Why should you care about writing Reactive Mobile Apps?

Mark Allibone

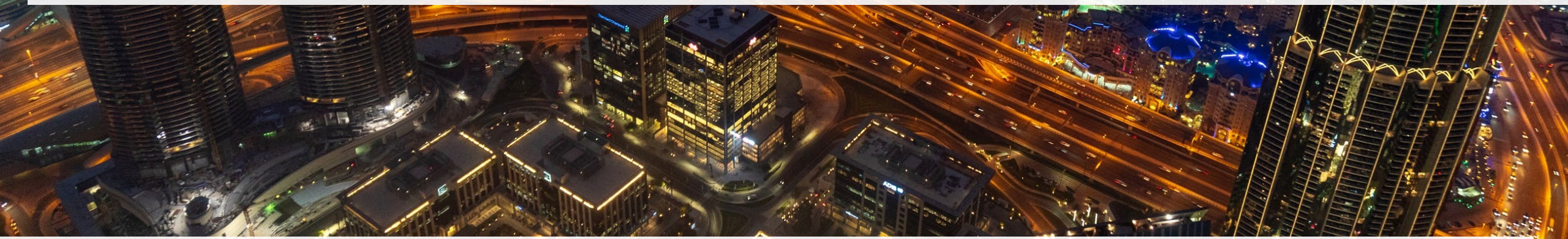
Rey Technology

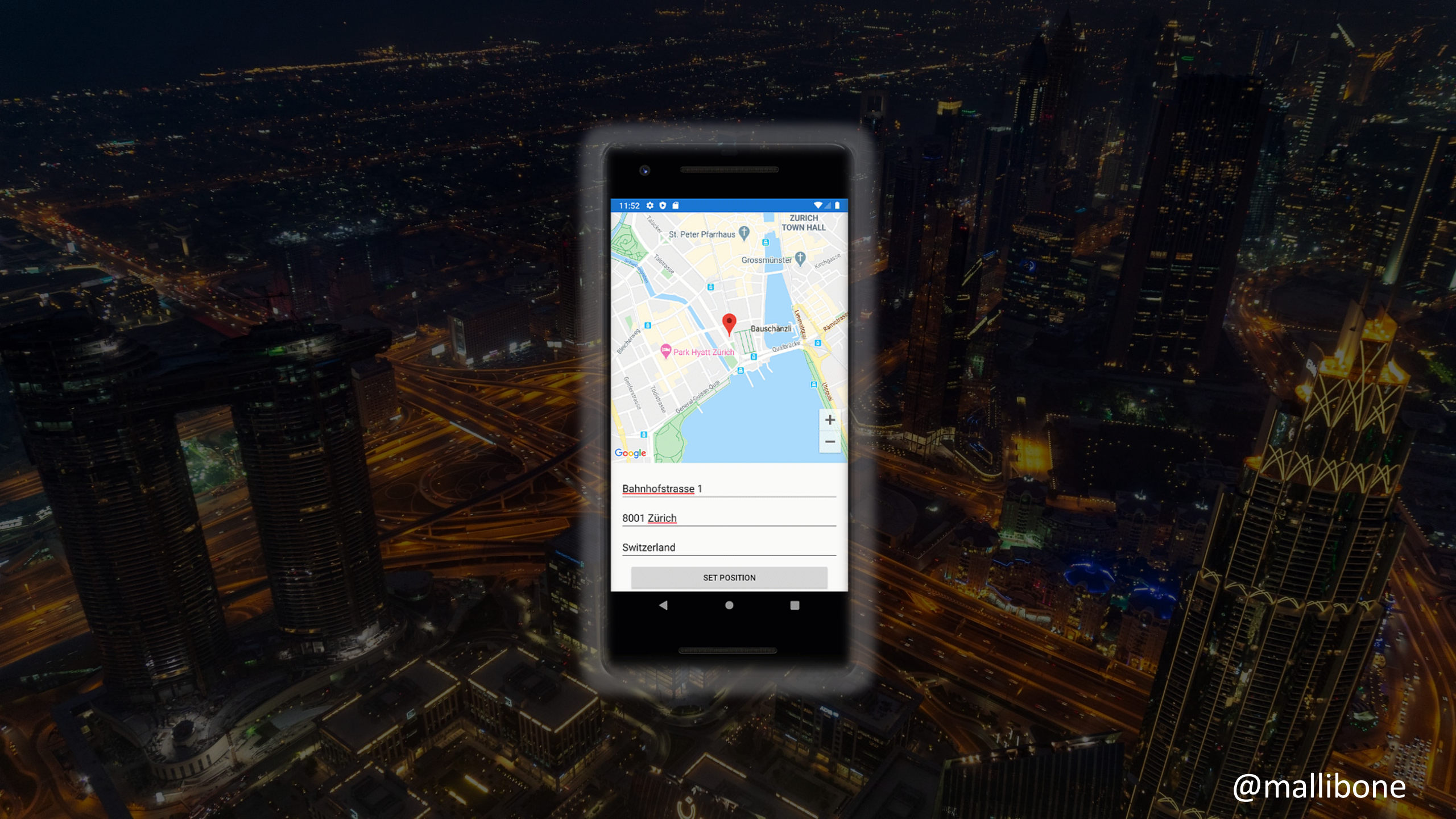
@mallibone


13:43
A-Z
Calculator
Calendar
Call
Camera
Chrome
Clock
Contacts
Daily Mail Online
Downloads
Drive
Dropbox
Duo
eBay
enjoy.now
Facebook
Files
Gmail
Google
Instagram
InstaSize

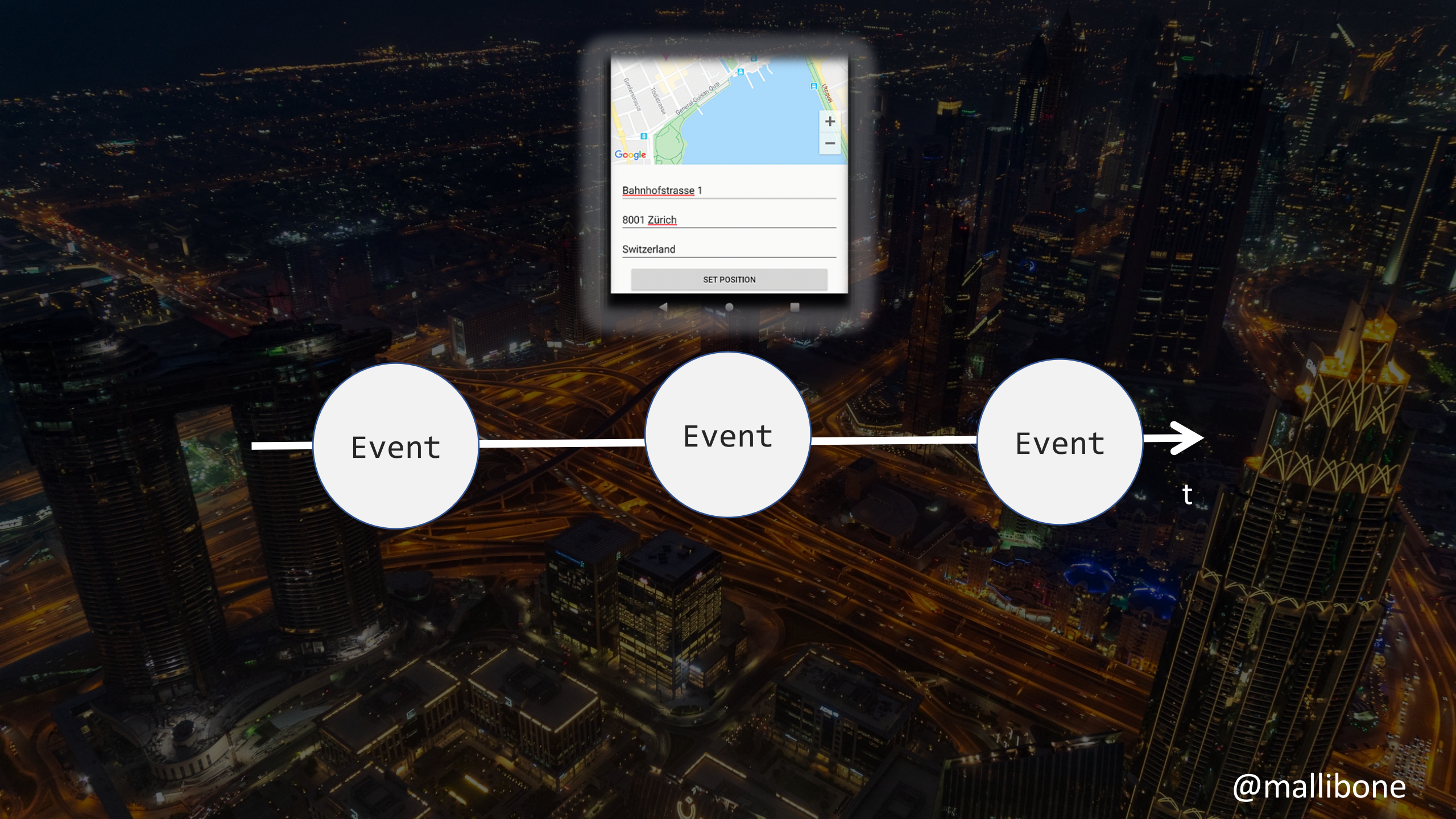
Most actions on a mobile app have a *reactive* nature.

What do you mean with *reactive*?




