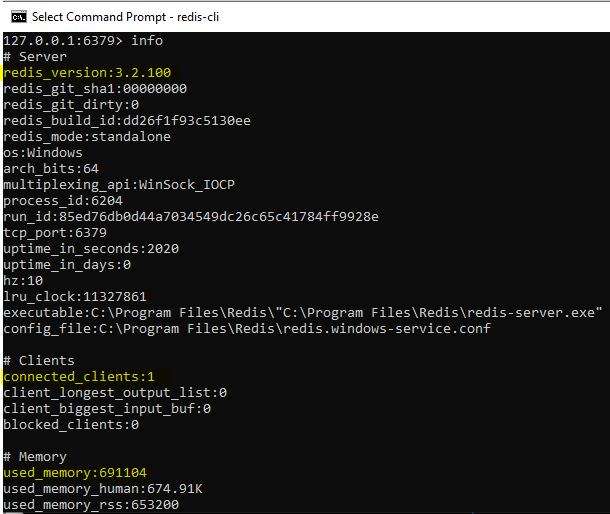
We can get all the key



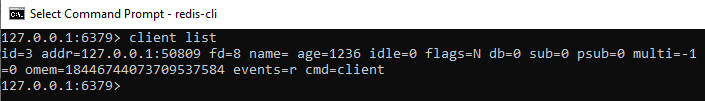
We can search key my a search pattern



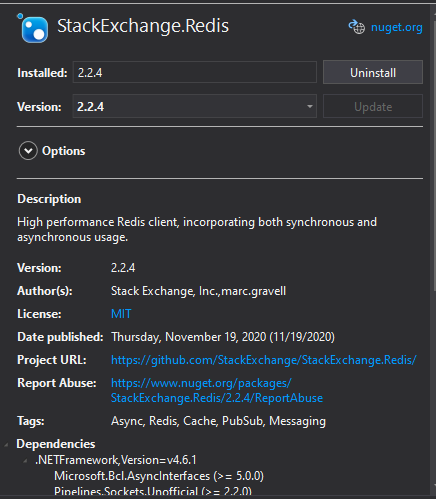
Using following command we can get all the information about Redis server

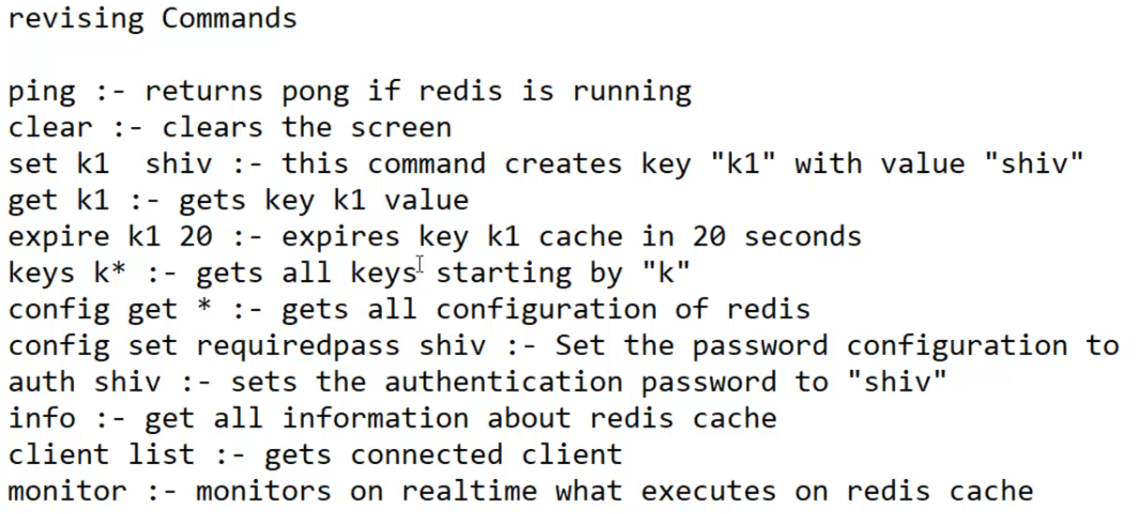


Get all the connected clients



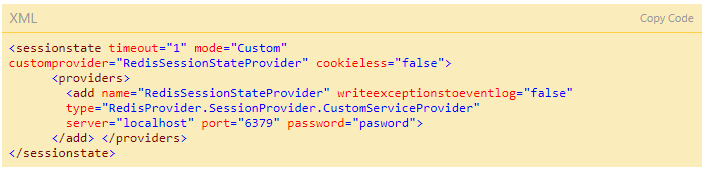
**Install following package in your .net application**





