

Using HttpClient to Consume APIs in .NET Core

UNDERSTANDING INTEGRATION WITH AN API USING HTTPCLIENT



Kevin Dockx

ARCHITECT

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



Coming Up



Prerequisites and Tooling

Supported Platforms, Frameworks, and
.NET Standard Versions

Introducing the Demo Application

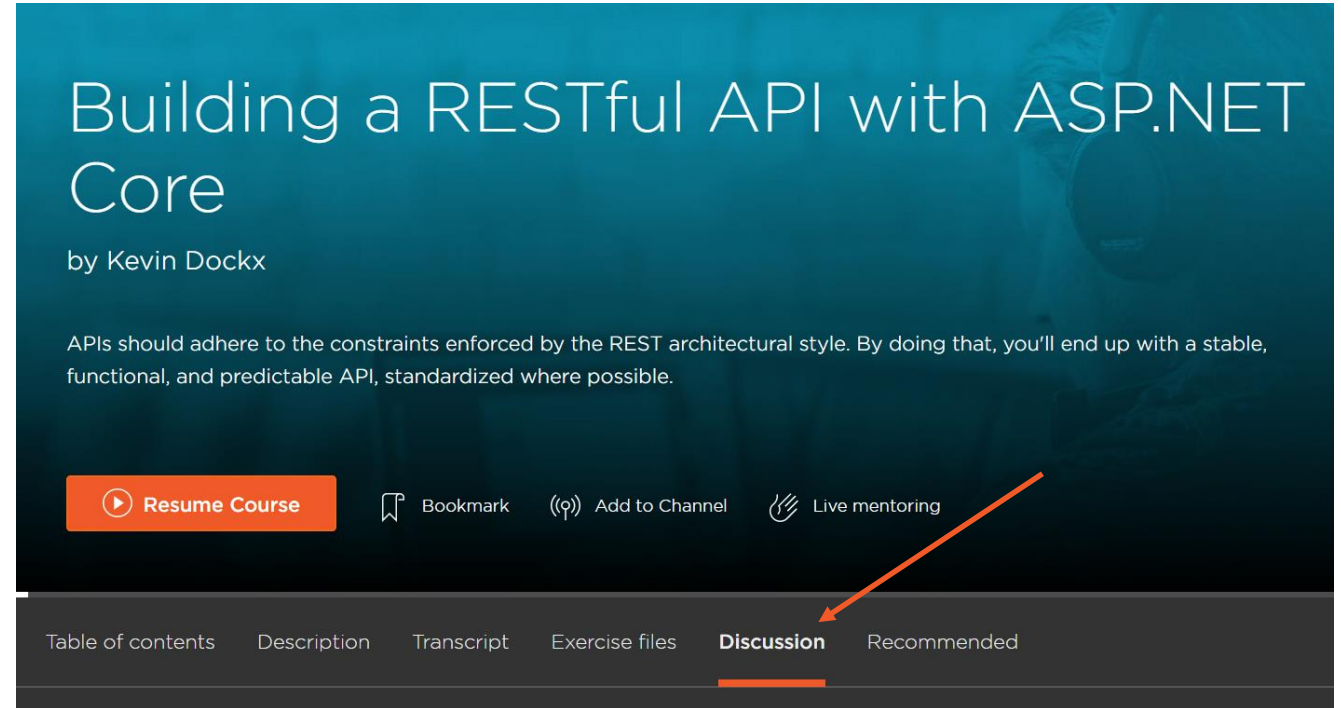
Strategies for Working with DTO
Model Classes

Tackling Integration with HttpClient



Discussion tab on the
course page

Twitter: [@KevinDockx](https://twitter.com/KevinDockx)



(Course shown is one of my other courses, not this one)



Course Prerequisites



Good knowledge of C#



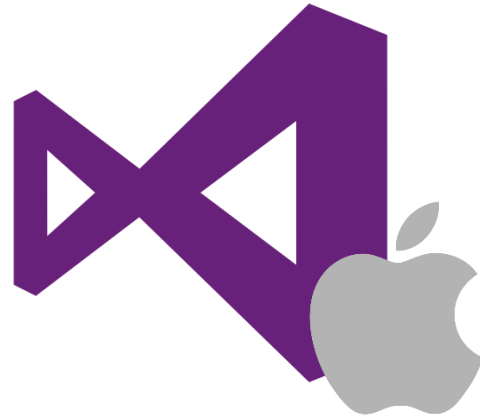
Knowledge of .NET Core

Frameworks and Tooling



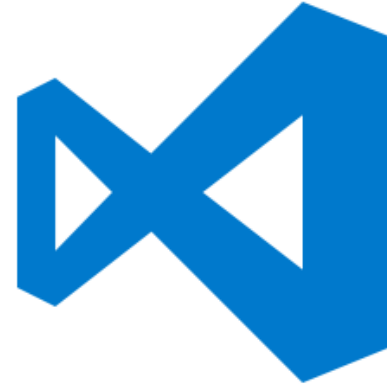
Visual Studio 2017
(15.3 or higher)

<http://bit.ly/2dSGoN5>



Visual Studio for Mac

<http://bit.ly/2fXmQpH>



Visual Studio Code

<http://bit.ly/1J6QrU6>



JetBrains Rider,
Sublime, ...



