

Supporting Asynchronous Streaming



Kevin Dockx

Architect

@Kevindockx | www.kevindockx.com

Coming Up

A few words on streaming

Synchronous vs. asynchronous iteration

Supporting streaming with
`IAsyncEnumerable<T>`



Streaming

A method of transmitting or receiving data over a network as a steady, continuous flow instead of all at once



A Few Words on Streaming

Often used for video/audio, but also useful
when working with large sets of data



Synchronous Vs. Asynchronous Iteration

Iteration over collections is enabled by
`System.Collections.IEnumerable`
– `System.Collections.IEnumerator`



```
IEnumerator<int> enumerator = collection.GetEnumerator();  
  
while (enumerator.MoveNext())  
{  
    var item = enumerator.Current;  
    Console.WriteLine(item.ToString());  
}
```

Synchronous Vs. Asynchronous Iteration



```
public static IEnumerable<T> ReturnEvenElements<T>(this IEnumerable<T> source)
{
    int index = 0;
    foreach (T item in source)
    {
        if (index % 2 == 0)
        {
            yield return item;
        }
        index++;
    }
}
```

Synchronous Vs. Asynchronous Iteration

Iterator methods are methods that create a source for iteration



```
foreach (T item in collection)
{
    await SomeOperation();
}
```

Synchronous Vs. Asynchronous Iteration

The `await` operator can be used inside of a `foreach` statement



Synchronous Vs. Asynchronous Iteration

iteration

operation

iteration

operation



Synchronous Vs. Asynchronous Iteration

iteration

operation

iteration

operation



Synchronous Vs. Asynchronous Iteration

iteration

operation

iteration

operation



Supporting Streaming with **IAsyncEnumerable** **<T>**

IAsyncEnumerable<T>

- Provides the ability to iterate over a set of values asynchronously

Since ASP.NET Core 6 System.Text.Json supports streaming, which, combined with IAsyncEnumerable<T>, can result in objects being streamed to the consumer



Demo

**Supporting streaming with
IAsyncEnumerable<T>**



Summary

The async iterator approach leads to threads getting freed up earlier and thus to better scalable streaming

Enabling components

- `IAsyncEnumerable<T>`
- `System.Text.Json`



Dealing with Asynchronous Service Integrations and Supporting Cancellation