**ASP.NET Session State with Redis**

To configure ASP.NET session state with Redis provider, add a new file to your web project, named RedisSessionStateProvider.cs, copy code from https://github.com/chadman/redis-service-provider/raw/master/RedisProvider/SessionProvider/RedisSessionProvider.cs, then add or change the following section in the configuration file (sessionState tag has to be inside system.web tag), or you can download attached sources and copy code.



NOTE, that password is optional, based on the server authentication. It must be replaced with real value, or removed, if Redis server doesn't require authentication. Server attribute and port also have to be replaced according to concrete values (default port is 6379). Then in the project, you can use the session state:

**Raids Cache Utility**

public class RadisCacheUtility

{

private readonly RedisEndpoint \_redisEndpoint;

public RadisCacheUtility()

{

var host = ConfigurationManager.AppSettings["RadisHost"].ToString();

var port = Convert.ToInt32(ConfigurationManager.AppSettings["port"]);

\_redisEndpoint = new RedisEndpoint(host, port);

}

public bool IsKeyExists(string key)

{

using (var redisClient = new RedisClient(\_redisEndpoint))

{

if (redisClient.ContainsKey(key))

{

return true;

}

else

{

return false;

}

}

}

public void SetStrings(string key, string value)

{

using (var redisClient = new RedisClient(\_redisEndpoint))

{

redisClient.SetValue(key, value);

}

}

public string GetStrings(string key)

{

using (var redisClient = new RedisClient(\_redisEndpoint))

{

return redisClient.GetValue(key);

}

}

public bool StoreList<T>(string key, T value, TimeSpan timeout)

{

try

{

using (var redisClient = new RedisClient(\_redisEndpoint))

{

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

}

return true;

}

catch (Exception)

{

throw;

}

}

public T GetList<T>(string key)

{

T result;

using (var client = new RedisClient(\_redisEndpoint))

{

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

result = wrapper.GetValue(key);

}

return result;

}

public long Increment(string key)

{

using (var client = new RedisClient(\_redisEndpoint))

{

return client.Increment(key, 1);

}

}

public long Decrement(string key)

{

using (var client = new RedisClient(\_redisEndpoint))

{

return client.Decrement(key, 1);

}

}

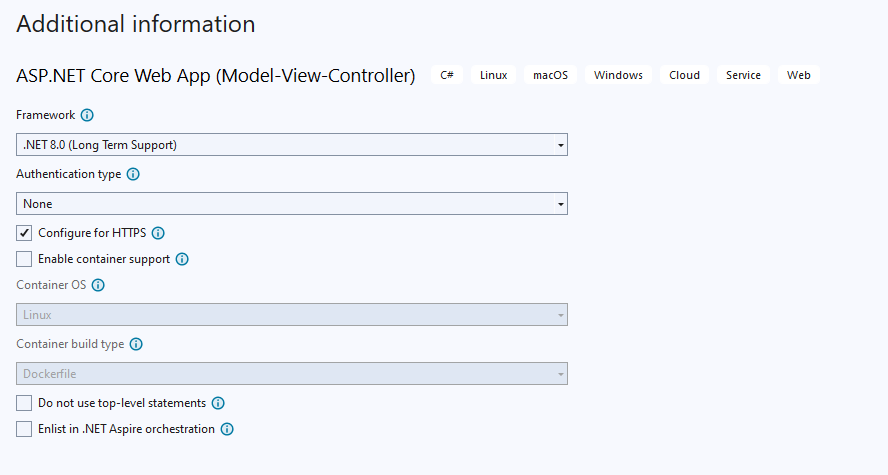
}

🧰 Step 1: Install Redis Locally (Without Docker)