Supported Platforms, Frameworks, and .NET Standard Versions

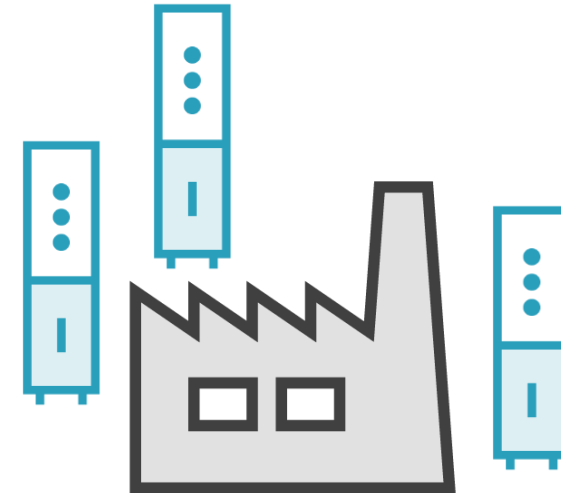


HttpClient

Released with .NET 4.5

Available via NuGet

<https://bit.ly/2DPj9kG>



HttpClientFactory

Released with .NET Core 2.1

Available via NuGet

<https://bit.ly/2OTM4Ff>



Supported Platforms, Frameworks, and .NET Standard Versions



Lowest common denominator: .NET Standard 2.0

- Lower-level packages can be referenced
- .NET Standard is cumulative

Platforms Implementing .NET Standard 2.0



.NET Core 2.0 (+)

- ASP.NET Core, Console, ...

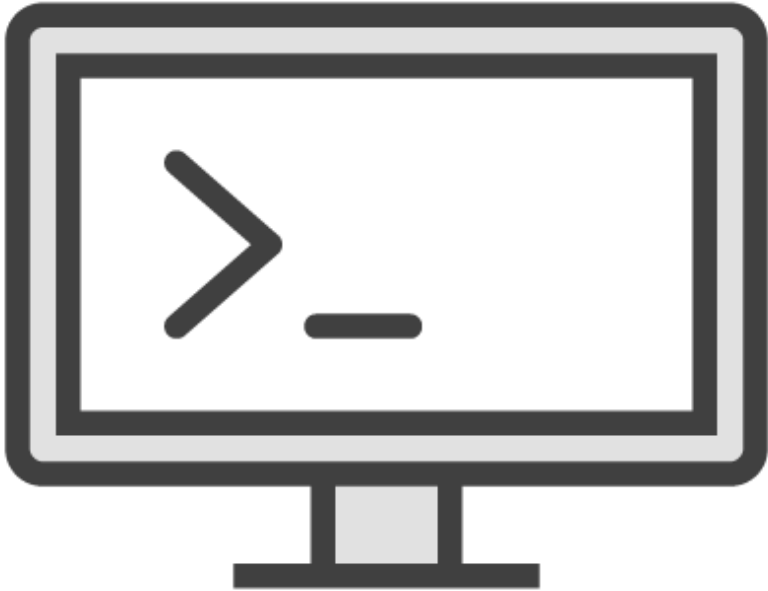
.NET Framework 4.6.1 (+)

- WinForms, WFP, ASP.NET, ...

**Mono 5.4 (+), Xamarin.iOS 10.14 (+),
Xamarin.Mac 3.8 (+), Xamarin.Android 8.0
(+), UWP 10.0.16299 (+)**



Introducing the Demo Application



We'll integrate with the API from a Console application

The Console application targets .NET Core 2.1

- Implements .NET Standard 2.0

We'll use C# 7.1



Demo



Introducing the Demo Application



Strategies for Working with DTO Model Classes

Shared Model Project

Diminishes code duplication, changes only have to be applied in one place

Useful when you want to deploy the model assembly independently

Requires control over API and client

Both must target supported platforms

Linked Files

Diminishes code duplication, changes only have to be applied in one place

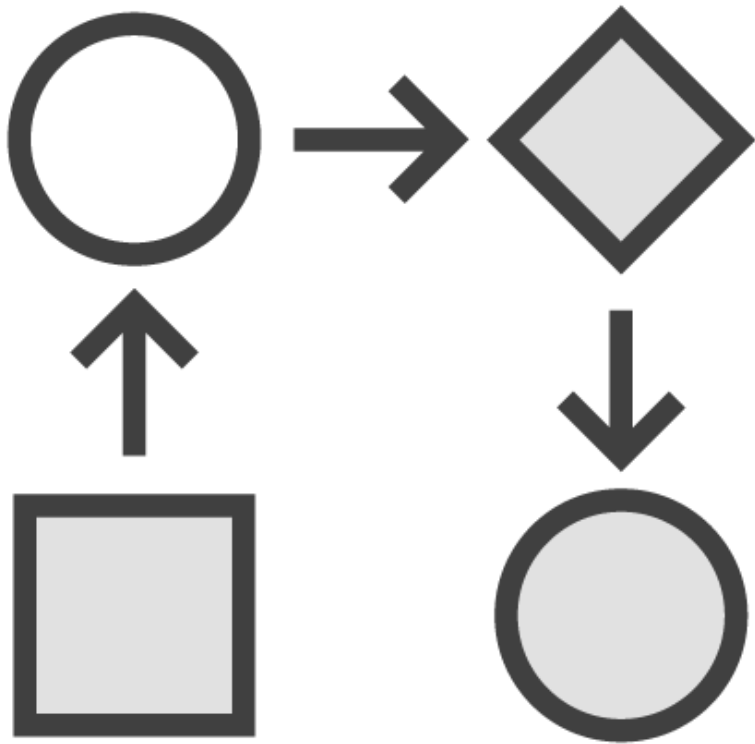
Model classes are packaged in API and client assemblies

