



# EducaCiência FastCode

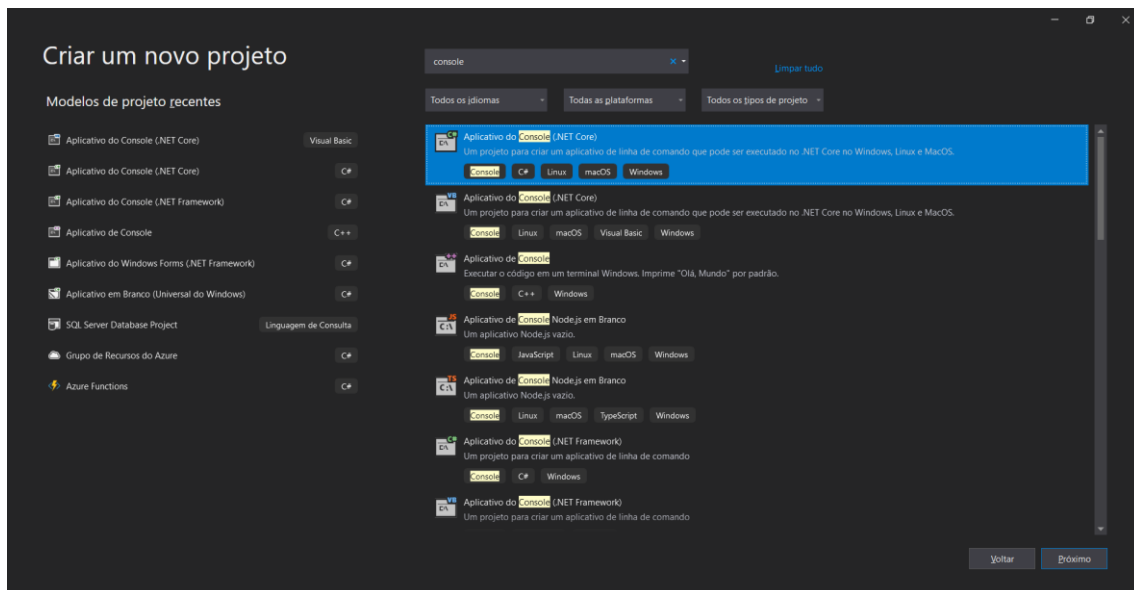
Fala Galera,

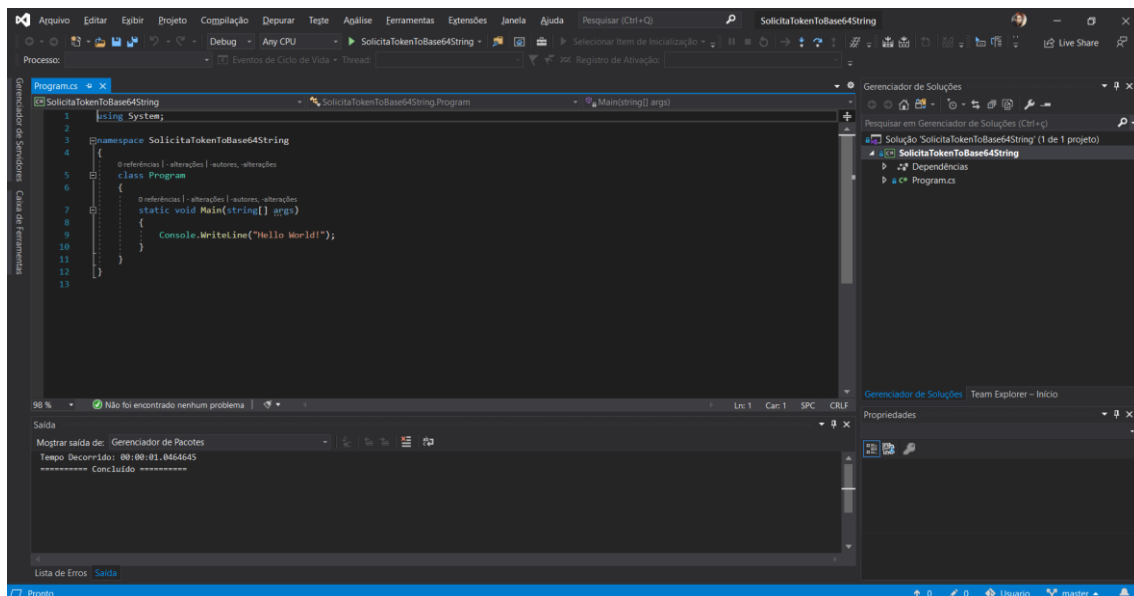
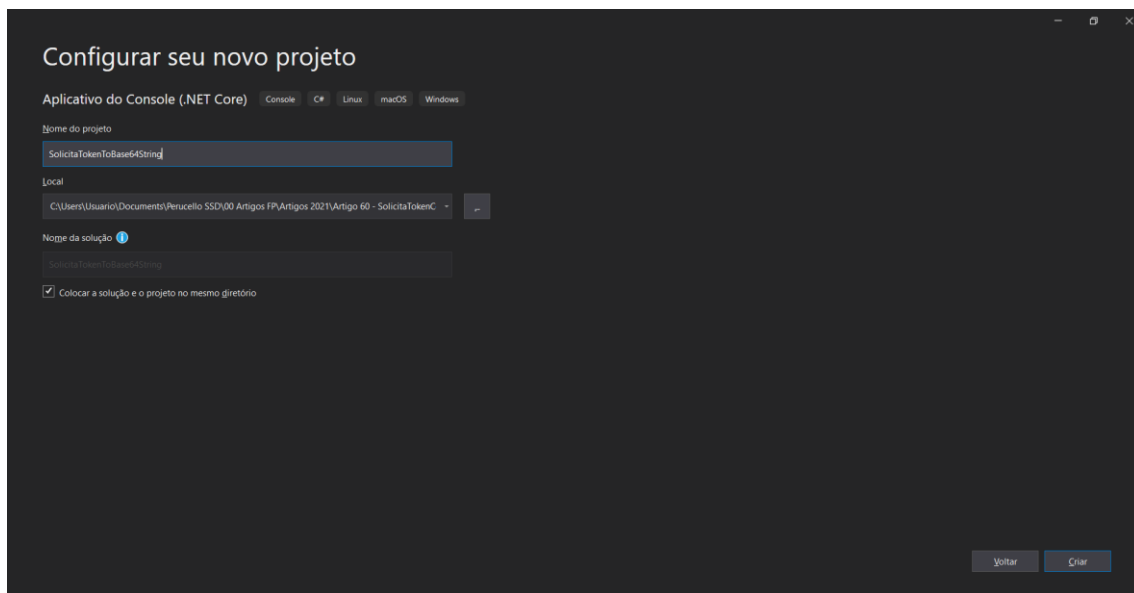
- Artigo: 60/2021 Data: Julho/2021
- Público-alvo: Desenvolvedores – Iniciantes
- Tecnologia: C#
- Tema: Artigo 60 - SolicitaTokenContent\_Type\_ConvertBase64\_CSharp -> Console .netCore
- Link: <https://github.com/perucello/DevFP>

Neste artigo, criaremos uma classe para fazermos solicitação de token em uma API externa , neste cenário, passamos o usuário e a senha para obter o token, porém nossa senha necessitará de ser convertida “`ToBase64String`”.

Lembrando que os fins são didáticos !

Vamos partir da premissa que será desenvolvido, portanto, iremos criar nosso projeto, para isso defina seu projeto como Console .netCore .





Com nosso projeto aberto, no nosso Program.cs , temos nosso método Main. Teremos fatores de autenticação para nossa requisição, nesse cenário temos:

- ⇒ **Usuário** -> usuário para a requisição
- ⇒ **senhaUsuario** -> senha do usuário de acesso
- ⇒ **endpointRequestToken** -> endpoint para requisição Post para recebermos o token

Para nossa solicitação, utilizaremos da classe **HttpClient** - vide documentação oficial no site:

⇒ <https://docs.microsoft.com/pt-br/dotnet/api/system.net.http.httpclient?view=net-5.0>

