

Extending HttpClient with Custom HttpMessageHandlers



Kevin Dockx

ARCHITECT

@KevinDockx <https://www.kevindockx.com>



Coming Up



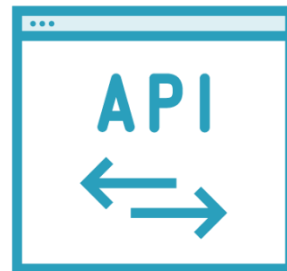
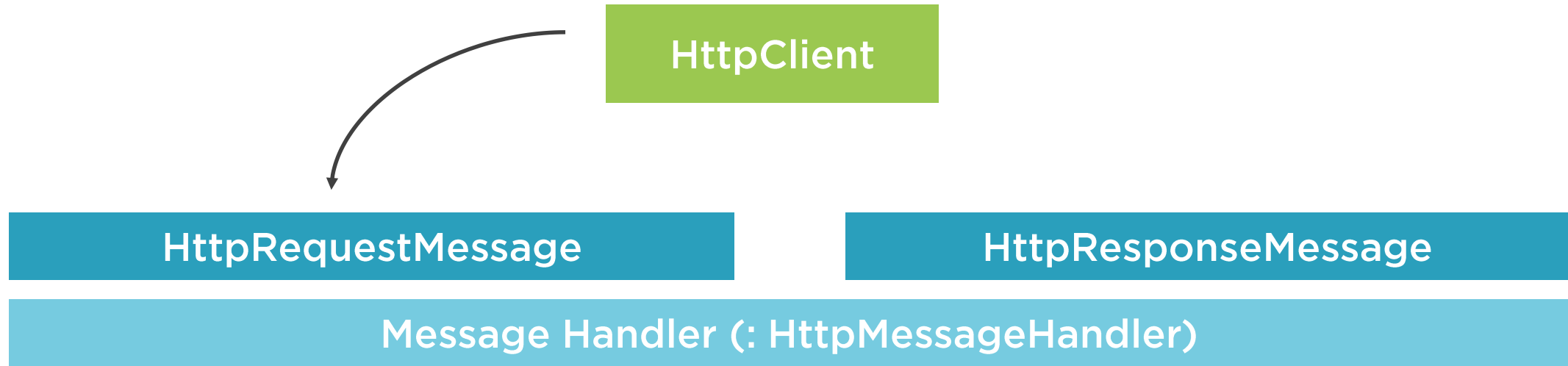
Delegating Handler Pattern