Requires control over API and client

Both must target supported platforms



Strategies for Working with DTO Model Classes

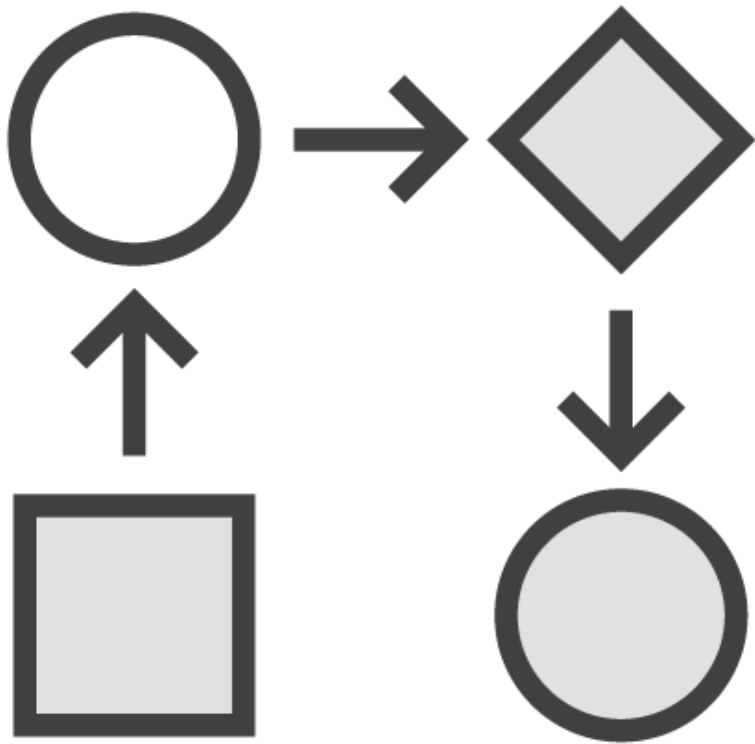


What if you don't have control over the API?

- Might be built in another technology
- Might be built by another team

The technology the API is built with shouldn't matter

Generating DTO Classes



Modern-day generation relies on a machine-readable description (specification) of the API

- OpenAPI (aka Swagger) is such a specification

Generating DTO Classes

**Swagger
CodeGen**

<https://bit.ly/2A4552k>

Online version

<http://editor.swagger.io/>

NSwagStudio

<https://bit.ly/2Tra7Pt>

**“Add REST API
Client”**

(in Visual Studio)

Support for .NET
Core not yet
integrated

Based on AutoRest

<https://bit.ly/2fXjXK1>

... to name a few



Demo



Generating DTO Classes



```
var httpClient = new HttpClient();  
var response = await  
    _httpClient.GetAsync("http://localhost:123/api/movies");  
  
response.EnsureSuccessStatusCode();  
  
var content = await response.Content.ReadAsStringAsync();  
var movies = JsonConvert.DeserializeObject<List<Movie>>(content);
```