11:52
ZURICH
TOWN HALL
St. Peter Pfarrhaus
Grossmünster
Bauschänzli
Park Hyatt Zürich
Google
Bahnhofstrasse 1
8001 Zürich
Switzerland
SET POSITION

@mallibone


Bahnhofstrasse 1
8001 Zürich
Switzerland
SET POSITION
Event
Event
Event
t
@mallibone

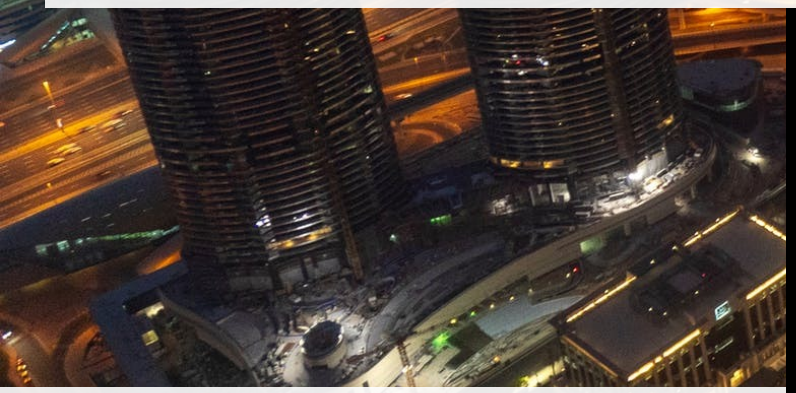
Reactive Extensions (Rx) 🥰 Events

@mallibone

Rx = Observables + LINQ + Schedulers

Events*

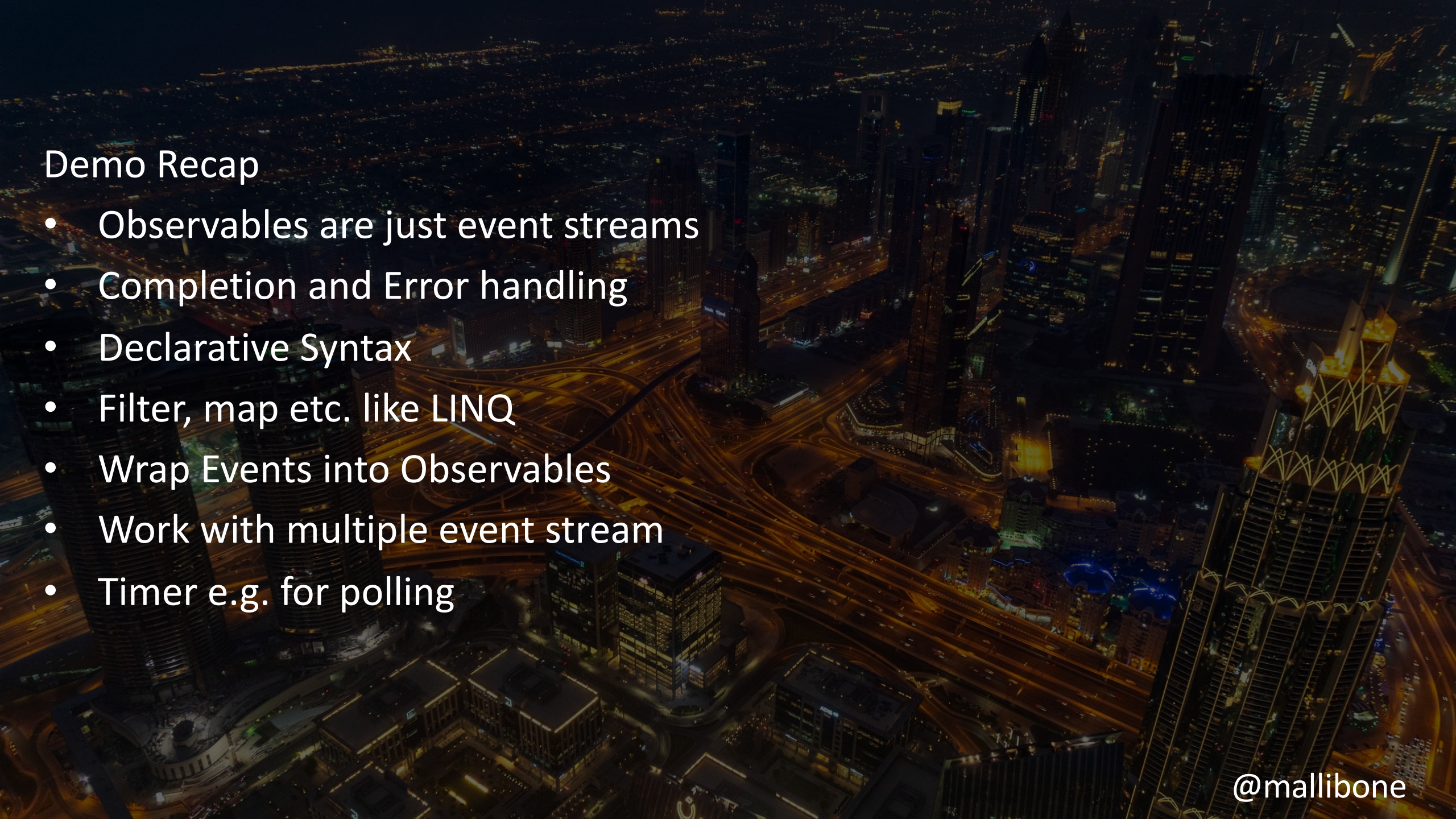
Threads/Tasks*



@mallibone

Rx101 · 2 projects
Rx101
Dependencies
Helpers
EventSample.cs
MeasurementUpdate.cs
ObservableSample.cs
Demo01.cs
Demo02.cs
Demo03.cs
Demo04.cs
Program.cs
Rx101.Test
Dependencies
UnitTest1.cs
Scratches and Consoles

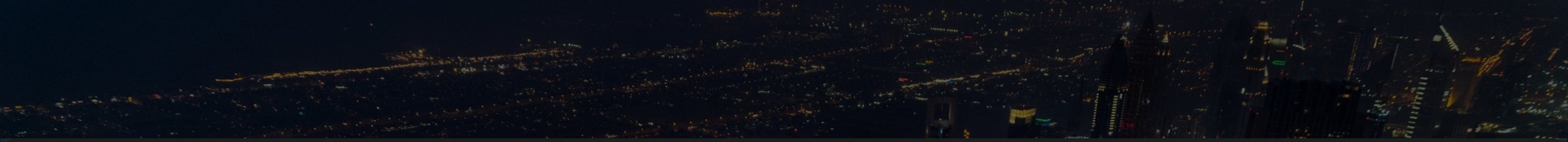
```
namespace Rx101
{
    1 usage  Mark Allibone *
    public static class Demo01
    {
        1 usage  Mark Allibone *
        public static void SimpleComparison()
        {
            Console.WriteLine("Simple Event comparison");
            RunEventSample();
            RunObservableSample();
        }

        1 usage  new *
        private static void RunObservableSample()
        {
            var observableSample = new ObservableSample();
            var measurmentChangedSubscription :IDisposable =
                observableSample.MeasurementChanged.Subscribe( onNext: update =>
                    Console.WriteLine($"Temperature update {update.CurrentMeasurement}"));
            observableSample.NewMeasurementReading( temperature: 24.0f);
            measurmentChangedSubscription.Dispose();

        }

        1 usage  new *
        private static void RunEventSample()
        {
            void EventSampleOnMeasurementChanged(object sender, MeasurementUpdate update) =>
                Console.WriteLine($"Temperature update {update.CurrentMeasurement}");

            var eventSample = new EventSample();
            eventSample.MeasurementChanged += EventSampleOnMeasurementChanged;
            eventSample.NewMeasruementReading( measurement: 22.0f);
            eventSample.MeasurementChanged -= EventSampleOnMeasurementChanged;
        }
    }
}
```



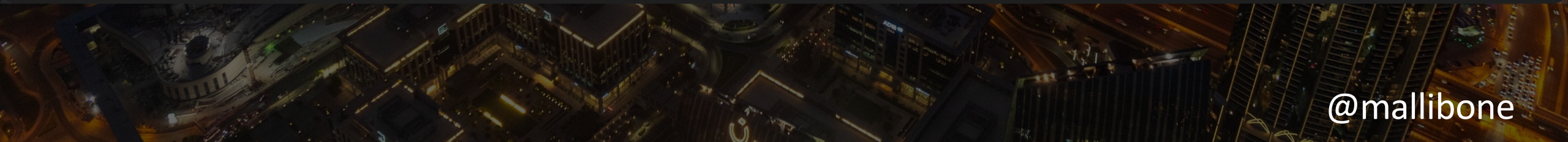
Demo Recap

- Observables are just event streams
- Completion and Error handling
- Declarative Syntax
- Filter, map etc. like LINQ
- Wrap Events into Observables
- Work with multiple event stream
- Timer e.g. for polling

