Working with Asynchronous Streams

Mark Heath
Software Architect

@mark_heath www.markheath.net

Overview

Async methods can't easily be used in LINQ pipelines

IAsyncEnumerable<T>

- Generating and consuming async sequences
- Creating pipelines
- Converting IEnumerable<T> to IAsyncEnumerable<T>

Parallel.ForEachAsync

Demo

The problem with async methods in LINQ pipelines

Working with “Asynchronous Streams”

IAsyncEnumerable<T>

Working with “Asynchronous Streams”

IAsyncEnumerable<T>

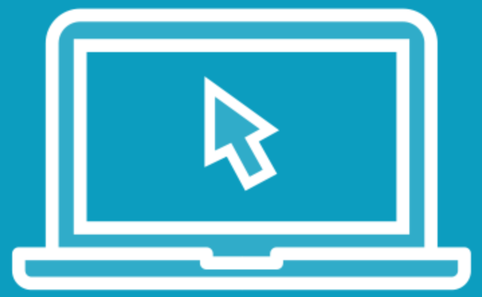
IEnumerable<T>

```
interface IEnumerable<T>
{
    IEnumerator<T>
        GetEnumerator();
}

interface IEnumerator<T>
{
    bool MoveNext();
    T Current { get; }
}
```

IAsyncEnumerable<T>

```
interface IAsyncEnumerable<T>
{
    IAsyncEnumerator<T>
        GetAsyncEnumerator();
}

interface IAsyncEnumerator<T>
{
    ValueTask<bool> MoveNextAsync();
    T Current { get; }
}
```

Demo



Generating “asynchronous streams” with IAsyncEnumerable<T>

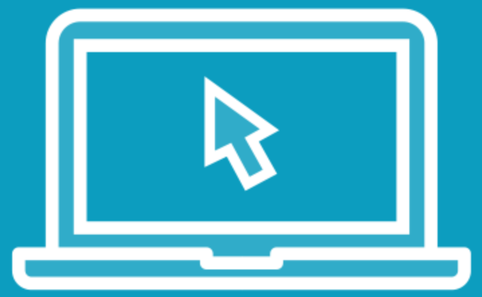
Demo

Consuming an IAsyncEnumerable<T> with foreach

Demo

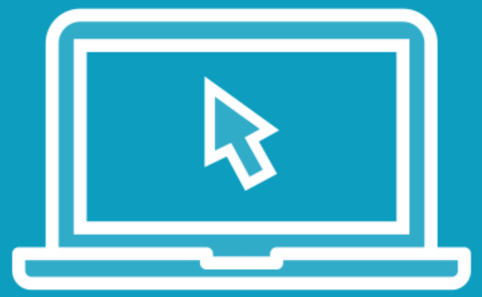
Cancelling enumeration of an IAsyncEnumerable<T>

Demo



Using IAsyncEnumerable<T> in pipelines

Demo



Calling asynchronous methods in pipelines

Mixing Async with LINQ Pipelines

Testability

Async methods need to mocked for unit testing

Consider avoiding side-effects in LINQ pipelines

Parallelization

IAsyncEnumerable<T> extension methods are not always the best choice

Demo

IAsyncEnumerable<T> in action

Demo

Using Parallel.ForEachAsync

Summary

IAsyncEnumerable<T>

- Generate “Asynchronous streams”
- await foreach
- System.Linq.Async NuGet package

Consider…

- Unit testing
- Parallelization
- Parallel.ForEachAsync

Up Next: Functional Programming