

Real-World Reactive Extensions (Rx)

Scott Reed

Brain Hz Consulting, Inc.

scott@brainhzconsulting.com

Llewellyn Falco

DevelopMentor

info@spunlabs.com

Rx Koans

- <http://rxkoans.codeplex.com>
- Developed by Llewellyn and Bart

Pull Iteration with IEnumerable

```
foreach (int num in GetRandomNumbers(15))
{
    Console.WriteLine(num);
}
```

```
public interface IEnumerable<out T> : IEnumerable
{
    new IEnumerator<T> GetEnumerator();
}
```

```
public interface IEnumerator<out T> : IDisposable, IEnumerator
{
    new T Current { get; }
    bool MoveNext();
    void Reset();
}
```

Pull Iteration with IEnumerable (2)

```
var enumerator = GetRandomNumbers(15).GetEnumerator();  
try  
{  
    while (enumerator.MoveNext())  
    {  
        Data → Console.WriteLine(enumerator.Current);  
    }  
} catch (Exception e)  
{  
    Error → Console.WriteLine(e.ToString());  
} finally  
{  
    Complete → enumerator.Dispose();  
}
```

Push Iteration with IObservable

```
public interface IObservable<out T>
{
    IDisposable subscribe(IObserver<T> observer);
}
```

```
public interface IObserver<in T>
{
    void onNext(T value);
    void onError(Exception error);
    void onCompleted();
}
```

Demo

- **Using AsObservable**
- **Lesson 1**
 - TheLastEvent
 - EverythingCounts

Threading

- Demo of why threading matters

Subjects

- Implements `IObservable<T>`, `IObserver<T>`
- Has public methods which allow firing of the events

Demo

- **Lesson 1**
 - ThisIsStillAnEventStream

Do

- **Like subscribe, but as the event chain is being formulated**

Demo

- **Lesson2**
 - ComposableBeforeAndAfter

Dispose

- **IObserver implements IDisposable**
- **Dispose can be called at any time to “Unsubscribe”**

Demo

- **Lesson1**
 - EventsWhileSubscribing
 - TakeUntilFull

Standard LINQ operators

- **Where**
- **Select**
- **GroupBy**
- **Sum, Avg, Min, Max, Any, All**

Demo

- **Lesson 2**
 - CreatingAMoreRelevantEventStream
 - WeWroteThis
 - ComposableAddition

Query Expression Syntax

- Same as LINQ, improves Readability

Demo

- **Lesson 3**
 - LinqOverMouseEvent

Time

- **Delay**
- **Timeout**
- **Throttle**
- **Buffer**

Demo

- **Lesson 4**
 - YouCanPlaceATimeLimitOnHowLongAnEventCanTake
 - Buffering

Converting Events

```
Observable.FromEventPattern<MouseEventArgs>(
    this, "MouseMove")
```

```
Observable.FromEvent<MouseEventHandler, MouseEventArgs>(
    h => MouseMove += h, h => MouseMove -= h)
```

Demo

- **Lesson 5**
 - TheMainEvent

Threading

- **Observable thread controlled using SubscribeOn**
- **Observer thread controlled using ObserveOn**

Converting Async Pattern

- **FromAsyncPattern**
- **ToAsync**

Demo

- **Lesson 7**
 - NiceAndEasyAsyncPattern

Real World Demos

- **A Silverlight Drag and Drop**
- **Throttling text changed events to call Web Services**

Thank You!



Scott Reed

Brain Hz Consulting, Inc.

scott@brainhzconsulting.com

www.brainhzconsulting.com