

Supporting Cancellation



Kevin Dockx

ARCHITECT

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



Coming Up



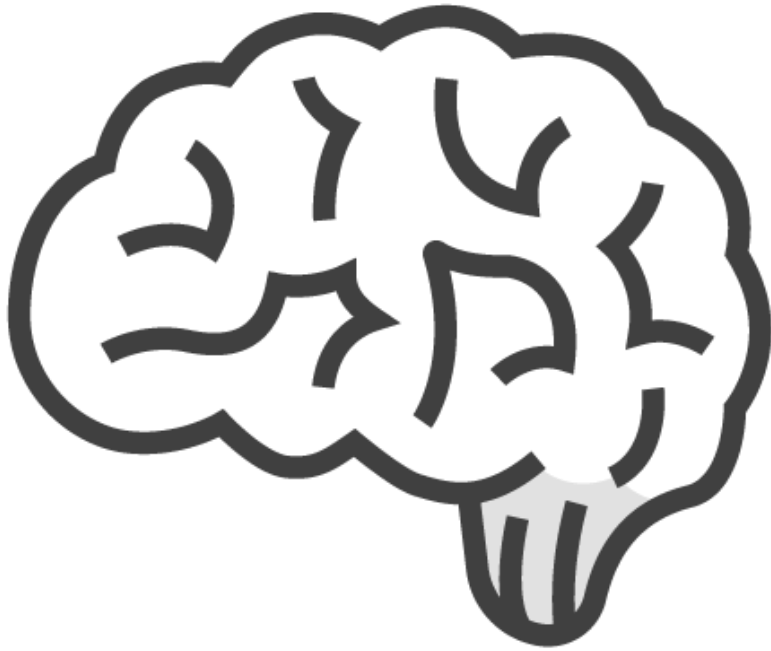
Why Cancelling Requests Is Important

Cancelling a Request

Handling a Cancelled Request



Why Cancelling Requests Is Important



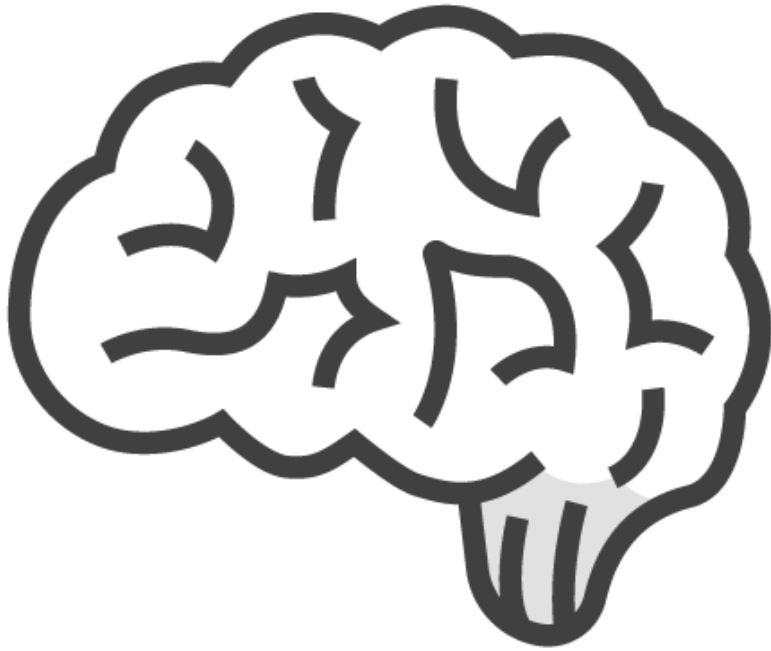
HttpClient works with async Tasks

- Cancelling such a Task potentially frees up a thread
- The Task is cancelled by us

Timeouts must be handled gracefully

- The Task is cancelled due to the timeout

CancellationTokenSource and CancellationToken



CancellationTokenSource

- Manages and sends cancellation notifications

CancellationToken

- Exposed through Token property of CancellationTokenSource

Asking for cancellation notifies the receiver of the token that it should cancel its Task

Demo



Cancelling a Request



Demo



CancellationTokenSource Initialization



Demo



Handling a Cancelled Request



Demo



Dealing with Timeouts



Summary



Manually cancelling a Task potentially frees up a thread

On timeout, HttpClient automatically cancels the Task



Summary



HttpClient is responsible for cancelling its Task(s)

- We can notify it of this request for cancellation by calling `Cancel` on the `CancellationTokenSource` providing the `CancellationToken`

To handle cancellation, catch the `TaskCanceledException` (or `OperationCanceledException`)