@mallibone

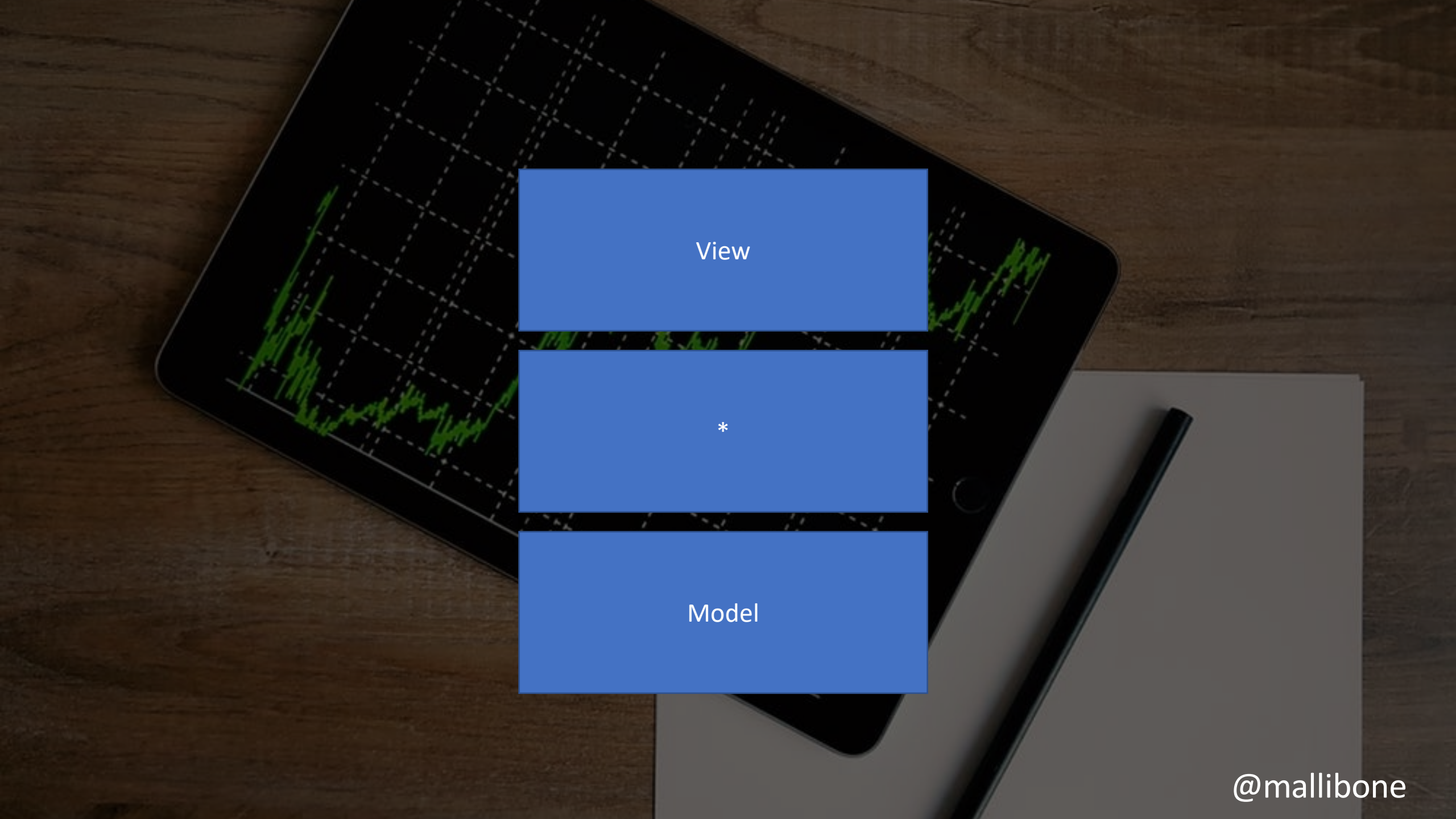


```
var observableSample = new ObservableSample();

observableSample
    .MeasurementChanged
    .Where(update => update.CurrentMeasurement > maxTemperature)
    .Subscribe(HandleTemperatureUpdate);
```

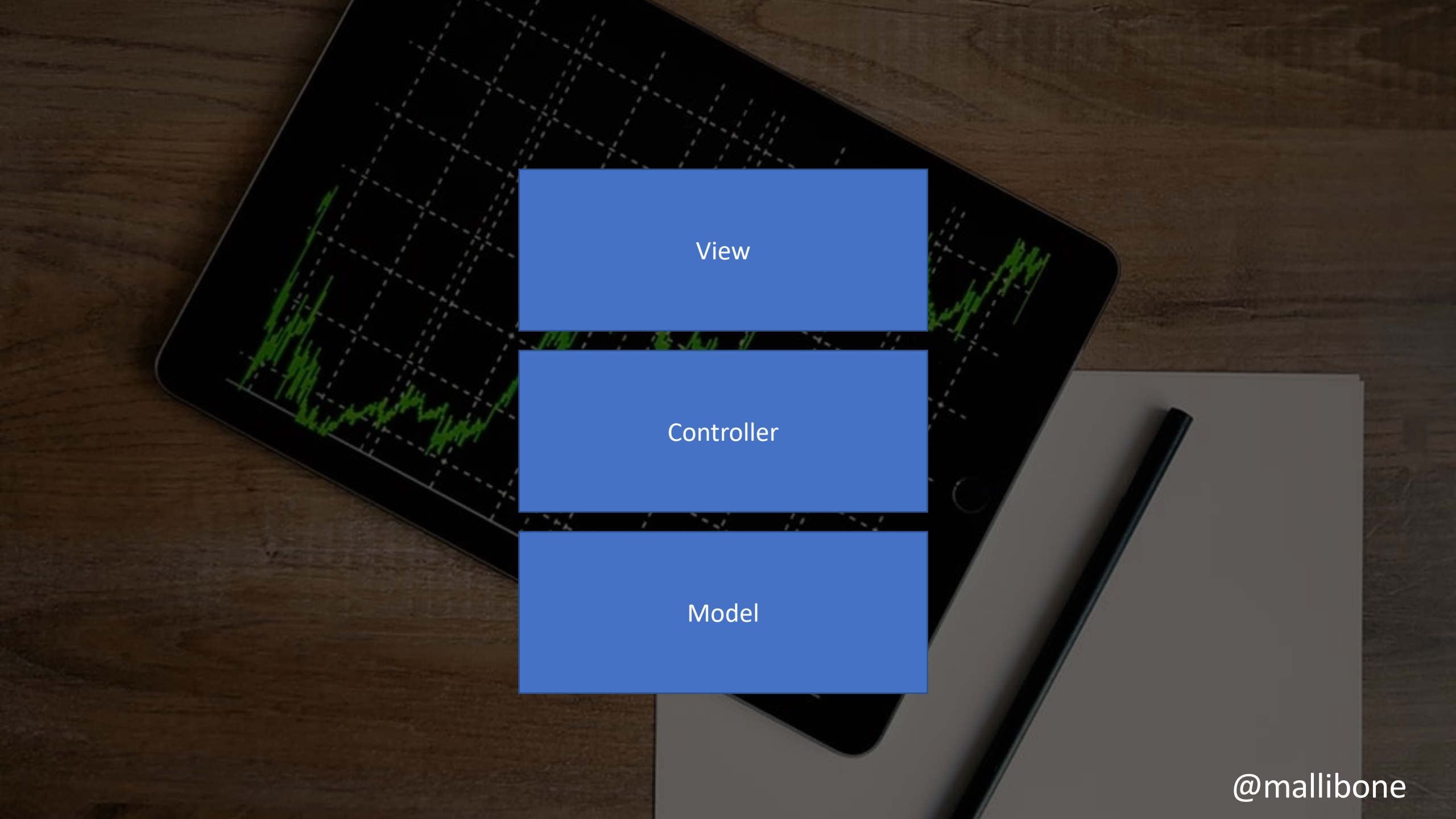


@mallibone

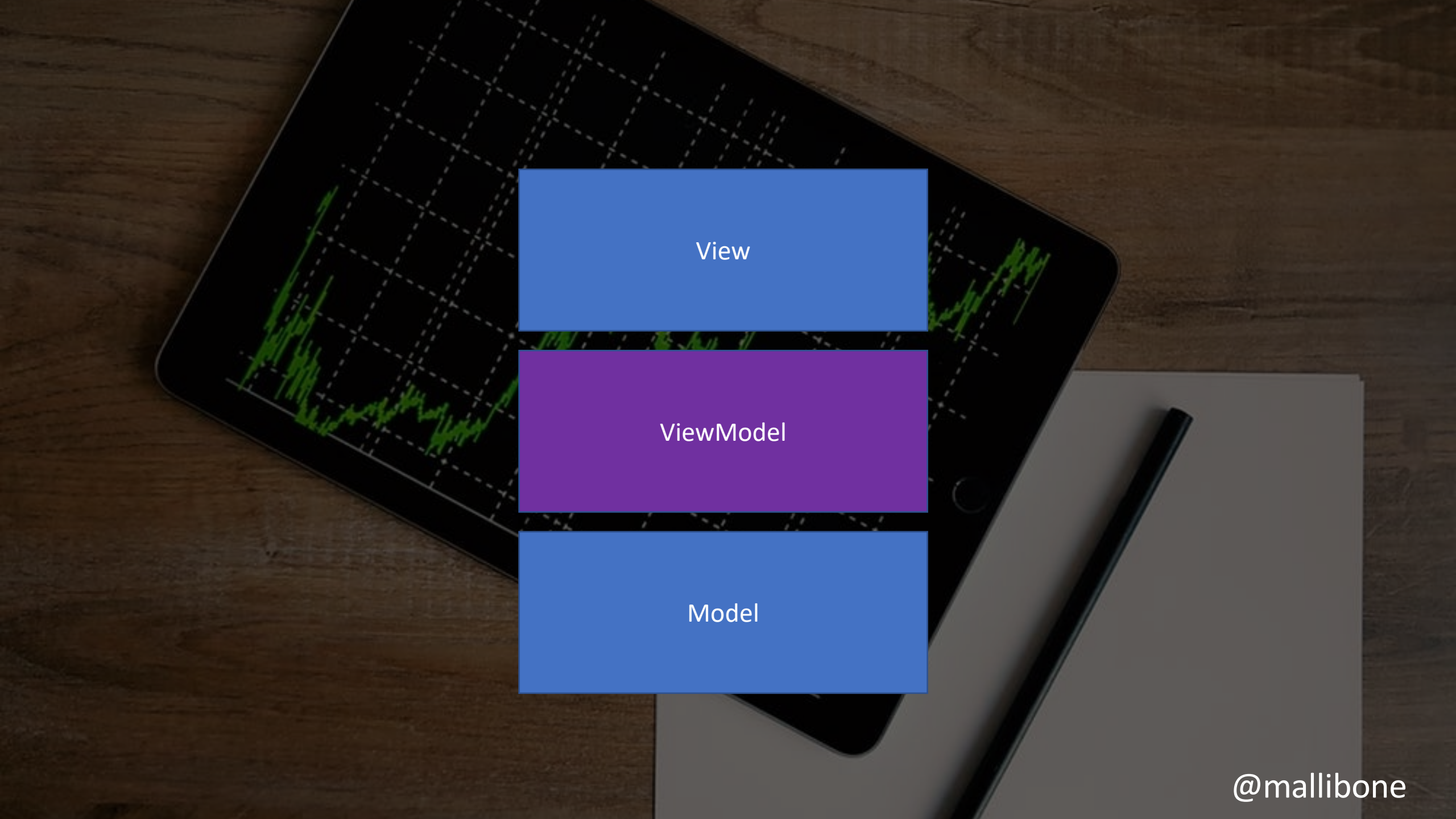

13:43
A-Z
Calculator
Calendar
Call
Camera
Chrome
Clock
Contacts
Daily Mail Online
Downloads
Drive
Dropbox
Duo
eBay
enjoy.now
Facebook
Files
Gmail
Google
Instagram
InstaSize