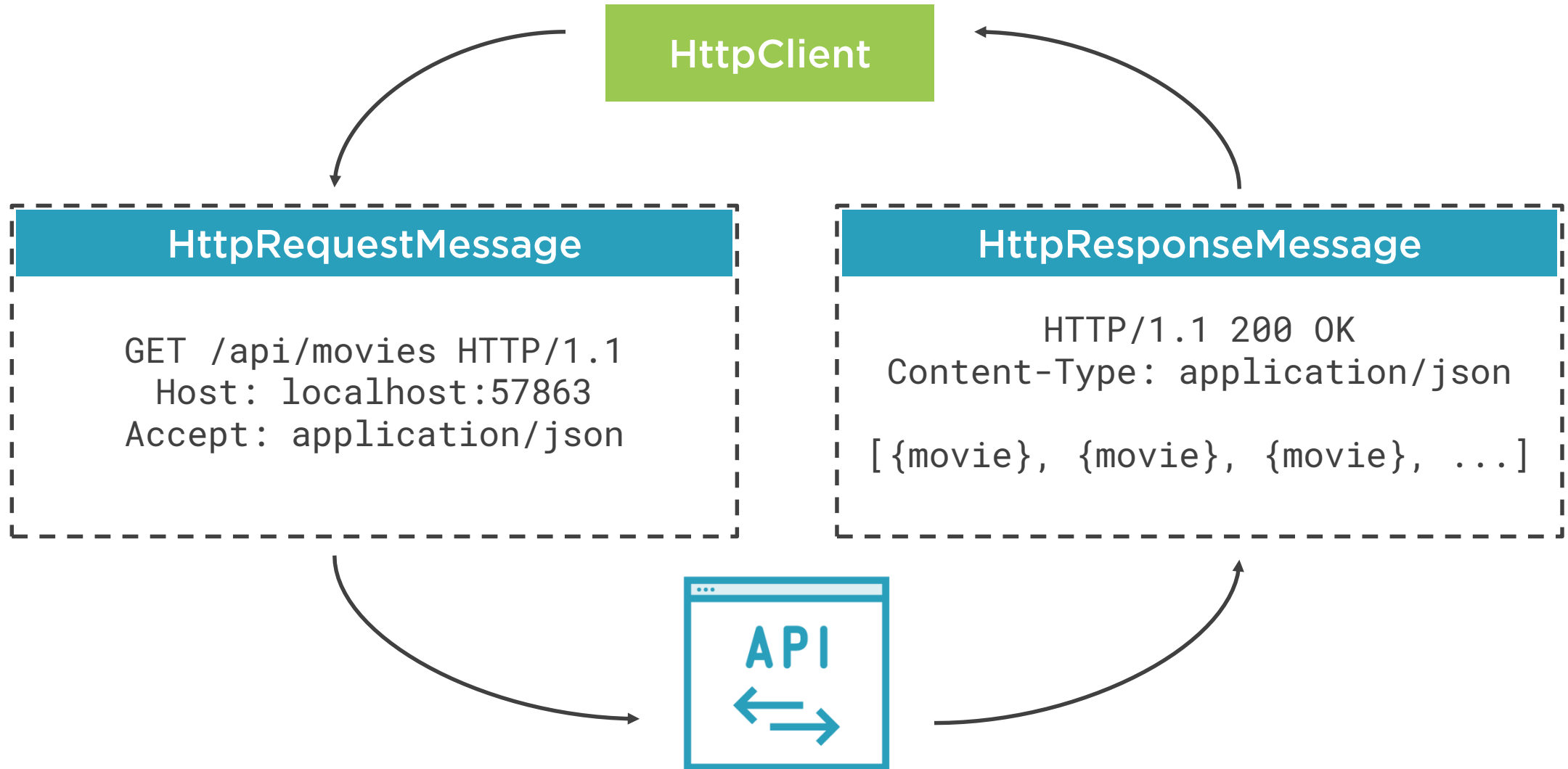
Tackling Integration with HttpClient

HTTP is a request-response protocol between a client and server

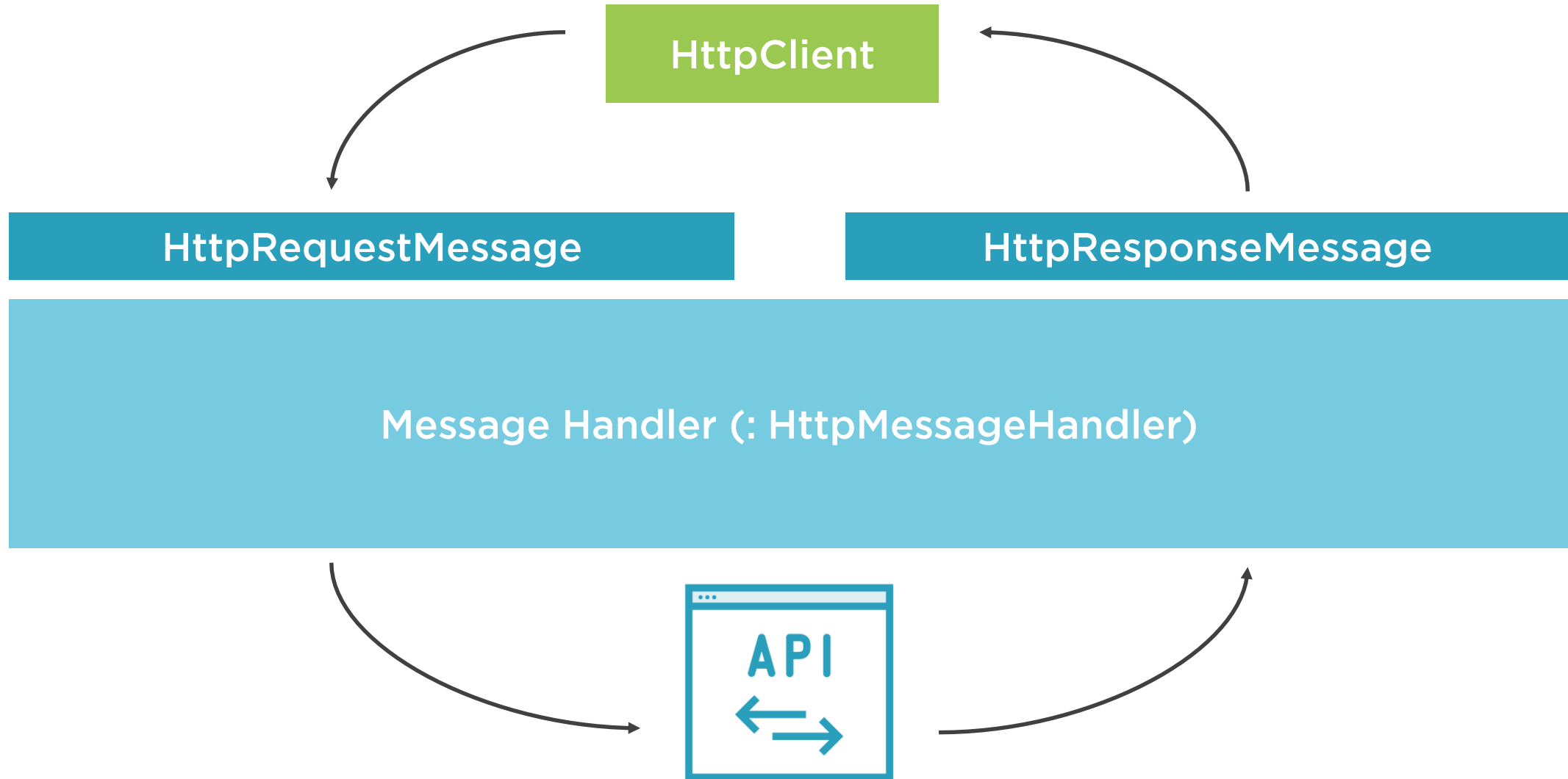
**A browser is an HTTP client that can send messages
and capture responses**



