

Executar uma Api do proprio agilityflow dentro de um Form ou dentro de uma outra Api

No FormContext

Chamando uma Api Post e **não** aguardando a resposta **(Assincronamente)**.

```
public async Task ExecuteAsync(){  
  
    await FormContext.InternalApi.ExecutePostAsynchronouslyAsync("rest-api/teste",  
        new { valorr= 334455.66, email="exemplo@email.com.br" },  
        new ExecutePostAsynchronouslyOptions() {  
            AuthUser = "MUDAR AQUI PELO API USER LOGIN",  
            AuthKey = "MUDAR AQUI PELO API USER KEY",  
        });  
  
}
```

Chamando uma Api Post e aguardando a resposta **(Sincronamente)**.

```

public async Task ExecuteAsync(){

    var result = await FormContext.InternalApi.ExecutePostSynchronouslyAsync("rest-api/teste",
        new { valorr= 33445566.77, email="exemplo@email.com.br" },
        new ExecutePostSynchronouslyOptions() {
            AuthUser = "MUDAR AQUI PELO API USER LOGIN",
            AuthKey = "MUDAR AQUI PELO API USER KEY",
        });

    //verificar se foi sucesso ou nao
    var isSuccess = result.IsSuccess;

    //se quiser o retorno no formato string
    var str = result.GetString();

    //se quiser o retorno no formato json
    var json = result.GetJson();

    FormContext.Log($"isSuccess: {isSuccess} | - str: {str}" ,"debug");

}

```

No PageContext

Chamando uma Api Post e **não** aguardando a resposta **(Assincronamente)**.

```

public async Task ExecuteAsync(){

    await PageContext.InternalApi.ExecutePostAsynchronouslyAsync("rest-api/teste",
        new { valorr= 334455.66, email="exemplo@email.com.br" },
        new ExecutePostAsynchronouslyOptions() {
            AuthUser = "MUDAR AQUI PELO API USER LOGIN",
            AuthKey = "MUDAR AQUI PELO API USER KEY",
        });

}

```

```
});  
  
}
```

Chamando uma Api Post e aguardando a resposta (Sincronamente).

```
public async Task ExecuteAsync(){  
  
    var result = await PageContext.InternalApi.ExecutePostSynchronouslyAsync("rest-api/teste",  
        new { valorr= 33445566.77, email="exemplo@email.com.br" },  
        new ExecutePostSynchronouslyOptions() {  
            AuthUser = "MUDAR AQUI PELO API USER LOGIN",  
            AuthKey = "MUDAR AQUI PELO API USER KEY",  
        });  
  
    //verificar se foi sucesso ou nao  
    var isSuccess = result.IsSuccess;  
  
    //se quiser o retorno no formato string  
    var str = result.GetString();  
  
    //se quiser o retorno no formato json  
    var json = result.GetJson();  
  
    FormContext.Log($"isSuccess: {isSuccess} | - str: {str}" ,"debug");  
  
}
```

No ApiContext

IMPORTANTE: Caso o usuário da API que está consumindo esta outra API tenha acesso a ambas, não será necessário fornecer os parâmetros AuthUser e AuthKey.

Chamando uma Api Post e não aguardando a resposta (Assincronamente).

```
public async Task RunAsync(){

    await ApiContext.InternalApi.ExecutePostAsynchronouslyAsync("rest-api/teste",
        new { valorr= 334455.66, email="exemplo@email.com.br" },
        new ExecutePostAsynchronouslyOptions() {

            /* ATENÇÃO: Caso o usuário da API que está consumindo esta outra API
            tenha acesso a ambas, não será necessário
            fornecer os parâmetros AuthUser e AuthKey.*/
            AuthUser = "MUDAR AQUI PELO API USER LOGIN",
            AuthKey = "MUDAR AQUI PELO API USER KEY",
        });

}
```

Chamando uma Api Post e aguardando a resposta (Sincronamente).

```
public async Task RunAsync(){

    var result = await ApiContext.InternalApi.ExecutePostSynchronouslyAsync("rest-api/teste",
        new { valorr= 33445566.77, email="exemplo@email.com.br" },
        new ExecutePostSynchronouslyOptions() {

            /* ATENÇÃO: Caso o usuário da API que está consumindo esta outra API
            tenha acesso a ambas, não será necessário
```

```
fornecer os parâmetros AuthUser e AuthKey.*/  
    AuthUser = "MUDAR AQUI PELO API USER LOGIN",  
    AuthKey = "MUDAR AQUI PELO API USER KEY",  
  
});
```

```
#####/verificar se foi sucesso ou nao
```

```
    var isSuccess = result.IsSuccess;
```

```
#####/se quiser o retorno no formato string
```

```
    var str = result.GetString();
```

```
#####/se quiser o retorno no formato json
```

```
    var json = result.GetJson();
```

```
    FormContext.Log($"isSuccess: {isSuccess} | - str: {str}" , "debug");
```

```
}
```

Revision #10

Created 30 November 2024 00:25:23 by agilityflow

Updated 5 March 2025 12:33:44 by agilityflow