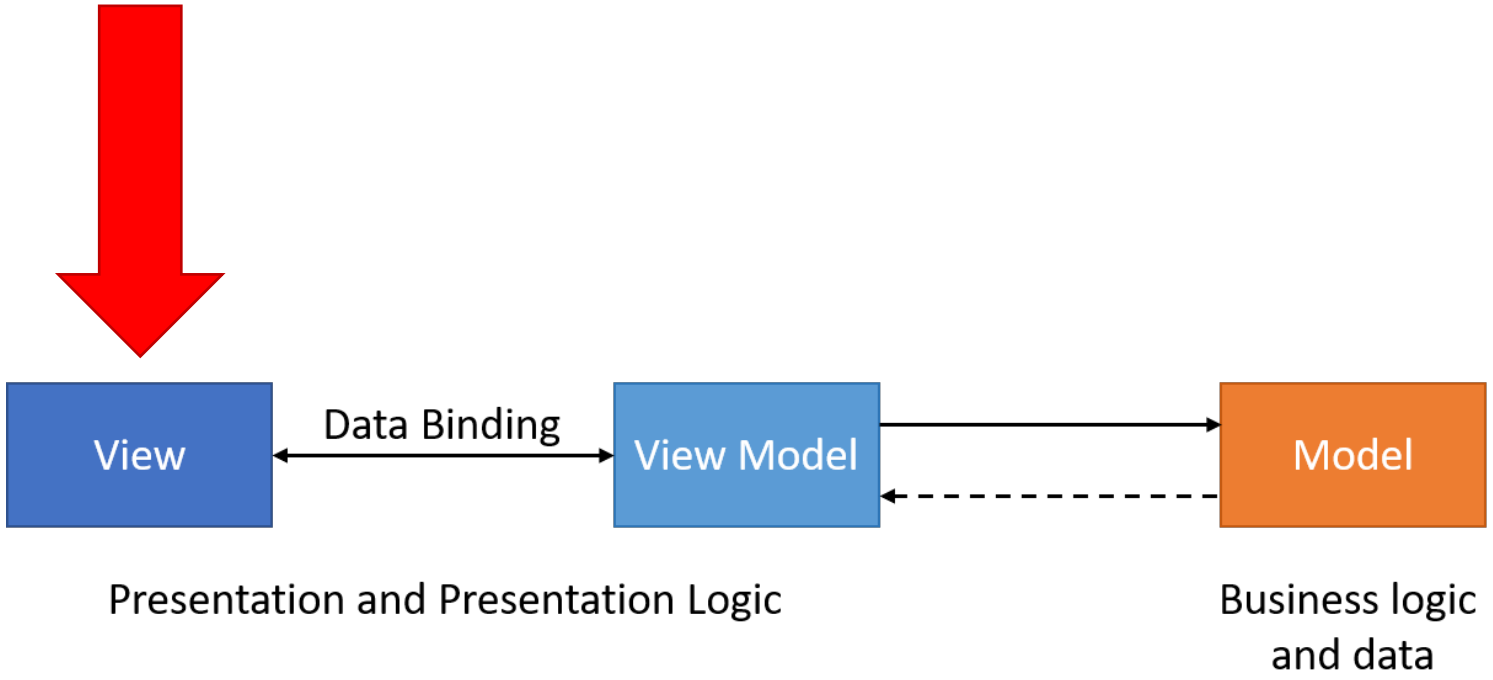
@mallibone


View
Controller
Model
@mallibone

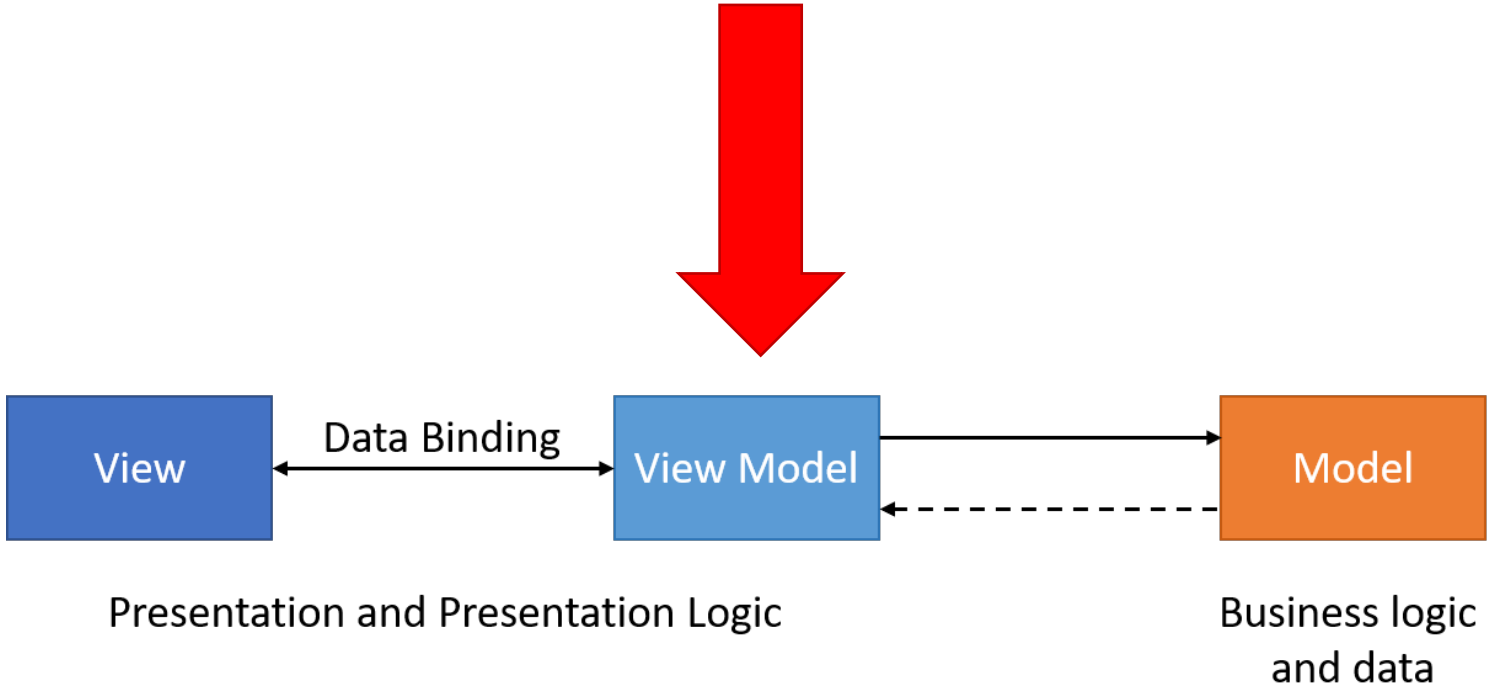
@mallibone


View
ViewModel
Model
@mallibone

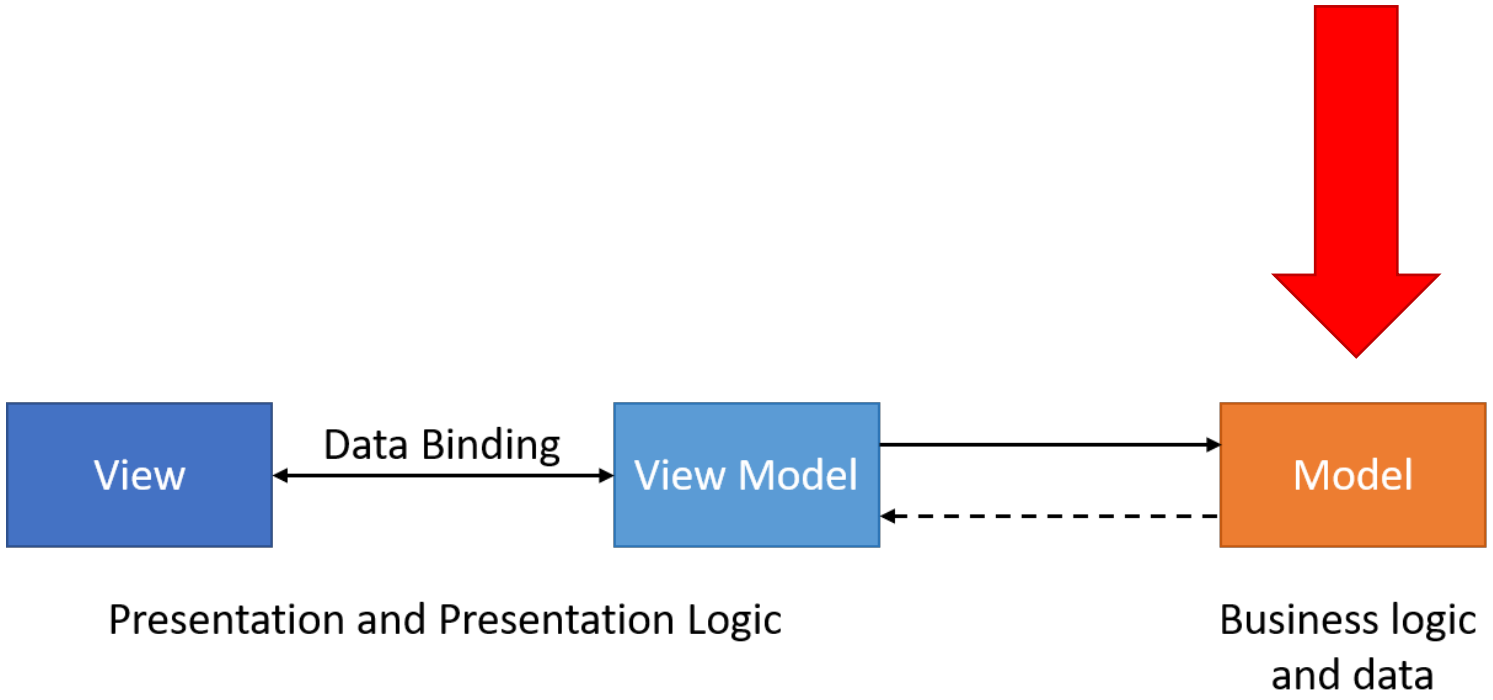
Model View View Model

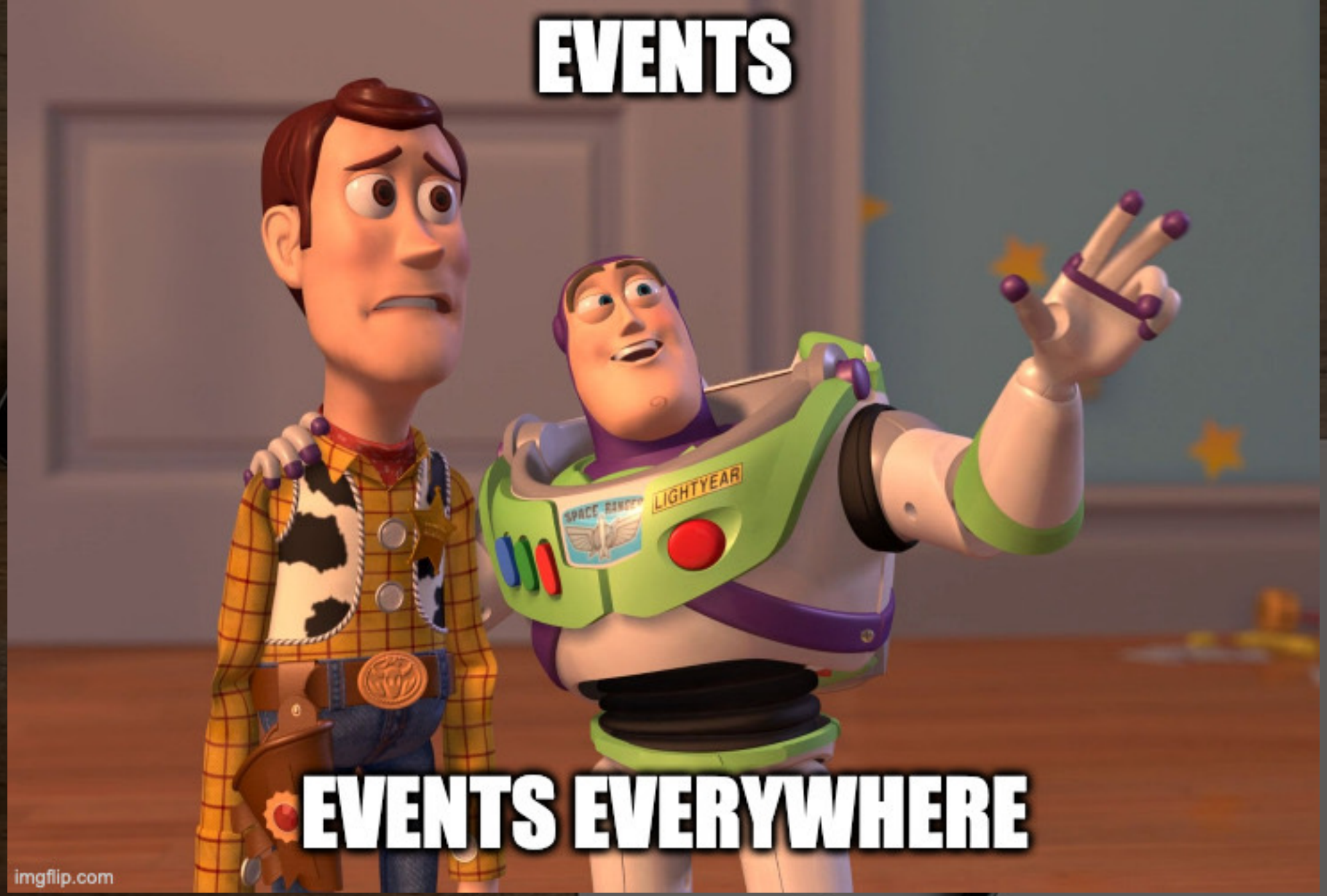


Model View View Model



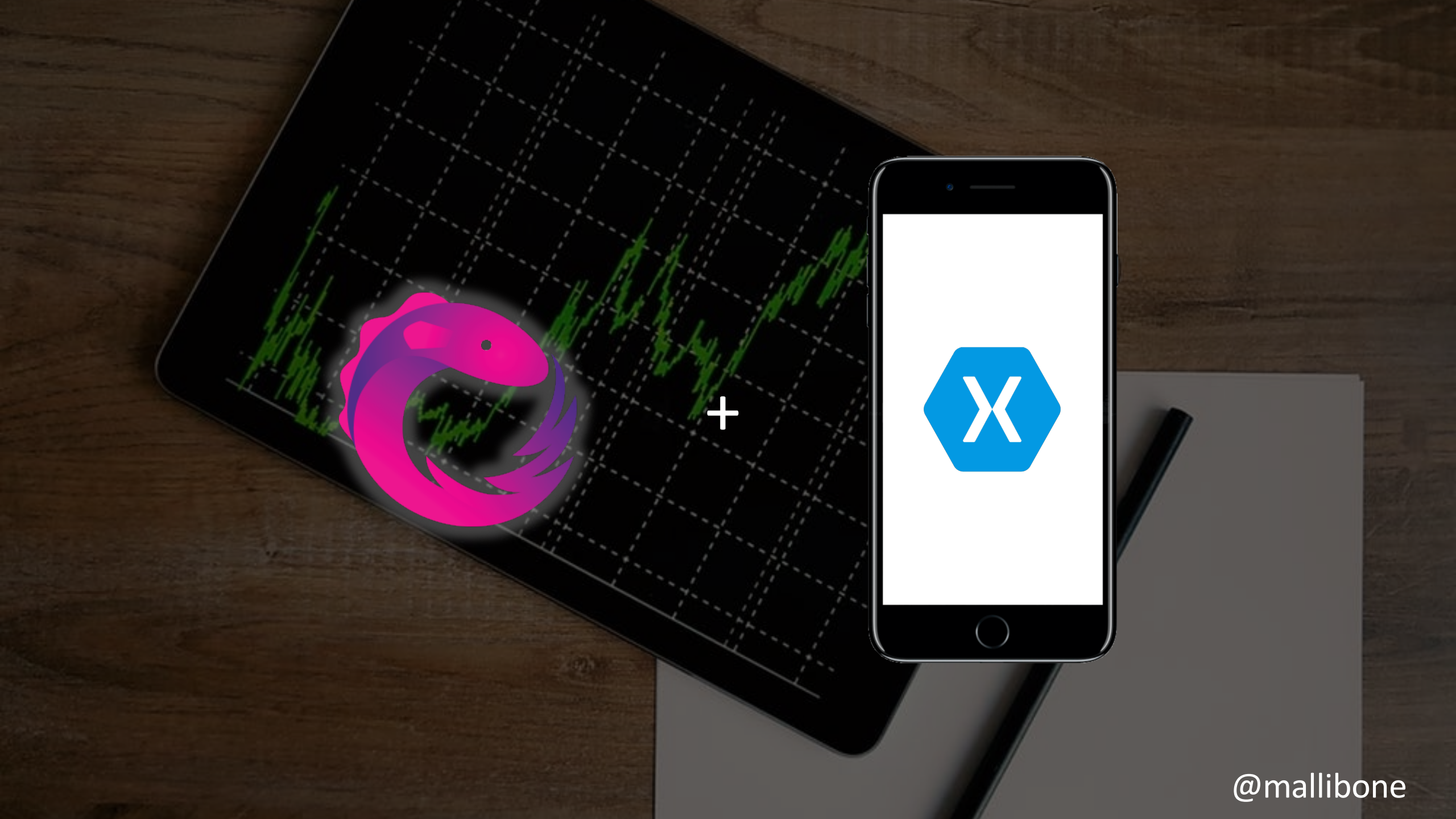
Model View View Model



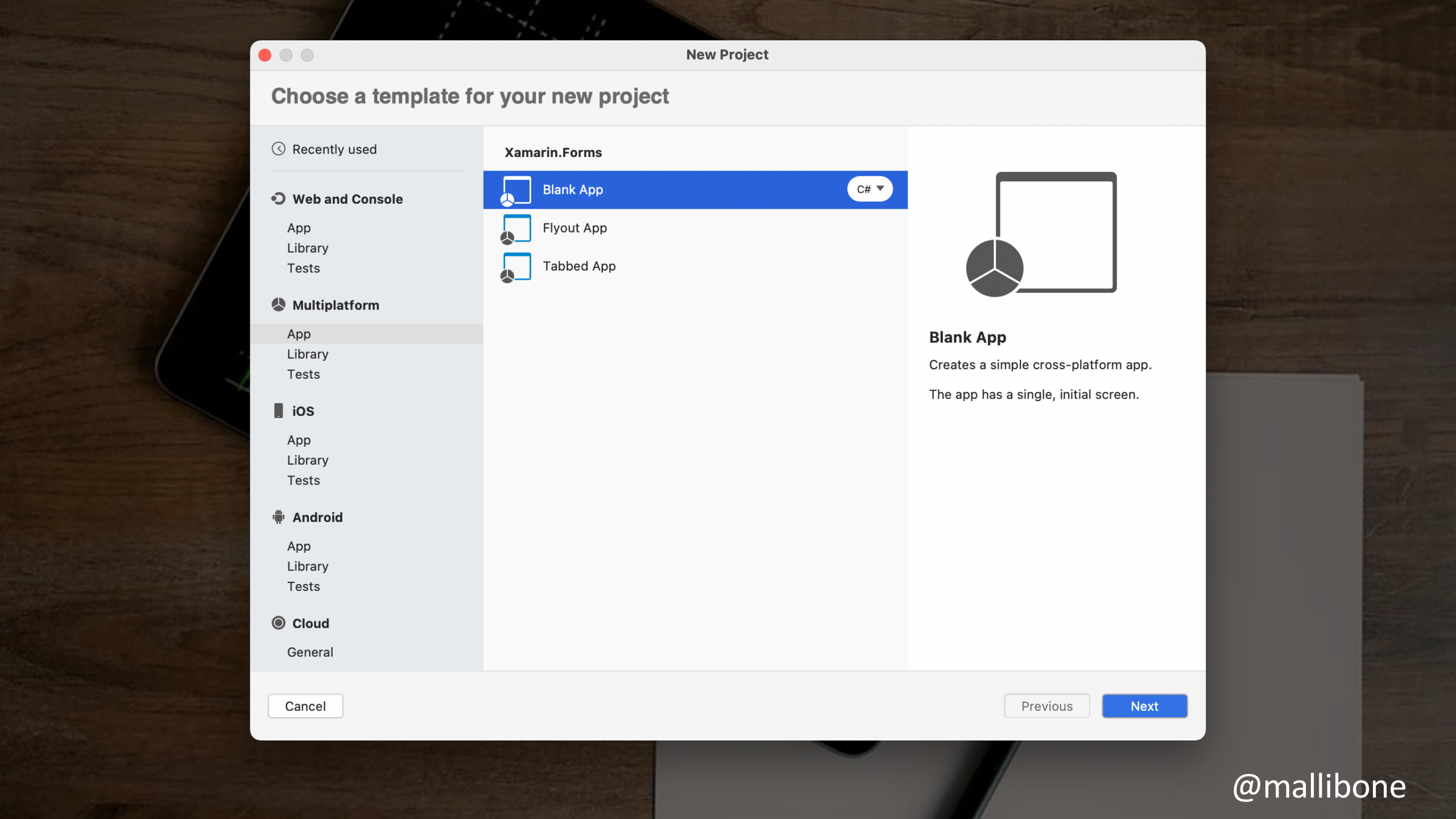


@mallibone

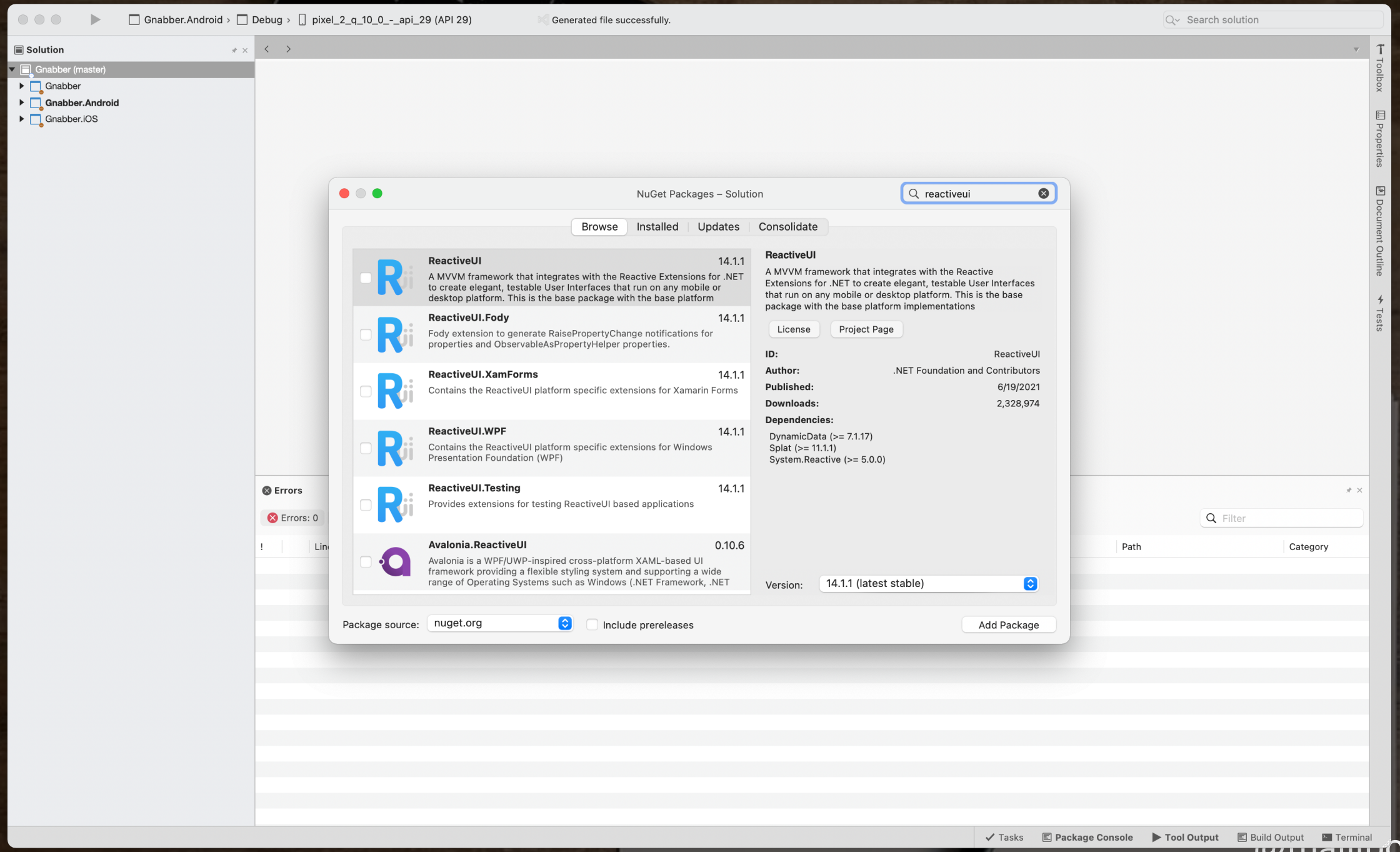

@mallibone

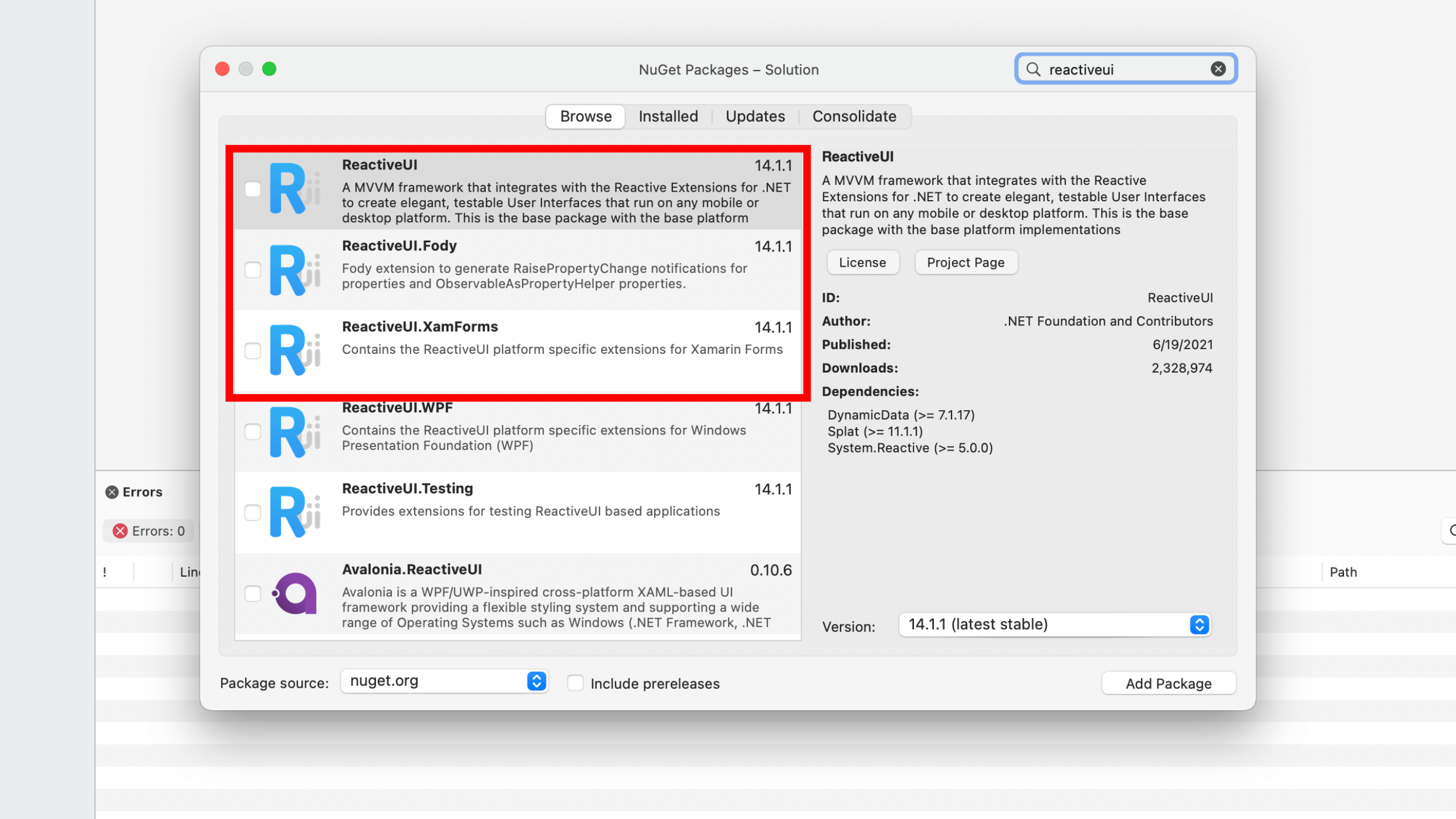

+
@mallibone


+
=
@mallibone


New Project
Choose a template for your new project
Recently used
Web and Console
App
Library
Tests
Multiplatform
App
Library
Tests
iOS
App
Library
Tests
Android
App
Library
Tests
Cloud
General
Xamarin.Forms
Blank App
C#
Flyout App
Tabbed App
Blank App
Creates a simple cross-platform app.
The app has a single, initial screen.
Cancel
Previous
Next

@mallibone


Gnabber.Android › Debug › pixel_2_q_10_0_-_api_29 (API 29)
Generated file successfully.
Search solution
Solution
Gnabber (master)
Gnabber
Gnabber.Android
Gnabber.iOS
