

dotnet

.NET ???

- [Installation](#)
- [Linux ????](#)
- [NuGet](#)
- [Sample Codes](#)

Installation

- [? Linux ???????? .NET - .NET | Microsoft Learn](#)
- [How To Install Microsoft .NET Core SDK On Linux](#)

RedHat 8

- <https://docs.microsoft.com/zh-tw/dotnet/core/install/linux-rhel>
- https://access.redhat.com/documentation/en-us/net/6.0/html/getting_started_with_.net_on_rhel_8/index

.NET 6.0 ?????????? AppStream ??????

?? SDK ?????

```
sudo yum install dotnet-sdk-6.0 -y
```

?? runtime ?????

```
sudo dnf install aspnetcore-runtime-6.0
```

????

```
dotnet --list-sdks  
dotnet --list-runtimes
```

Ubuntu 22.04

- [.NET 6 is now in Ubuntu 22.04](#)

Ubuntu 20.04

- [Getting started with cross-platform development using .NET on Ubuntu on WSL | Ubuntu](#)

Linux ?????

Create new project

```
dotnet new console --output my-app
dotnet run --project my-app
```

Add Package from NuGet

```
# The internet is required
cd /path/to/your/project
dotnet add package MySql.Data

# Without the internet
# Download the package from https://www.nuget.org/
dotnet add package MySql.Data --source <path-to-package>

# Verify the package installer
dotnet list package
```

Publishing Applications

```
# Publish the framework-dependent application
dotnet publish my-app.csproj -f net6.0 -c Release

# Optional: If the application is for RHEL only, trim out the dependencies needed for other
platforms
# Replace architecture based on the platform you are using
# - For Intel: x64
# - For IBM Z and LinuxONE: s390x
# - For 64-bit Arm: arm64
dotnet publish my-app -f net6.0 -c Release -r rhel.8-architecture --self-contained false
```

NuGet

- [??????? NuGet ???? | Microsoft Learn](#)

NuGet Server

- [NuGet Server](#)
- [Running a NuGet Server on Docker | dotnetthoughts](#)

Porting NuGet Package

- [.NET Core NuGet Package Offline Restore\(????\)](#)
- [.NET Core NuGet Package Offline Restore\(????\) - ?](#)

NuGet Export

```
# 1. 00 Internet 0000
# 2. 000 packages 000000 dotnet 00
cd <project-root>
mkdir packages
dotnet restore --packages ./packages
```

NuGet Import

- ?????????????????? nuget.org
- ???????? project ??????????????nuget package ???? restore ?????/????????????????
package ?????????? `add package` ???????

```
# 000 Internet
cd <project-root>
tar xzf packages.tar.gz
dotnet restore -s ./packages

# 00000 nuget.org
dotnet nuget list source
dotnet nuget disable source "nuget.org"

# 0000 package
```

```
# □□□□□□□□□□
```

```
dotnet add package MySql.Data --version 8.0.32
```

Sample Codes

MySQL Test

```
using System;
using System.Data;
using MySql.Data.MySqlClient;

namespace MySQL_test
{
    class Program
    {
        static void Main(string[] args)
        {
            string connstring =
@"server=example.com;userid=example_user;password=example_password;database=example_database";

            MySqlConnection conn = null;

            try
            {
                conn = new MySqlConnection(connstring);
                conn.Open();

                string query = "SELECT * FROM table_name;";
                MySqlDataAdapter da = new MySqlDataAdapter(query, conn);
                DataSet ds = new DataSet();
                da.Fill(ds, "table_name");
                DataTable dt = ds.Tables["table_name"];

                foreach (DataRow row in dt.Rows)
                {
                    foreach (DataColumn col in dt.Columns)
                    {
                        Console.Write(row[col] + "\t");
                    }
                }
            }
        }
    }
}
```



```

[Argument('p', "password", "Password")]
private static string _passWord { get; set; }

[Argument('v', "virtualhost", "Virtualhost")]
private static string _virtualHost { get; set; }

[Argument('c', "queueCNT", "Queue Counts.")]
public static int _queueCNT { get; set; }

[Argument('q', "queueName", "Queue Name.")]
public static string _queueName { get; set; }

[Argument('t', "queueType", "Queue Type.")]
public static string _queueType { get; set; }

[Argument('m', "mqType", "MQ Type.")]
public static string _rabbitMQType { get; set; }

[Argument('h', "help", "Show the help.")]
public static bool Help { get; set; }

static void Main(string[] args)
{
    Arguments.Populate();
    if (Help)
    {
        ShowHelp();

        // It guarantees that you will not proceed because you asked to show help.
        return;
    }
    Console.WriteLine("=====");
    Console.WriteLine("Hostname: " + _hostName);
    Console.WriteLine("Port: " + _port);
    Console.WriteLine("Username: " + _userName);
    Console.WriteLine("Password: " + _passWord);
    Console.WriteLine("Virtualhost: " + _virtualHost);
    Console.WriteLine("QueueCNT: " + _queueCNT);
    Console.WriteLine("Queuename: " + _queueName);
    Console.WriteLine("Queuetype: " + _queueType);
}

```

```
Console.WriteLine("RabbitMQType: " + _rabbitMQType);
Console.WriteLine("=====");

private static void ShowHelp()
{
    var helpAttributes = Arguments.GetArgumentInfo(typeof(Program));

    Console.WriteLine("Short\tLong\tFunction");
    Console.WriteLine("-----\t-----\t-----");

    foreach (var item in helpAttributes)
    {
        var result = "-" + item.ShortName + "\t" + "--" + item.LongName + "\t" +
item.HelpText;
        Console.WriteLine(result);
    }
}
```