



EducaCiência FastCode

Fala Galera,

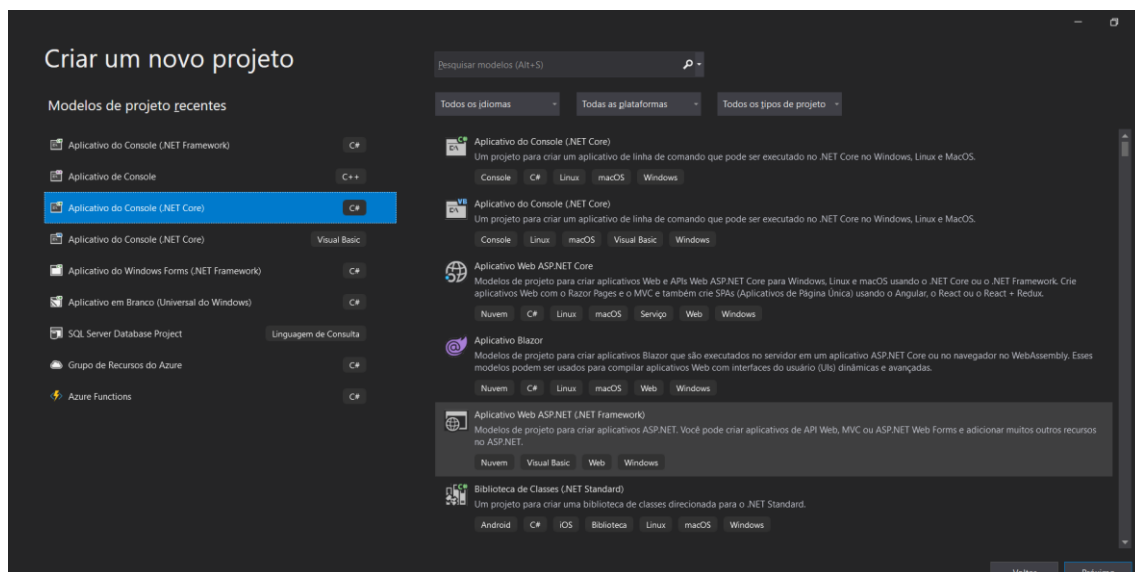
- Artigo: 59/2021 Data: Julho/2021
- Público-alvo: Desenvolvedores – Iniciantes
- Tecnologia: C#
- Tema: Artigo 59 - SolicitaTokenFormUrlEncodedContent_CSharp -> Console .netCore
- Link: <https://github.com/perucello/DevFP>

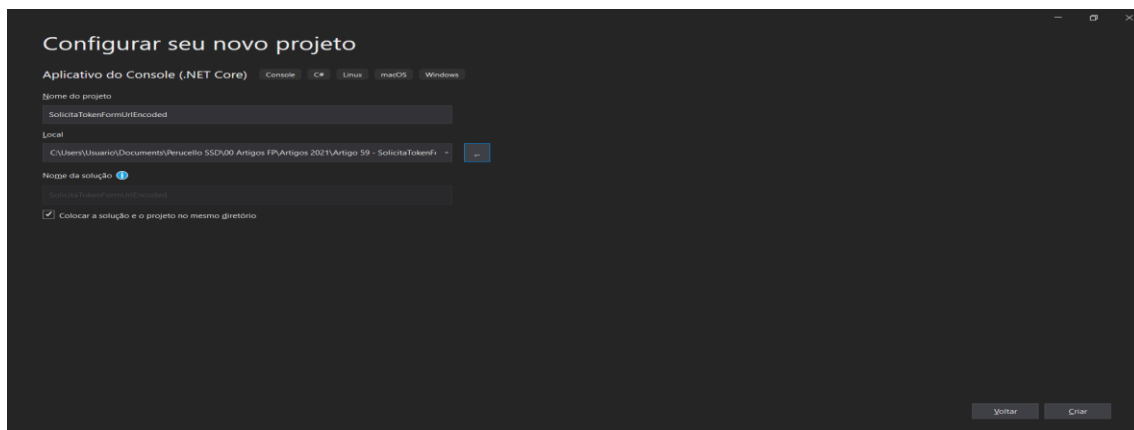
Neste artigo, criaremos uma classe para fazermos solicitação de token em uma API externa , utilizaremos parâmetro – **FormUrlEncodedContent**.

Neste cenário, teremos um formulário de autenticação para que possamos realizar nossa requisição e obter token.