NuGet Packages – Solution
reactiveui
Browse
Installed
Updates
Consolidate
ReactiveUI 14.1.1
A MVVM framework that integrates with the Reactive Extensions for .NET to create elegant, testable User Interfaces that run on any mobile or desktop platform. This is the base package with the base platform
ReactiveUI.Fody 14.1.1
Fody extension to generate RaisePropertyChange notifications for properties and ObservableAsPropertyHelper properties.
ReactiveUI.XamForms 14.1.1
Contains the ReactiveUI platform specific extensions for Xamarin Forms
ReactiveUI.WPF 14.1.1
Contains the ReactiveUI platform specific extensions for Windows Presentation Foundation (WPF)
ReactiveUI.Testing 14.1.1
Provides extensions for testing ReactiveUI based applications
Avalonia.ReactiveUI 0.10.6
Avalonia is a WPF/UWP-inspired cross-platform XAML-based UI framework providing a flexible styling system and supporting a wide range of Operating Systems such as Windows (.NET Framework, .NET
ReactiveUI
A MVVM framework that integrates with the Reactive Extensions for .NET to create elegant, testable User Interfaces that run on any mobile or desktop platform. This is the base package with the base platform implementations
License
Project Page
ID: ReactiveUI
Author: .NET Foundation and Contributors
Published: 6/19/2021
Downloads: 2,328,974
Dependencies:
DynamicData (>= 7.1.17)
Splat (>= 11.1.1)
System.Reactive (>= 5.0.0)
Version: 14.1.1 (latest stable)
Package source: nuget.org
Include prereleases
Add Package
Errors
Errors: 0
Filter
Path
Category
Tasks
Package Console
Tool Output
Build Output
Terminal
Toolbox
Properties
Document Outline
Tests


NuGet Packages – Solution
reactiveui
Browse
Installed
Updates
Consolidate
ReactiveUI 14.1.1
A MVVM framework that integrates with the Reactive Extensions for .NET to create elegant, testable User Interfaces that run on any mobile or desktop platform. This is the base package with the base platform
ReactiveUI.Fody 14.1.1
Fody extension to generate RaisePropertyChange notifications for properties and ObservableAsPropertyHelper properties.
ReactiveUI.XamForms 14.1.1
Contains the ReactiveUI platform specific extensions for Xamarin Forms