Implementing a Retry Policy and Dealing with Time-outs

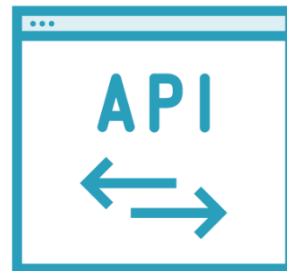
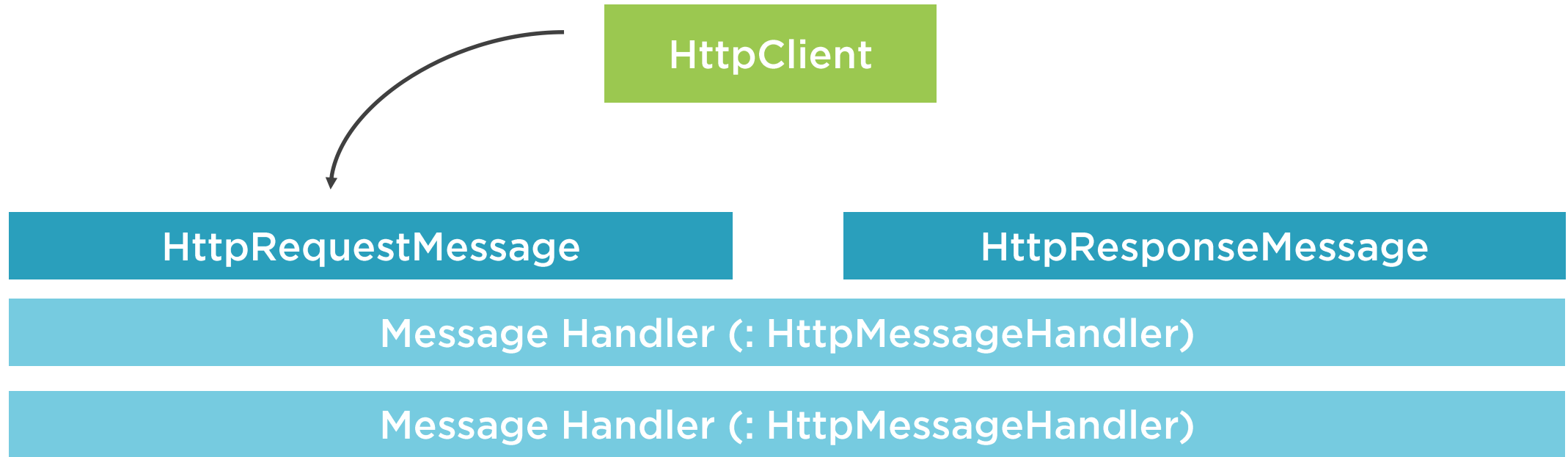
Additional Use Cases



