Anton Martyniuk • Following

Microsoft MVP | Helping 30K+ Software Engineers Improve .NET Skills and Craft Better Software |...

View my newsletter

2h •

Are you correctly using HttpClient?

Most devs use it in the wrong way 👇

You call external APIs almost in every project.

⛔ Many devs just use one Singleton HttpClient for all HTTP calls.

And they are completely wrong.

Singleton HttpClient doesn't resolve **DNS and network changes**. I faced an issue where a network failure from the internet provider caused my application to stop sending HTTP requests until it was restarted.

⛔ So maybe just create HttpClient for each call with a using statement?

This is a second common mistake. Sooner or later you can get a **Port Exhaustion** problem. As disposed HttpClient doesn't immediately free up the used Socket under the hood.

Let's explore the 3 ways to correctly use HttpClient:

☑ HttpClientFactory

☑ HttpClientFactory with Typed Clients

☑ HttpClientFactory with Refit library

☑HttpClientFactory contains a pool of HttpClients and caches them. Each time you need an HttpClient - you resolve it from the Factory.

Pros:

• Solves DNS refresh & socket exhaustion

• Central place to configure default policies

• Lightweight, part of Microsoft.Extensions.Http

Cons:

• You need to write boilerplate code to resolve HttpClient before sending each request

• You still pass raw URLs & strings around

☑ Typed HttpClient — you register a named class that gets an HttpClient injected in the constructor.
You encapsulate all the message-sending logic in this class.
Your business code calls the strongly-typed service instead of juggling with HttpClientFactory, URLs and request/response serialization.

Pros:

• Encapsulates endpoints in one class

• Strong typing & DI-friendly

• Easy unit-testing with mocks / handlers

Cons:

• Manual mapping between DTOs & endpoints

• More setup code than Refit

☑ Refit — Refit turns your C# interface into a strongly-typed REST client at runtime.

Pros:

• No boilerplate code for REST calls

• Interface-first design, clear contracts

Cons:

• Adds third-party dependency

How do you manage HttpClient in your projects: HttpClientFactory, Typed Clients, or Refit? Leave a comment down below 👇

✅ If you like this post - **repost** to your network and **follow** me

✅ Download my .NET 2025 Backend Roadmap, learn exactly what top employers are looking for.

Roadmap contains high-quality resources for every topic:

https://lnkd.in/e2FZueef

👍💡 67 44 comments 9 reposts