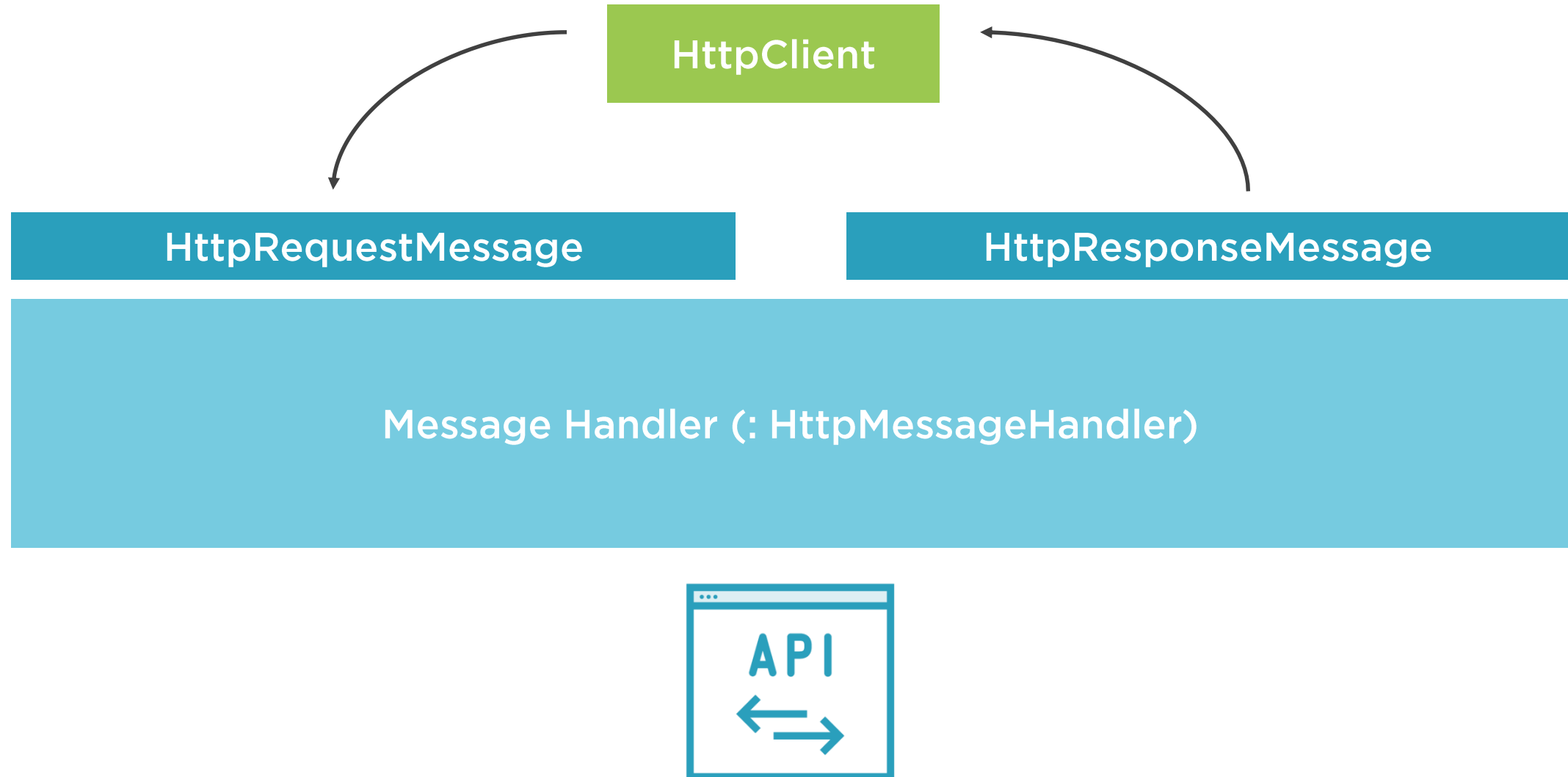
Tackling Integration with HttpClient



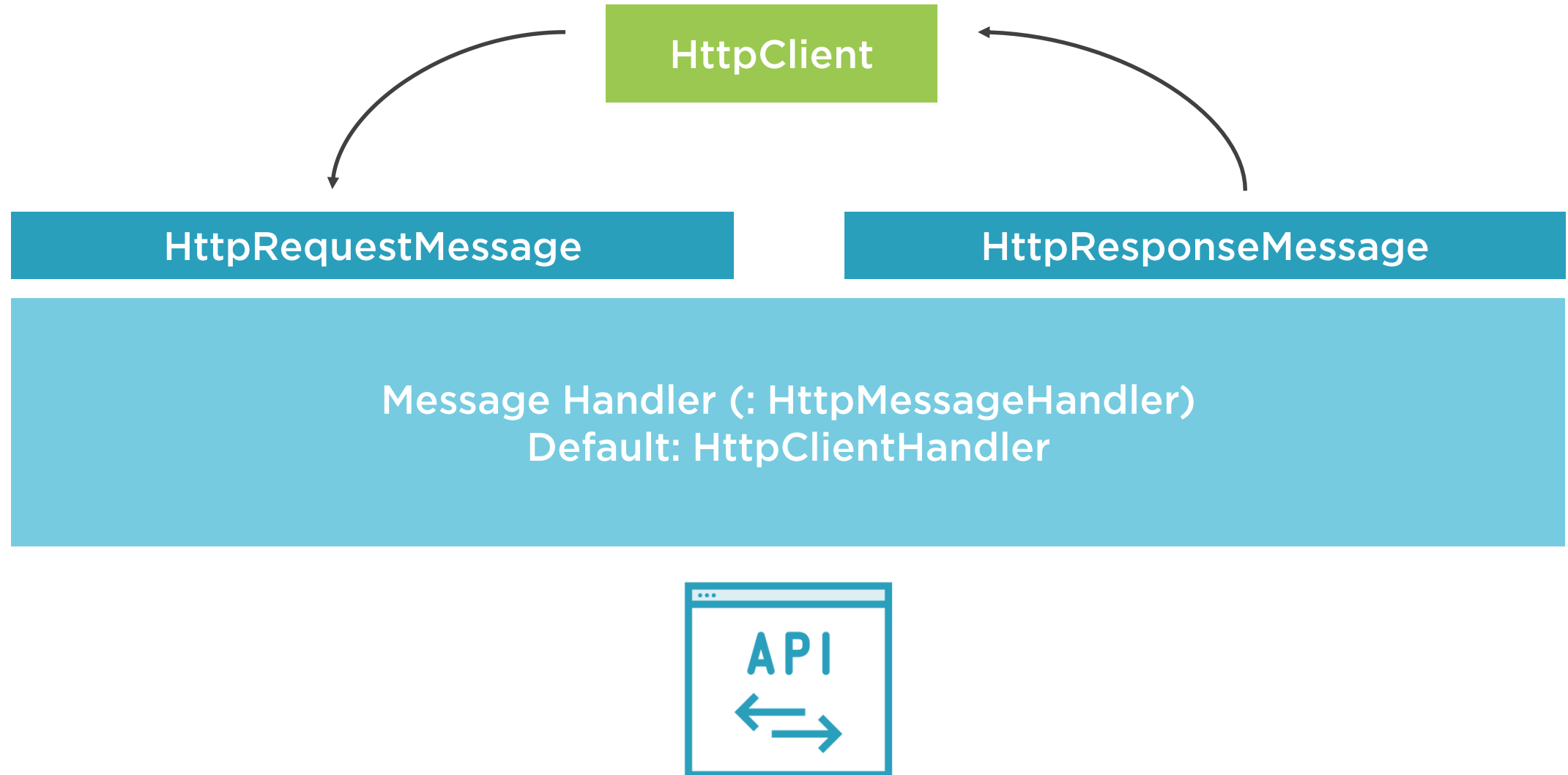
Tackling Integration with HttpClient



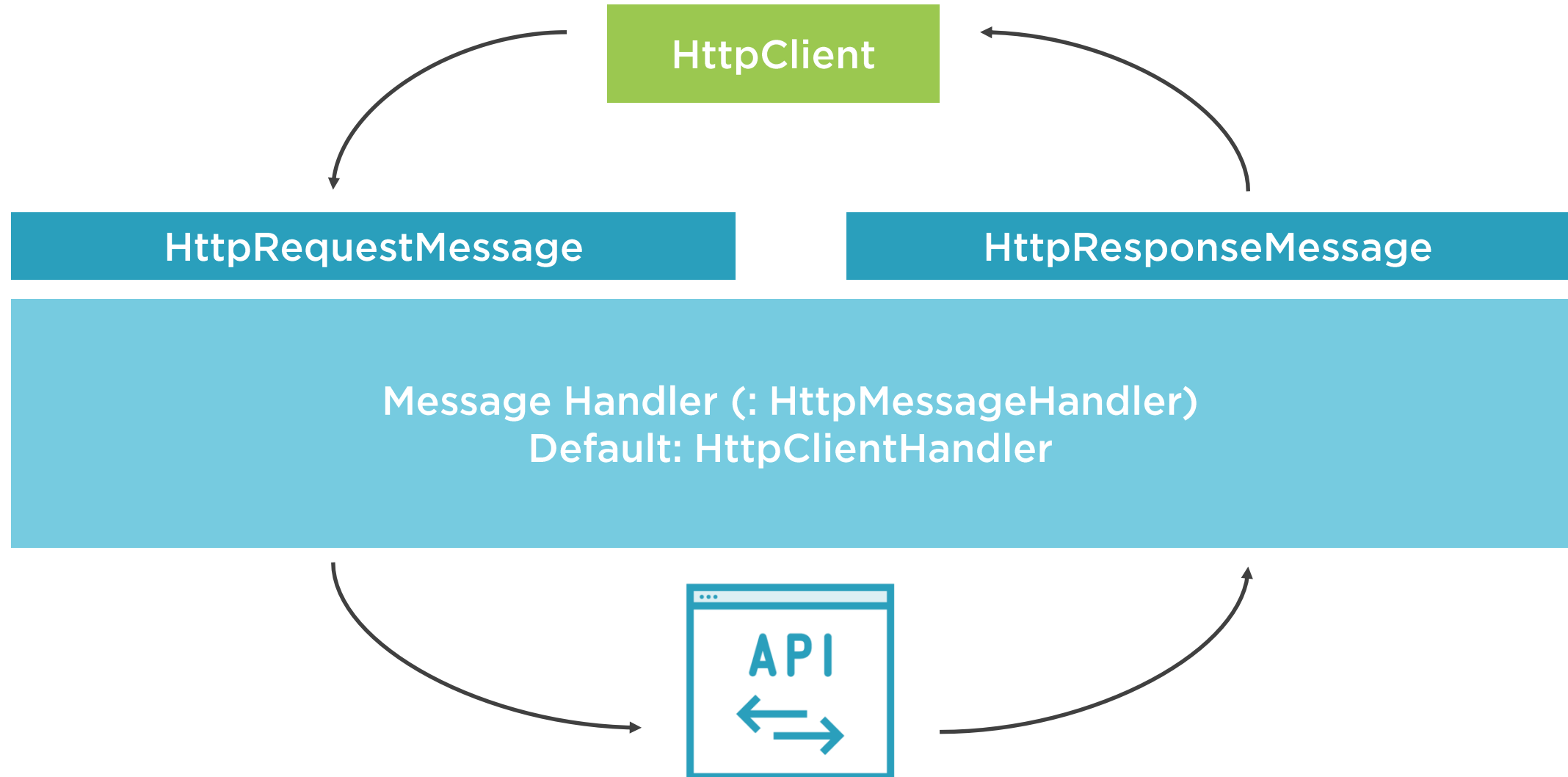
Tackling Integration with HttpClient



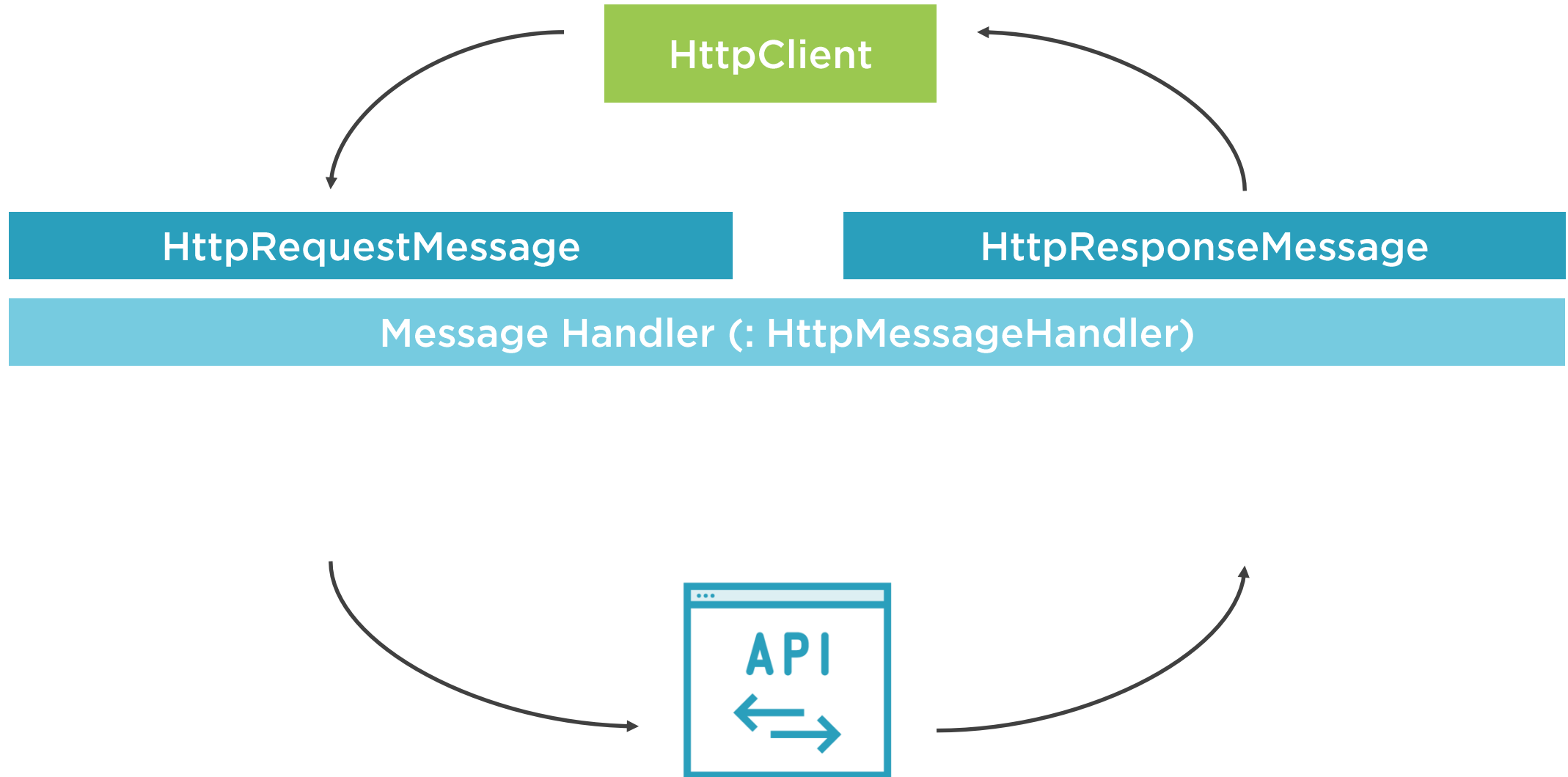
