

Como chamar uma API externa a partir do agilityflow

Este exemplo mostra como fazer uma requisição HTTP `POST` para uma API externa no contexto do AgilityFlow.

Importante sobre `CancellationToken`

O método `PostAsync` **deve sempre utilizar** um `CancellationToken` para garantir que a requisição seja cancelável, evitando problemas em fluxos interrompidos ou expirados.

Para recuperar o `CancellationToken`, utilize o método adequado ao **contexto em que o código está sendo executado**. Alguns exemplos:

- Em **regras de negócio** do formulário:
`FormContext.GetCurrentCancellationToken()`
- Em chamadas feitas dentro de uma **API customizada**:
`ApiContext.GetCurrentCancellationToken()`
- Em **componentes Razor (.cshtml)** do formulário:
`PageContext.GetCurrentCancellationToken()`
- E assim por diante, conforme o contexto (`GlobalContext`, `CustomPageContext`, `CardContext`, etc.)

“ Basta utilizar o método `GetCurrentCancellationToken()` do contexto atual.

Observação

- Troque `<<URL_DA_API>>` pela URL real da API que deseja acionar.

Como chamar a api em cada contexto?

No FormContext

Crie uma regra de negócio e dentro da regra de Programação em C#, utilize o código abaixo

```
public async Task ExecuteAsync()
{
    var formId = FormContext.FormId.Value.ToString();
    await CallExternalApiAsync(formId);
}

public async Task CallExternalApiAsync(string qualquerParametro)
{
    if (string.IsNullOrEmpty(qualquerParametro)) return;

    var payload = new
    {
        parametroParaApi = qualquerParametro
    };

    using (var httpClient = new HttpClient())
    {
        var content = new StringContent(
            JsonConvert.SerializeObject(payload),
            Encoding.UTF8,
            "application/json"
        );

        await httpClient.PostAsync("<<URL_DA_API>>", content, FormContext.GetCurrentCancellationTokens());
    }
}
```

Na ApiContext

Na programação em C#

```

public async Task RunAsync()
{
    await CallExternalApiAsync("exemplo de parametro");
}

public async Task CallExternalApiAsync(string qualquerParametro)
{
    if (string.IsNullOrEmpty(qualquerParametro)) return;

    var payload = new
    {
        parametroParaApi = qualquerParametro
    };

    using (var httpClient = new HttpClient())
    {
        var content = new StringContent(
            JsonConvert.SerializeObject(payload),
            Encoding.UTF8,
            "application/json"
        );

        await httpClient.PostAsync("<<URL_DA_API>>", content, ApiContext.GetCurrentCancellationTokens());
    }
}

```

No PageContext dos componentes Cshtml do formulário

Na programação em C# razor dos componente

```

@{
    string qualquerParametro = "";
    if (!string.IsNullOrEmpty(qualquerParametro)){
        var payload = new
        {
            parametroParaApi = qualquerParametro
        };
    }
}

```

```
using (var httpClient = new HttpClient())
{
    var content = new StringContent(
        JsonConvert.SerializeObject(payload),
        Encoding.UTF8,
        "application/json"
    );

    await httpClient.PostAsync("<<URL_DA_API>>", content, PageContext.GetCurrentCancellationTokens());
}
}
```

No GlobalContext

Na programação em C#

```
public async Task CallExternalApiAsync(string qualquerParametro)
{
    if (string.IsNullOrEmpty(requestId)) return;

    var payload = new
    {
        parametroParaApi = qualquerParametro
    };

    using (var httpClient = new HttpClient())
    {
        var content = new StringContent(
            JsonConvert.SerializeObject(payload),
            Encoding.UTF8,
            "application/json"
        );

        await httpClient.PostAsync("<<URL_DA_API>>", content, GlobalContext.GetCurrentCancellationTokens());
    }
}
```

No CustomPageContext

Na programação em C# do custom page context

```
public async Task CallExternalApiAsync(string qualquerParametro)
{
    if (string.IsNullOrEmpty(requestId)) return;

    var payload = new
    {
        parametroParaApi = qualquerParametro
    };

    using (var httpClient = new HttpClient())
    {
        var content = new StringContent(
            JsonConvert.SerializeObject(payload),
            Encoding.UTF8,
            "application/json"
        );

        await httpClient.PostAsync("<<URL_DA_API>>", content,
CustomPageContext.GetCurrentCancellationTokens());
    }
}
```

Revision #11

Created 27 May 2025 22:13:55 by agilityflow

Updated 14 July 2025 12:44:51 by agilityflow