

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");
                    }
                }
            }
        }
    }
}
```

```

        Console.WriteLine("\n");
    }
}
catch (Exception e)
{
    Console.WriteLine("Error: {0}", e.ToString());
}
finally
{
    if (conn != null)
    {
        conn.Close();
    }
}
}
}
}

```

Console App ?????

- [GitHub - jpdillingham/Utility.CommandLine.Arguments](https://github.com/jpdillingham/Utility.CommandLine.Arguments)

Program.cs:

```

using System;
using System.Collections.Generic;
using System.Linq;
using Utility.CommandLine;

namespace Console_RabbitMQ
{
    class Program
    {
        [Argument('n', "hostname", "Host Names or IPs")]
        private static string _hostName { get; set; }

        [Argument('P', "Port", "Server Port")]
        private static int _port { get; set; }

        [Argument('u', "username", "Username")]

```

```

private static string _userName { get; set; }

[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);
    }
}
```

Revision #5

Created 10 February 2023 17:23:26 by Admin

Updated 22 February 2023 14:05:25 by Admin