ReactiveUI.WPF 14.1.1
Contains the ReactiveUI platform specific extensions for Windows Presentation Foundation (WPF)
ReactiveUI.Testing 14.1.1
Provides extensions for testing ReactiveUI based applications
Avalonia.ReactiveUI 0.10.6
Avalonia is a WPF/UWP-inspired cross-platform XAML-based UI framework providing a flexible styling system and supporting a wide range of Operating Systems such as Windows (.NET Framework, .NET
ReactiveUI
A MVVM framework that integrates with the Reactive Extensions for .NET to create elegant, testable User Interfaces that run on any mobile or desktop platform. This is the base package with the base platform implementations
License
Project Page
ID: ReactiveUI
Author: .NET Foundation and Contributors
Published: 6/19/2021
Downloads: 2,328,974
Dependencies:
DynamicData (>= 7.1.17)
Splat (>= 11.1.1)
System.Reactive (>= 5.0.0)
Version: 14.1.1 (latest stable)
Package source: nuget.org
Include prereleases
Add Package
Errors
Errors: 0
Path

ViewModels 101

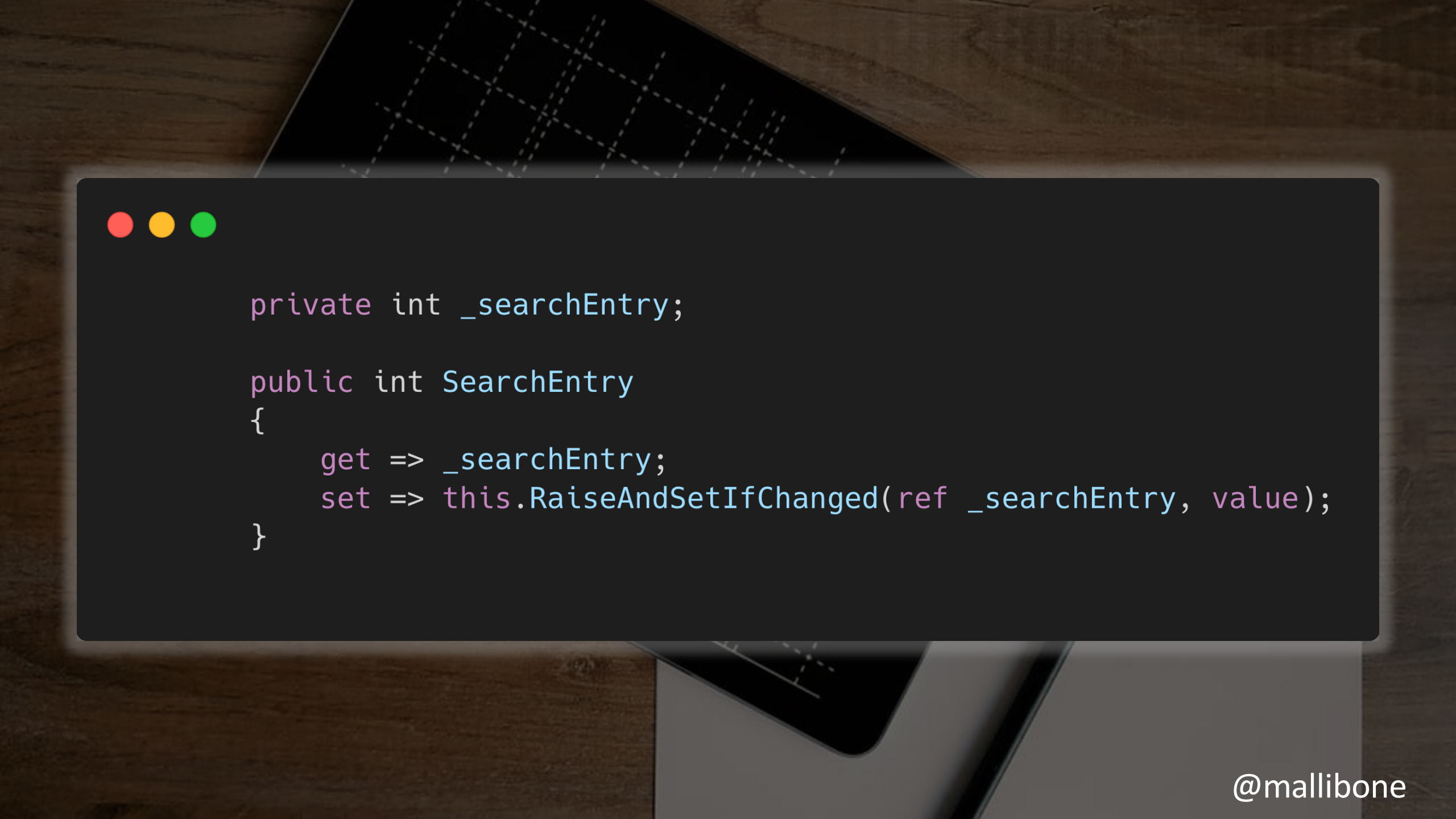
- Properties
- Commands
- Model (Stuff)