**🔴 On Windows:**

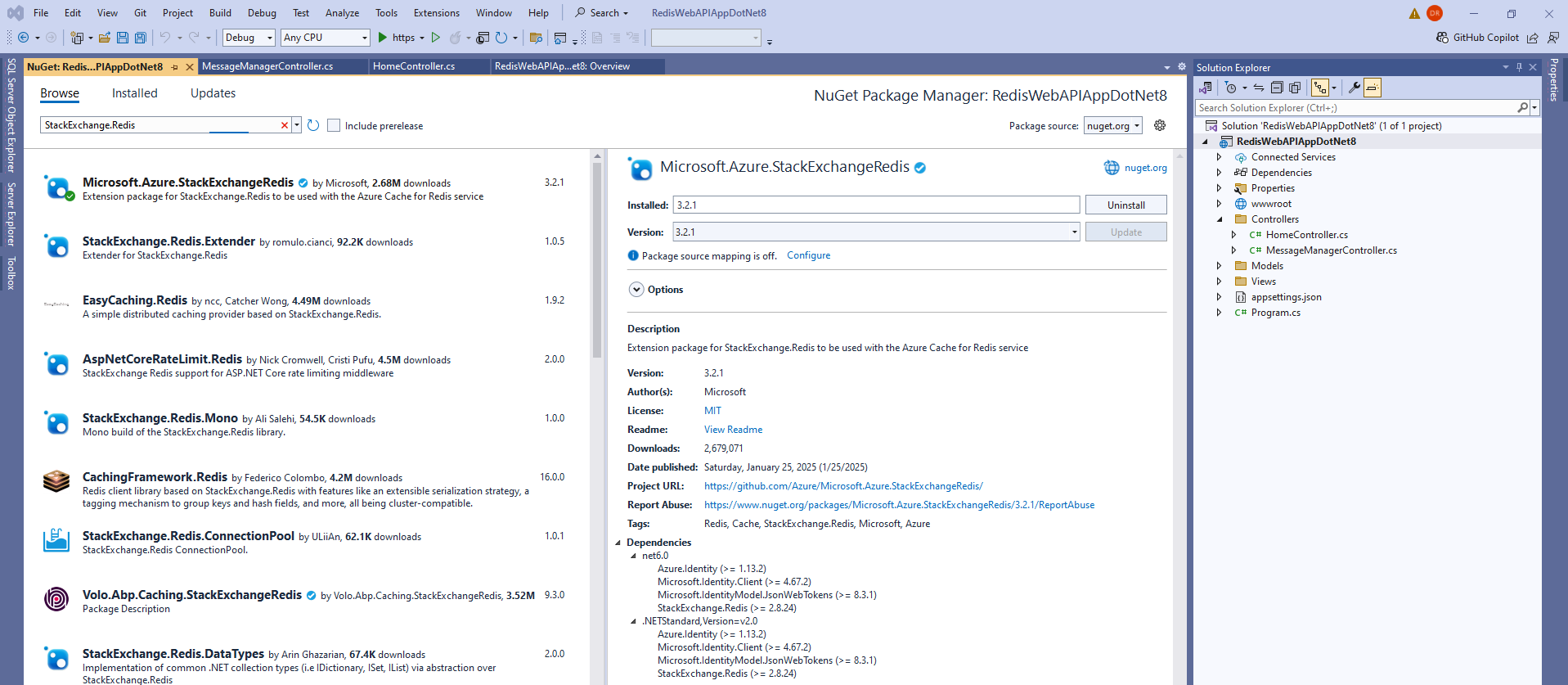
1. Visit the Redis download page.
2. Download the **Redis Windows executable** from the Microsoft archive.
3. Extract and run redis-server.exe.
4. Optionally, install it as a Windows service using redis-server --service-install.