Tackling Integration with HttpClient



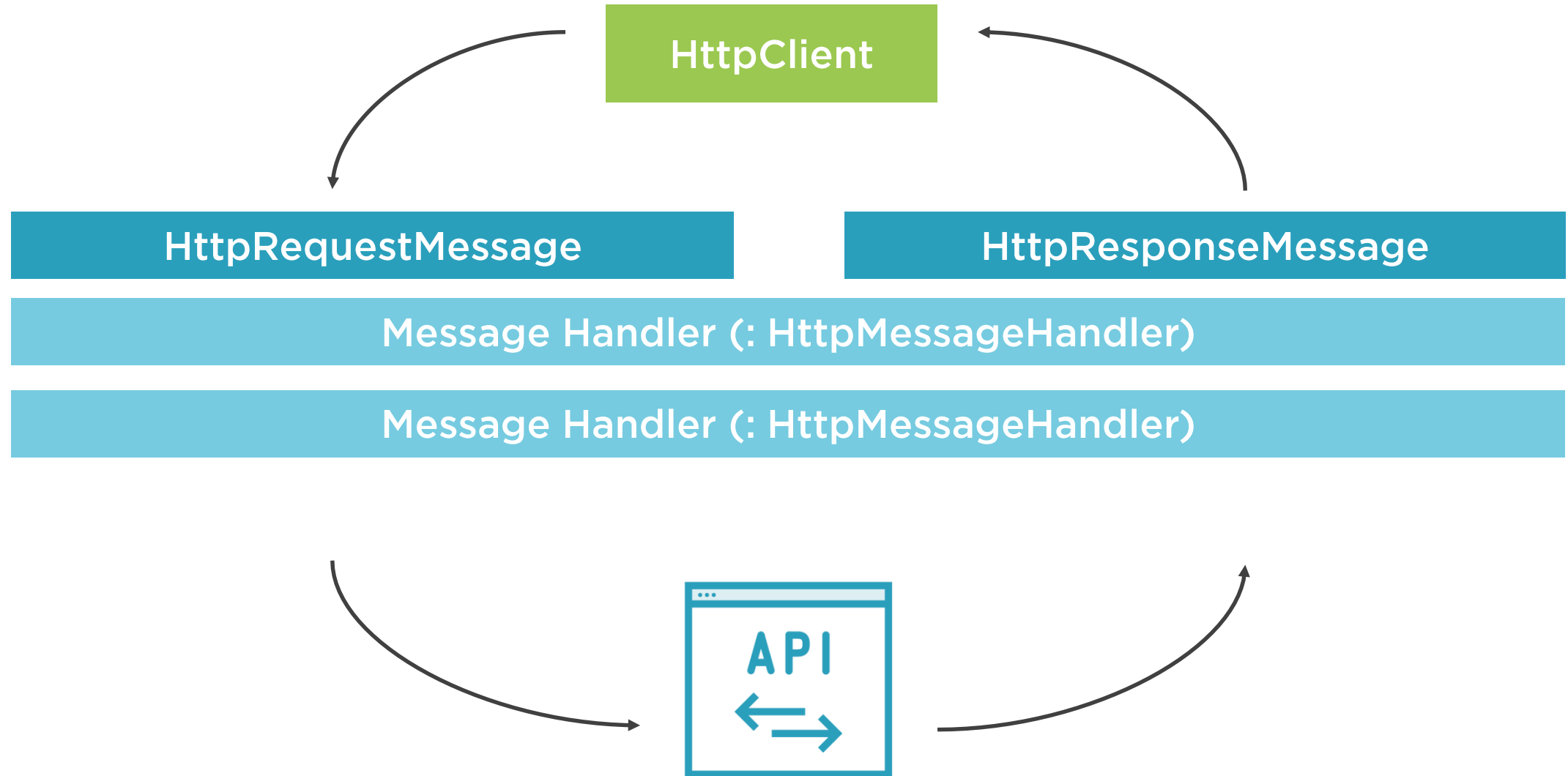
Tackling Integration with HttpClient



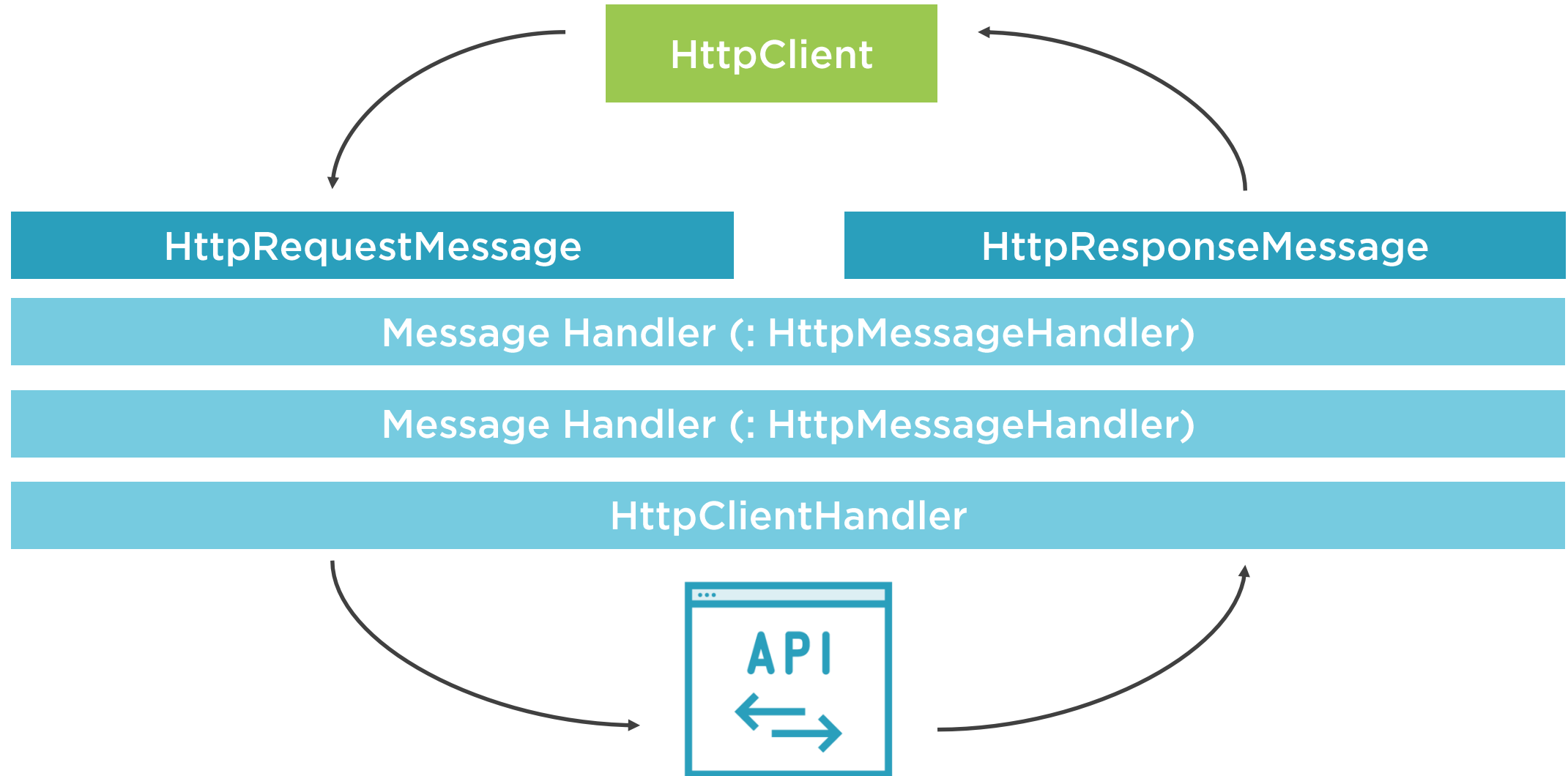
Tackling Integration with HttpClient



Tackling Integration with HttpClient



Tackling Integration with HttpClient



Summary



The combination of `HttpClient` & `HttpClientFactory` can be used on any platform that implements .NET Standard 2.0 (or higher)

Sharing DTO classes

- Linked files
- Shared assembly

Generating DTO classes

- Start from an OpenAPI specification

Summary



Each `HttpRequestMessage` travels through a set of handlers, and the `HttpResponseMessage` travels back up through the same set

- Handlers can pass on requests or cancel them
- `HttpClientHandler` is responsible for sending the actual request