Com estas informações passadas, vamos ao código:



```
using System;
using System.IO;
using System.Net;
using System.Net.Http;
using System.Text;

namespace SolicitaTokenToBase64String
{
    public class Program
    {
        static void Main(string[] args)
        {
            try
            {
                string usuario = ""; // insira o Usuario
                string senhaUsuario = ""; //Insira a senha - codigifcada
                string endpointRequestToken = ""; //insira o endpoint

                var client = new HttpClient();
                String username = usuario;
                String password = senhaUsuario;

                //Convert to Base 64 da senha
                String encoded = System.Convert.ToBase64String(System.Text.Encoding.GetEncoding("ISO-8859-1").GetBytes(username + ":" + password));

                //Validação
                var formData = "var1=val1&var2=val2";
                var encodedFormData = Encoding.ASCII.GetBytes(formData);

                //Criando request ao Endpoint
                var request = (HttpWebRequest)WebRequest.Create(endpointRequestToken);

                //Application
                request.ContentType = "application/x-www-form-urlencoded";
                request.Method = "POST";
                request.ContentLength = encodedFormData.Length;

                //Informando que Authorization é Basic Auth
                request.Headers.Add("Authorization", "Basic " + encoded);
                request.PreAuthenticate = true;

                var token = client + endpointRequestToken;
                using (var stream = request.GetRequestStream())
                {
                    stream.Write(encodedFormData, 0, encodedFormData.Length);
                }
                HttpWebResponse response = (HttpWebResponse)request.GetResponse();
                var responseString = new StreamReader(response.GetResponseStream()).ReadToEnd();

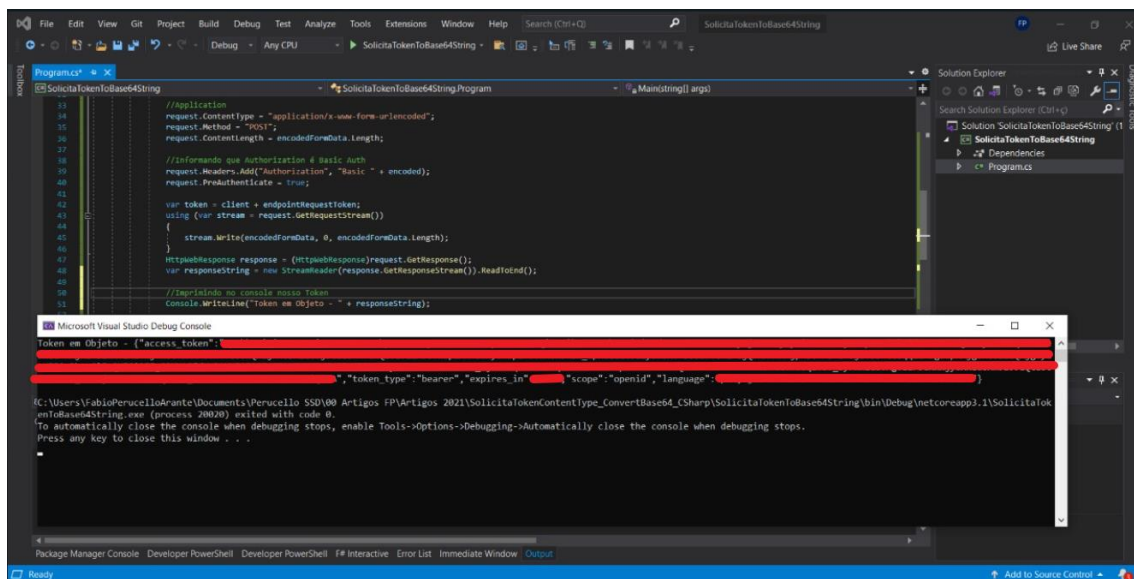
                //Imprimindo no console nosso Token
                Console.WriteLine("Token em Objeto - " + responseString);

            }
            catch (Exception e)
            {
                Console.WriteLine($"Exception Catch SolicitaToken : {e} ");
                Console.WriteLine(e);
                Console.WriteLine("Http Status: " + HttpStatusCode.NotFound);

                throw new Exception($"Exception Catch SolicitaToken : {e} ");
            }
        }
    }
}
```

Lembrando que você deverá preencher os requisitos de acordo com seu acesso, dito isso, foi adicionado nossa credencial e iremos iniciar nossa aplicação para comprovar o resultado – vide abaixo:





Nesse cenário, temos o retorno do json completo !

Podemos ainda ajustar caso nosso token for ser chamado em outra solicitação do tipo Bearer.