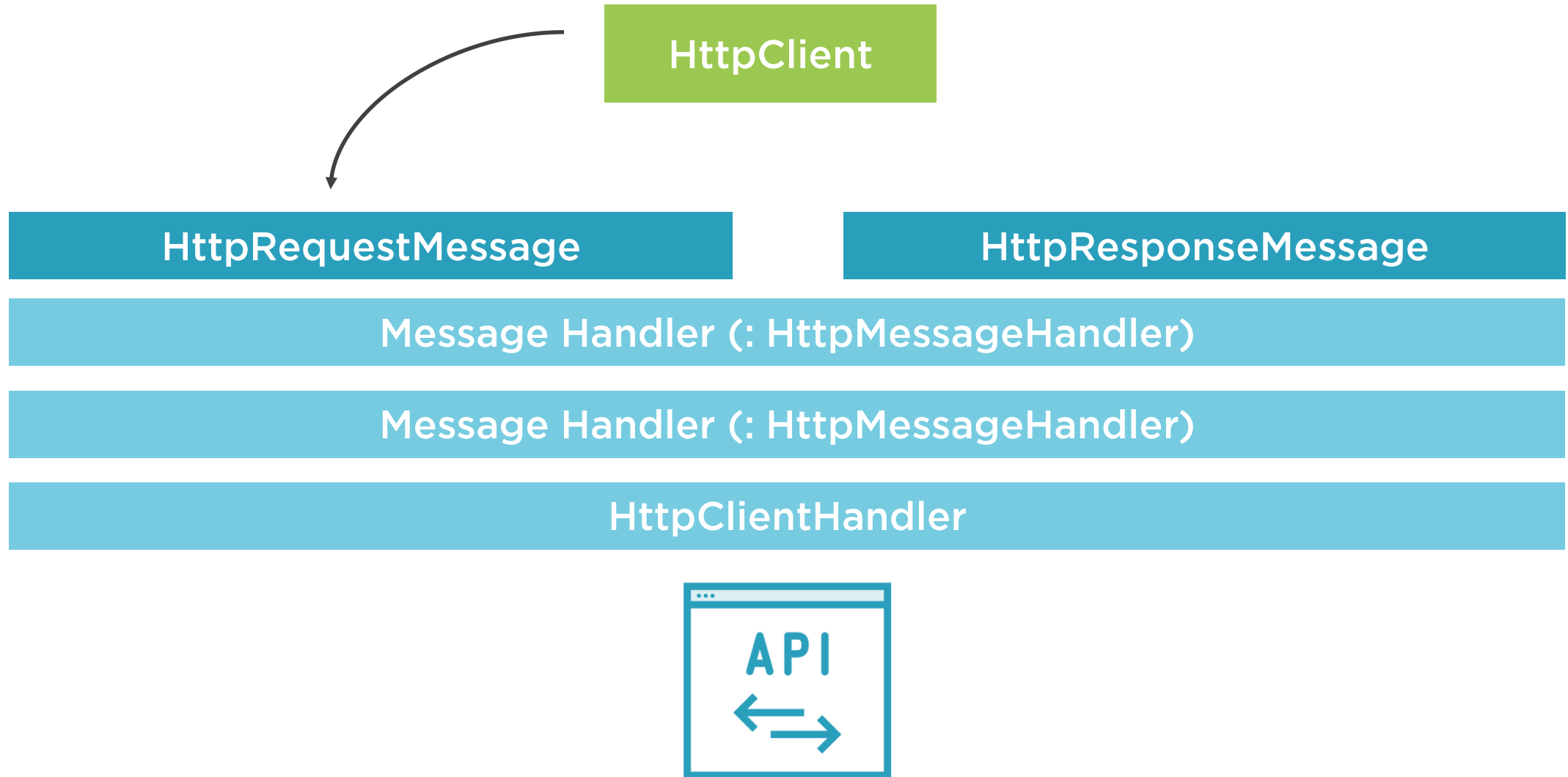
Delegating Handler Pattern



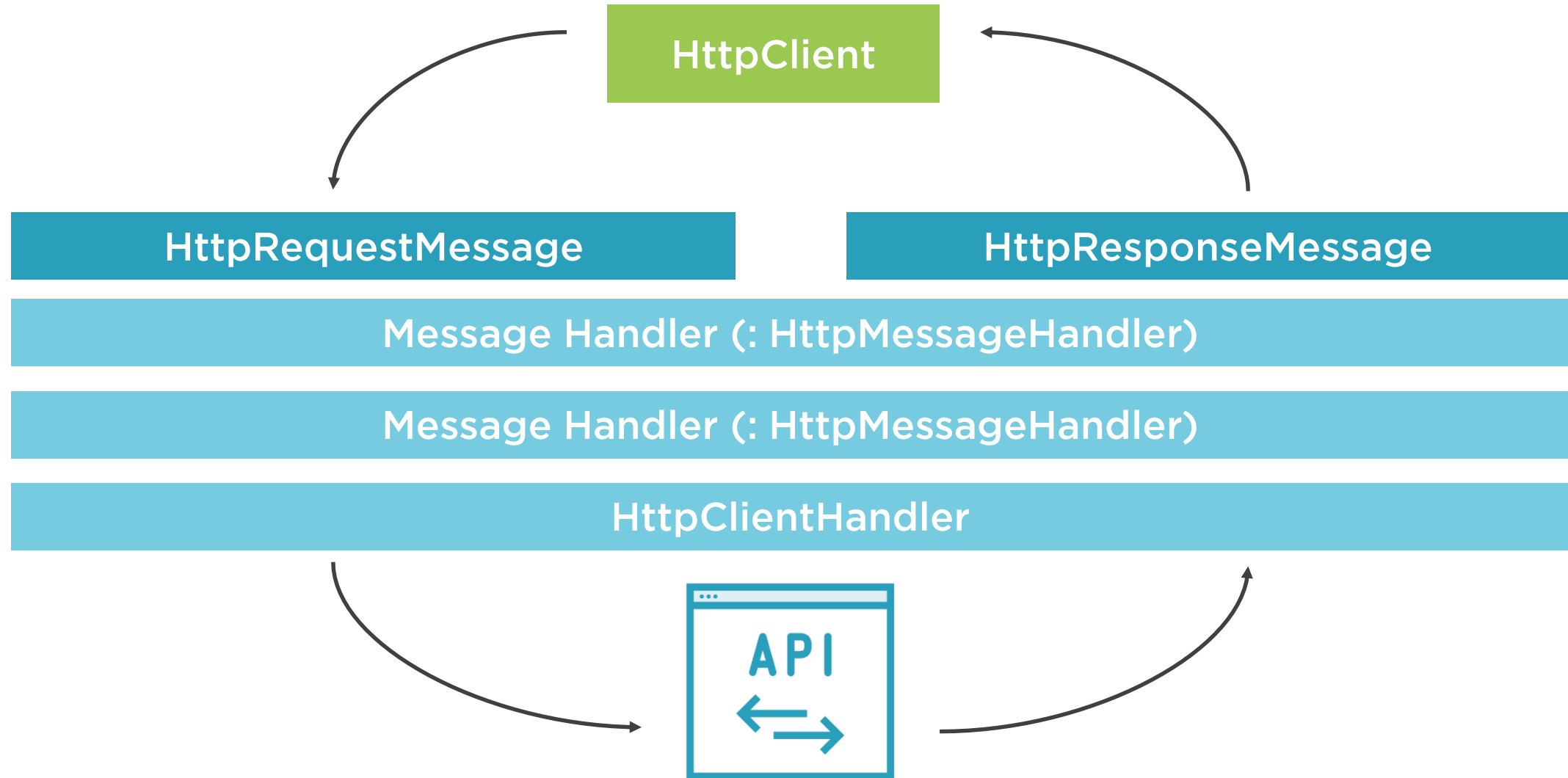
Delegating Handler Pattern



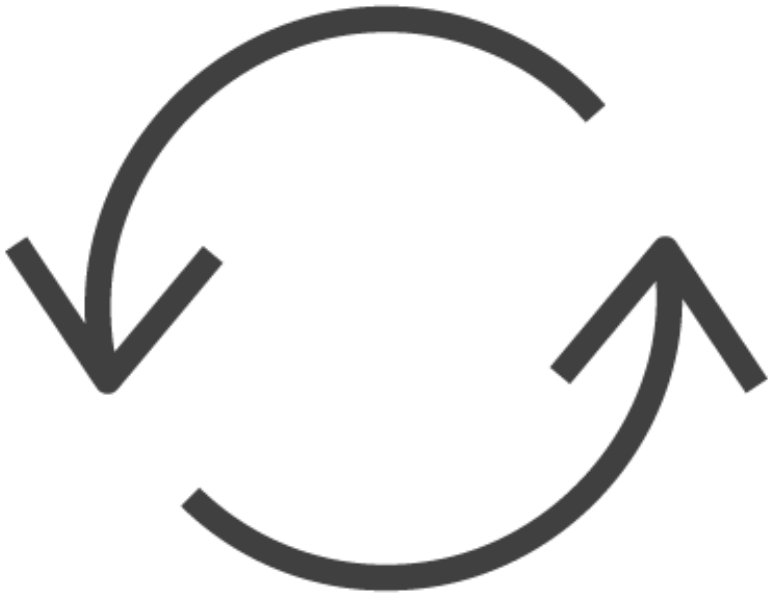
Delegating Handler Pattern



Delegating Handler Pattern



Implementing a Retry Policy



Requests might fail due to a network hick-up or temporary connection issue

A retry policy states that if a request fails it should be tried again (for a set number of times)

