Circuit breaker pattern

(Demo prep)

Implementing the circuit breaker pattern

1. To implement the circuit breaker pattern all we have to do is add a new transient http error policy just like we did for the wait and retry policy.

2. Add the circuit breaker policy:

```
.AddTransientHttpErrorPolicy(builder => builder.Or<TimeoutRejectedException>().CircuitBreakerAsync(
    3,
    TimeSpan.FromSeconds(15),
    onBreak: (outcome, timespan) =>
    {
        var serviceProvider = services.BuildServiceProvider();
        serviceProvider.GetService<ILogger<CatalogClient>>()?
            .LogWarning($"Opening the circuit for {timespan.TotalSeconds} seconds...");
    },
    onReset: () =>
    {
        var serviceProvider = services.BuildServiceProvider();
        serviceProvider.GetService<ILogger<CatalogClient>>()?
            .LogWarning("Closing the circuit...");
    }
))
```

3. **dotnet run** in both terminals

4. Show the circuit breaking and what happens when querying while circuit open.

5. Try again

At this point we are doing a fairly good job at handling partial failures between our Inventory and Catalog services. However, could there be a better way to have Inventory get the information it needs from Catalog without having to rely on Catalog to be available most of the time?

In the next module we will learn about the Asynchronous Communication style and how it can help us enable much more resilient communication between our services.