Para isso, ajustamos o código e apresentamos a seguir.

```

using System;
using System.IO;
using System.Net;
using System.Net.Http;
using System.Text;
    
```

```

namespace SolicitaTokenToBase64String
    
```

```

{
    public class Program
    {
        static void Main(string[] args)
        {
            try
            {
                string usuario = ""; // insira o Usuario
                string senhaUsuario = ""; //Insira a senha - codigifcada
                string endpointRequestToken = ""; //insira o endpoint

                var client = new HttpClient();
                String username = usuario;
                String password = senhaUsuario;

                //Convert to Base 64 da senha
                String encoded = System.Convert.ToBase64String(System.Text.Encoding.GetEncoding("ISO-8859-1").GetBytes(username + ":" + password));

                //Validação
                var formData = "var1=val1&var2=val2";
                var encodedFormData = Encoding.ASCII.GetBytes(formData);

                //Criando request ao Endpoint
                var request = (HttpWebRequest)WebRequest.Create(endpointRequestToken);

                //Application
                request.ContentType = "application/x-www-form-urlencoded";
                request.Method = "POST";
                request.ContentLength = encodedFormData.Length;

                //Informando que Authorization é Basic Auth
                request.Headers.Add("Authorization", "Basic " + encoded);
                request.PreAuthenticate = true;

                var token = client + endpointRequestToken;
                using (var stream = request.GetRequestStream())
                {
                    stream.Write(encodedFormData, 0, encodedFormData.Length);
                }
            }
        }
    }
}
    
```



```

HttpWebResponse response = (HttpWebResponse)request.GetResponse();
var responseString = new StreamReader(response.GetResponseStream()).ReadToEnd();

//Imprimindo no console nosso Token
//Console.WriteLine("Token em Objeto - " + responseString);

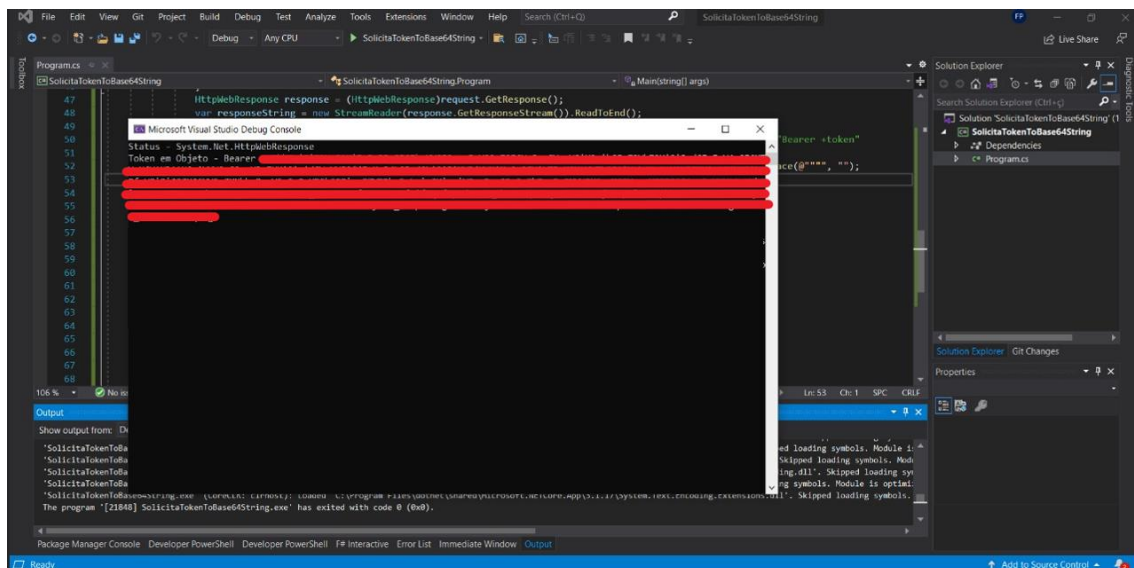
//Tratando o retorno do json do token para que fique descrito da maneira correta - neste
cenário ficará "Bearer +token"
string tratamento = responseString.Substring(0, 609);
string tokenAcess = tratamento.Replace("access_token", "Bearer ").Replace("{",
"").Replace(":", "").Replace("@\"", "");

//Imprimindo no console nosso Token
Console.WriteLine("Token em Objeto - " + tokenAcess);

}
//Tratando Exceção genérica.
catch (Exception e)
{
    Console.WriteLine($"Exception Catch SolicitaToken : {e} ");
    Console.WriteLine(e);
    Console.WriteLine("Http Status: " + HttpStatusCode.NotFound);

    throw new Exception($"Exception Catch SolicitaToken : {e} ");
}
}
}
}

```



Com isso concluímos nossa proposta.  
Até mais !  
Espero ter ajudado !

