Reactive Extension

RX - Version: 1

2 days Course

Description:

2 days that target the Reactive Extension (RX) library.
Rx is a functional programming library designed to handle complex event processing. The course deep-dives into the library's concept and guidelines. It will cover topics like exception handling, testing, remote processing, and scheduling. Students will master both practice and theory and become familiar with numerous RX operators.

Intended audience:

.NET developers or team leaders with:</br>
•at least 2 years of experience with C#;</br>
•Programming experience and some practice with LINQ Query; </br>
•Programming experience and some practice with multithreading; </br>
•Familiarity with TPL (.NET 4) is recommended.

Prerequisites:

Objectives:

Appreciate the architectural and design principles of the RX programming model
Practice complex event processing
Learn to handle exceptions
Test complex event pipeline
Know how to design RX flow
Master advanced topics like remote execution and scheduling

Topics:

Introduction

- What is Rx?
- Push standard
- LINQ-able
- Like Events but better
- Course Goals
- Why Rx

    - Push vs. pull

    - Test Case: Cloud search

Get started

- NuGet

Concept

- Producer / Consumer

Library structure

- Different offering
- Enlighten concept

Marble Diagram

- Concept
- Select
- Where

Built-in factories

- Interval
- Timer
- Range
- Return
- Create
- Generate

Monitoring

- Do operator
- Visual RX
- LAB

Concurrency model

- Scheduler
- Built-in Scheduler
- ObserveOn
- SubscribeOn

Backwards compatibility

- Composite events

Exception Handling

- Retry
- Catch
- Finally
- SubscribeSafe
- LAB

Producer nature

- Hot Vs. Cold Observables
- Publish
- RefCount

Subjects

- Subject
- Replay and ReplaySubject

Operators

- District
- DistinctUntilChanged
- Sample
- Aggregate
- Scan

Common Combinators

- Merge
- Zip
- CombineLatest
- Amb

Splitting

- Buffer
- Window
- Select Many
- Group By
- LAB

Time-oriented

- Interval
- Timeout
- Timestamp
- TimeInterval
- Throttle
- Sample
- Take and Skip
- Generate Delay
- Delay Subscription

Time-based Combinators

- Join
- Group Join

Disposables

- Create
- CompositeDisposable
- RefCountDisposable
- CancellationDisposable
- BooleanDisposable
- ContextDisposable
- ScheduledDisposable
- SingleAssignmentDisposable
- MultipleAssignmentDisposable
- SerialDisposable
- LAB

Scheduling

- Custom Scheduler
- Virtual time
- Historical Scheduler

Testing

- Notification
- TestScheduler
- CreateHotObservable / CreateColdObservable

Custom Operations

- Defer
- Extending the framework
- LAB

Remote

- Rx-Remoting
- IQbservable

Guidelines

- Design, design, design
- Serialized Fashion
- Resources
- Unsubscribe
- Subscription
- Hot vs. Cold Observable
- LINQ query syntax
- Limit buffering
- Scheduler
- ObserveOn
- Design for Testability
- Custom operators

º Summary