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
- ⇒ **clientId** => ClientID de acesso
- ⇒ **clientSecret** -> ClientSecret de acesso
- ⇒ **serviceToken** -> ServiceToken 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.Collections.Generic;
using System.IO;
using System.Net;
using System.Net.Http;

namespace SolicitaTokenFormUrlEncoded
{
    public class Program
    {
        static async System.Threading.Tasks.Task Main(string[] args)
        {
            string usuario = ""; //Usuario
            string senhaUsuario = ""; //SenhaUsuario
            string clientId = ""; // senha ClientID
            string clientSecret = ""; //Senha ClientSecret
            string serviceToken = ""; //Senha serviceToken
            string endpointRequestToken = "https://"; // Endpoint para solicitar token
            try
            {
                //Classe HttpClient
                var client = new HttpClient();

                //Parametros/Formulario para requisição do Token Access
                var requestContent = new FormUrlEncodedContent(new[] {
                    new KeyValuePair<string, string>("username", usuario),
                    new KeyValuePair<string, string>("password", senhaUsuario),
                    new KeyValuePair<string, string>("client_id", clientId),
                    new KeyValuePair<string, string>("client_secret", clientSecret),
                    new KeyValuePair<string, string>("grant_type", serviceToken),
                });
            }
        }
    }
}
```





```
});

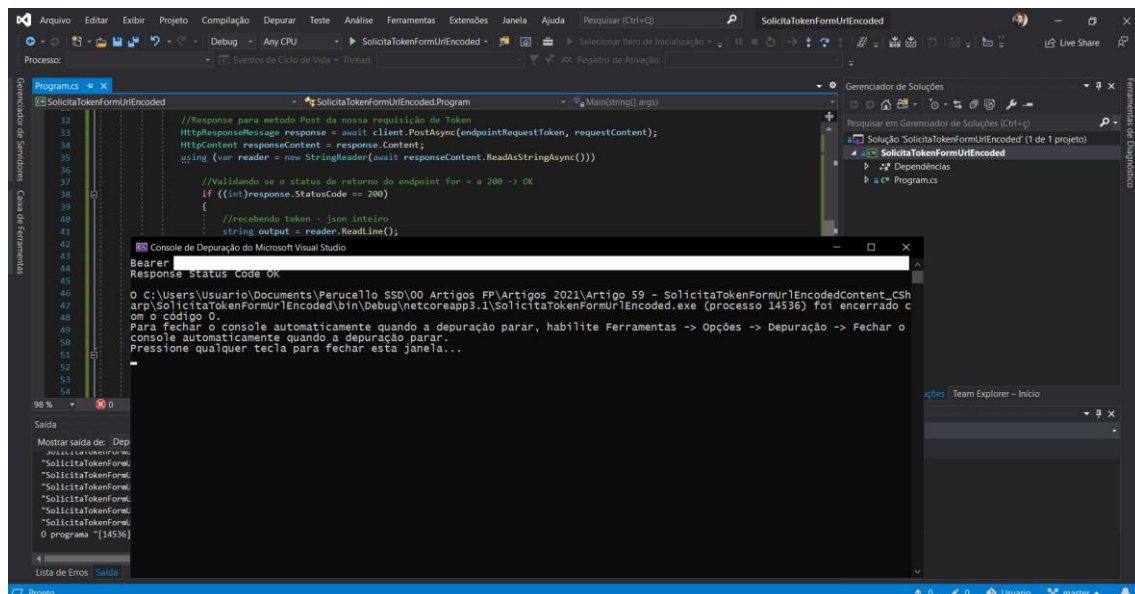
//Resposta para metodo Post da nossa requisição de Token
HttpResponseMessage response = await client.PostAsync(endpointRequestToken, requestContent);
HttpContent responseContent = response.Content;
using (var reader = new StringReader(await responseContent.ReadAsStringAsync()))

//Validando se o status de retorno do endpoint for = a 200 -> OK
if ((int)response.StatusCode == 200)
{
    //recebendo token - json inteiro
    string output = reader.ReadLine();

    //Para Tratar o retorno Json do Token
    string tratamento = output.Substring(0, 130);
    string token = tratamento.Replace("access_token", "Bearer ").Replace("{",
    "").Replace(":", "").Replace("@\"", "");

    Console.WriteLine("Recebendo Json inteiro da Requisição do Token -> " + output);
    Console.WriteLine("Recebendo Token tratado -> " + token);
    Console.WriteLine("Response Status Code " + HttpStatusCode.OK);
    //return token;
}
else
{
    string output = "Falha não foi possível gerar Token ! ";
    //return output;
}
}
catch (Exception e)
{
    Console.WriteLine($"Exception Catch: {e} ");
    Console.WriteLine("Http Status: " + HttpStatusCode.NotFound);
    throw new Exception($"Erro: " + " " + 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:



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