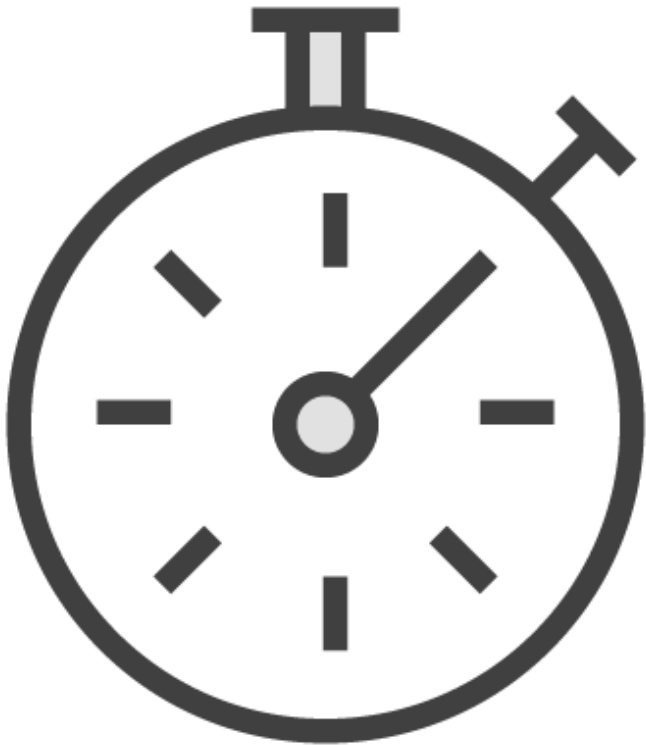
Demo



Implementing a Retry Policy



Implementing a Handler for Dealing with Time-outs



When a request times out, a `TaskCanceledException` is thrown

Through a custom handler we can throw a `TimeoutException` instead, which allows differentiation

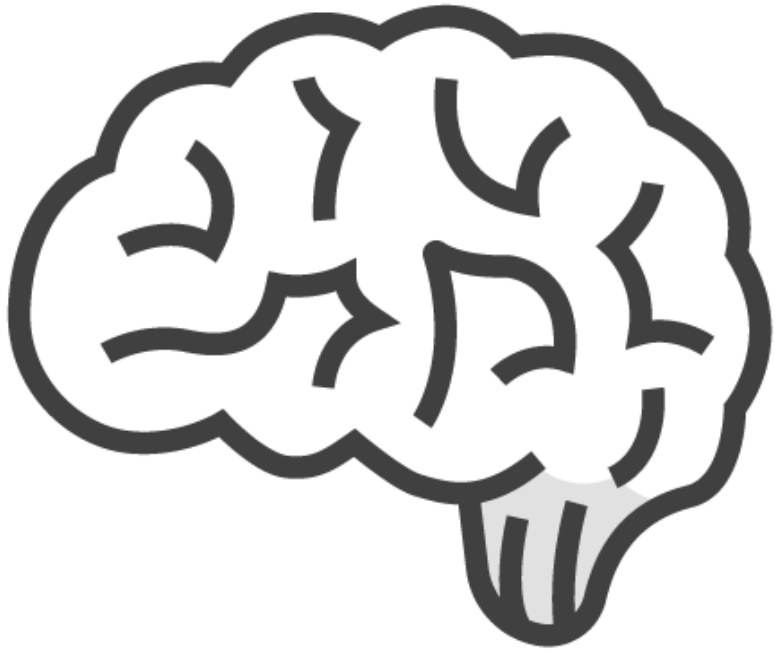
Demo



Implementing a Handler for Dealing with Time-outs



Other Use Cases for Delegating Handlers



Dealing with authentication

- Passing (and refreshing) an access token

Dealing with caching

- Client-side cache implementation

Implementing HSTS support

- <https://bit.ly/2QihSZ5> (Maarten Balliauw)

Using Polly to Improve Interaction with an API



Polly is a .NET resilience and transient-fault-handling library that allows developers to express policies in a fluent and thread-safe manner

- <https://bit.ly/28MXdom>

Fault Tolerant Web Service Requests with Polly (Bryan Hogan)

- <https://bit.ly/2Ap1OuD>

Summary



Delegating Handlers allow us to create a pipeline through which requests and responses flow

- Retry, time-out, authentication, caching, ...

Polly is today's de facto standard