🌐 Step 2: Create a .NET 8 Web API Project



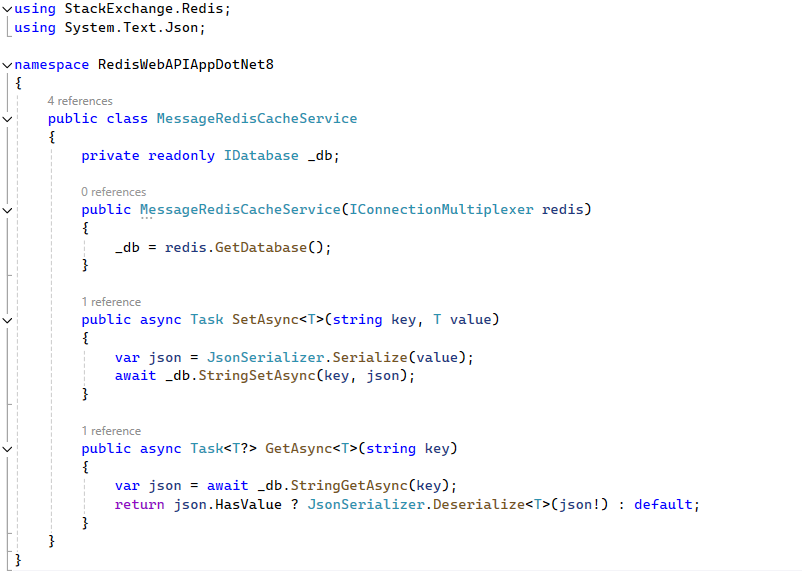
📦 Step 3: Add Redis Client Library

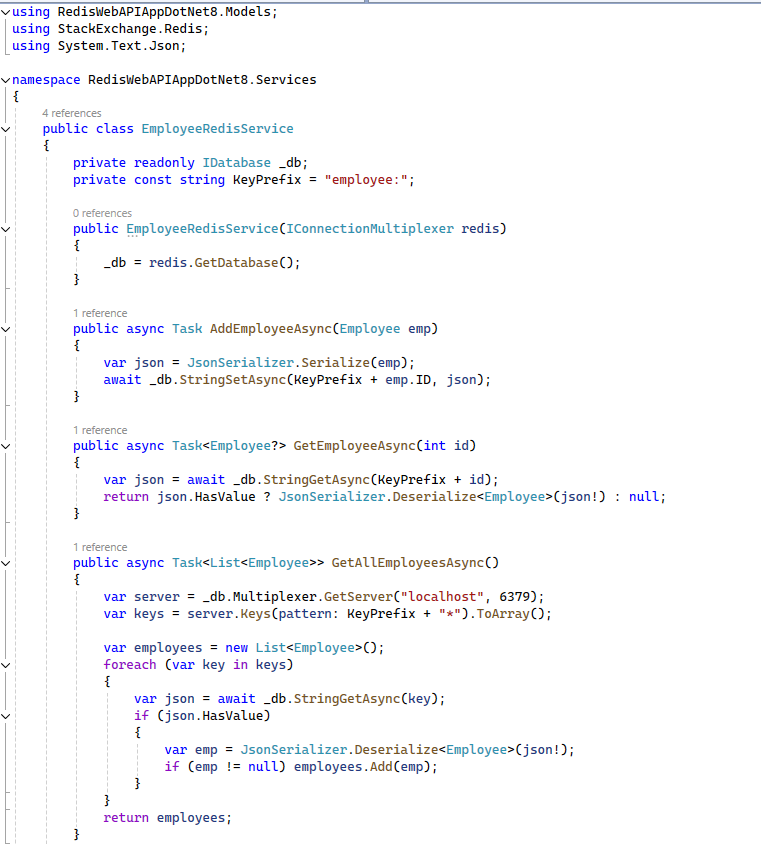
Install the StackExchange.Redis package:

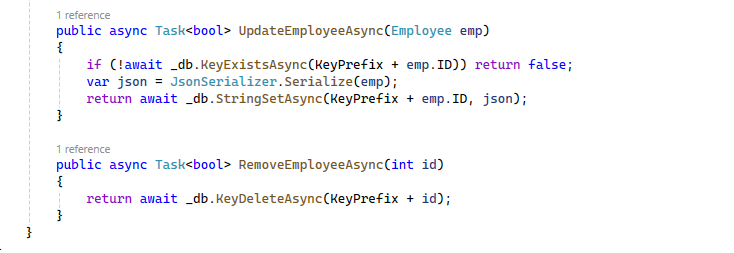


🧩 Step 5: Create a Redis Service

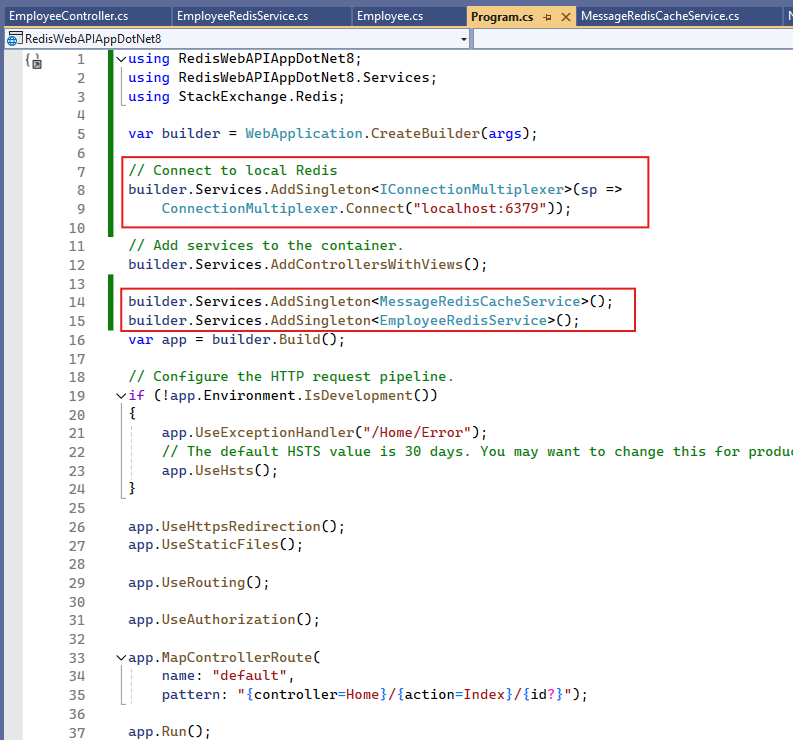
Create RedisCacheService.cs:



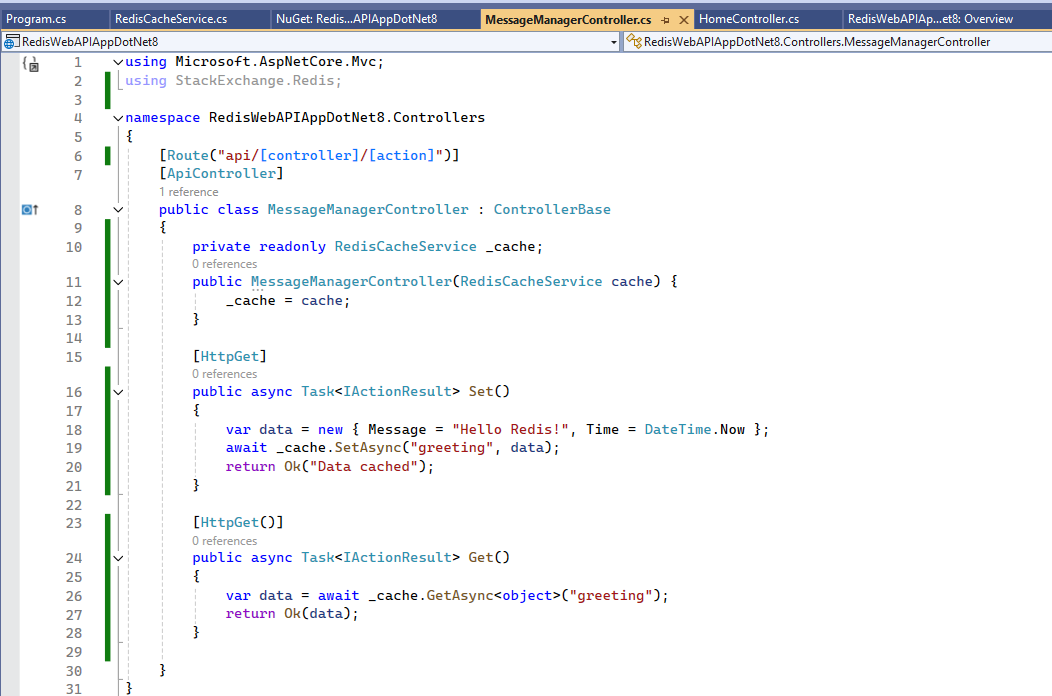




⚙️ Step 4: Configure Redis in Program.cs



🚀 Step 6: Use Redis in a Controller



Set