@mallibone

ViewModels 101

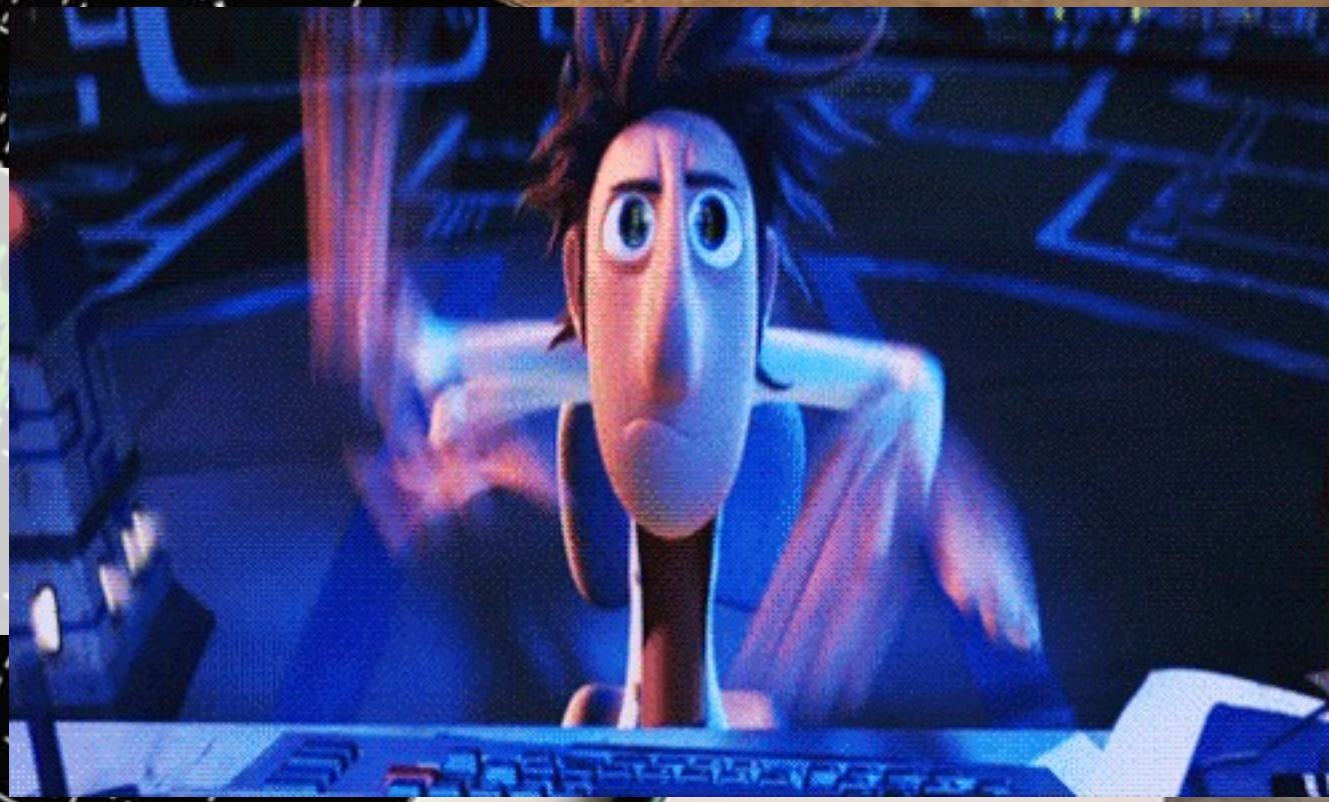
- **Properties**
- Commands
- Model (Stuff)

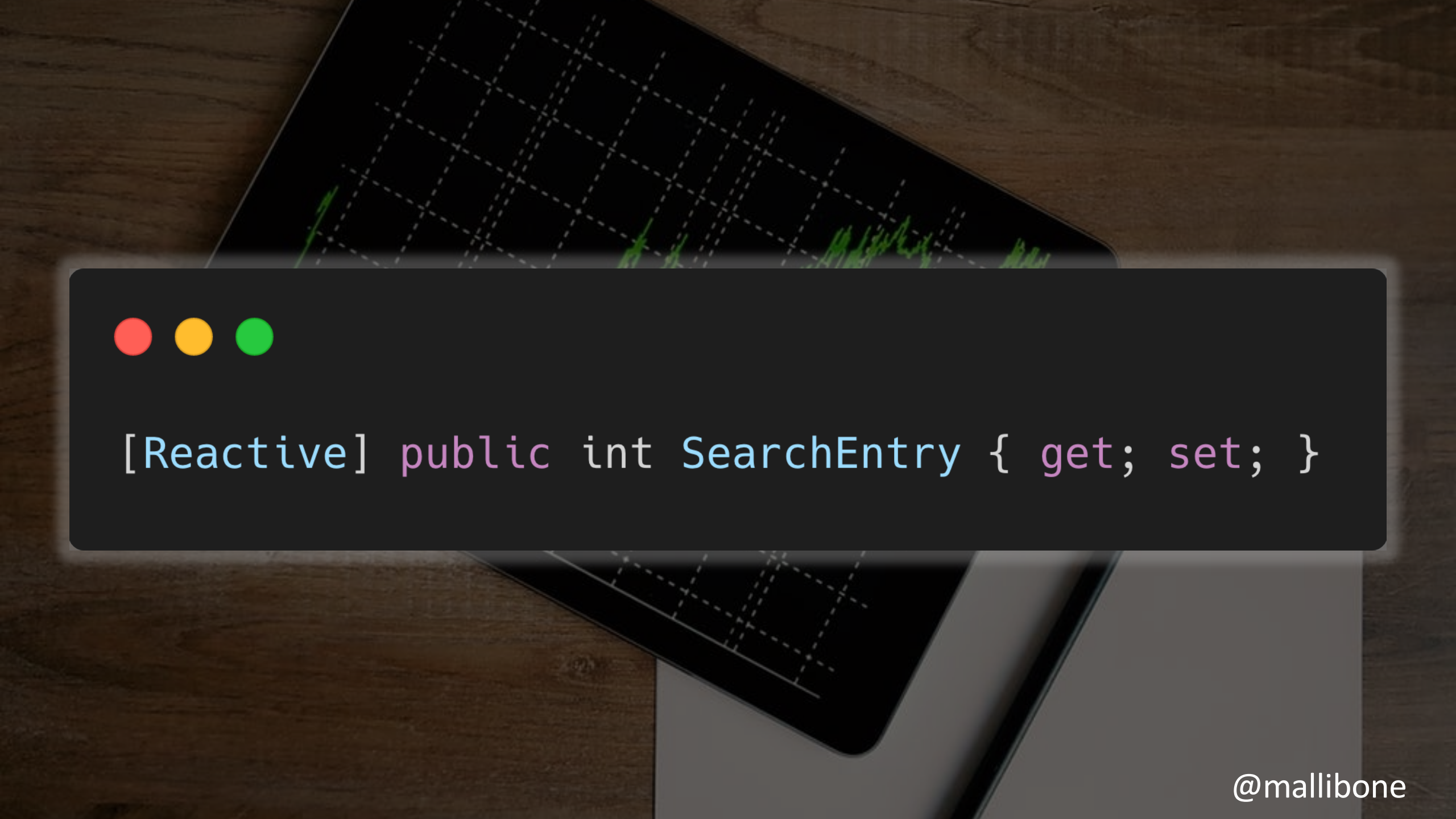
@mallibone



```
private int _searchEntry;

public int SearchEntry
{
    get => _searchEntry;
    set => this.RaiseAndSetIfChanged(ref _searchEntry, value);
}
```

So much typing..




[Reactive] public int SearchEntry { get; set; }
@mallibone

ViewModels 101

- Properties
- **Commands**
- Model (Stuff)

@mallibone

Synchronous Commands

```
// Your standard ICommand
ICommand syncCommand = ReactiveCommand.Create(() => "Sync");

// Can be subscribed to
ReactiveCommand<Unit, string> syncCommand = ReactiveCommand.Create(() => "Sync");
```

Synchronous Commands

```
// Your standard ICommand
ICommand syncCommand = ReactiveCommand.Create(() => "Sync");

// Can be subscribed to
ReactiveCommand<Unit, string> syncCommand = ReactiveCommand.Create(() => "Sync");
```

Synchronous Commands

```
// Your standard ICommand
ICommand syncCommand = ReactiveCommand.Create(() => "Sync");

// Can be subscribed to
ReactiveCommand<Unit, string> syncCommand = ReactiveCommand.Create(() => "Sync");
```

Async Commands

```
// When you await a task
var asyncCommand = ReactiveCommand.CreateFromTask(() => Task.FromResult("Hello"));

// When you start working with observables
var observableCommand = ReactiveCommand.CreateFromObservable(() => Observable.Return("There"));
```

@mallibone

Async Commands

```
// When you await a task
var asyncCommand = ReactiveCommand.CreateFromTask(() => Task.FromResult("Hello"));

// When you start working with observables
var observableCommand = ReactiveCommand.CreateFromObservable(() => Observable.Return("There"));
```

@mallibone

Combined Commands

```
// List of Commands
var commands = List<ReactiveCommand<Unit, string>>{asyncCommand, observableCommand, syncCommand};

// Combined Commands
CombinedReactiveCommand<Unit, string> combindedCommand = ReactiveCommand.CreateCombined(commands);
```

@mallibone

Combined Commands

```
// List of Commands
var commands = List<ReactiveCommand<Unit, string>>{asyncCommand, observableCommand, syncCommand};

// Combined Commands
CombinedReactiveCommand<Unit, string> combindedCommand = ReactiveCommand.CreateCombined(commands);
```

@mallibone

Combined Commands

```
// List of Commands
var commands = List<ReactiveCommand<Unit, string>>{asyncCommand, observableCommand, syncCommand};

// Combined Commands
CombinedReactiveCommand<Unit, string> combindedCommand = ReactiveCommand.CreateCombined(commands);
```

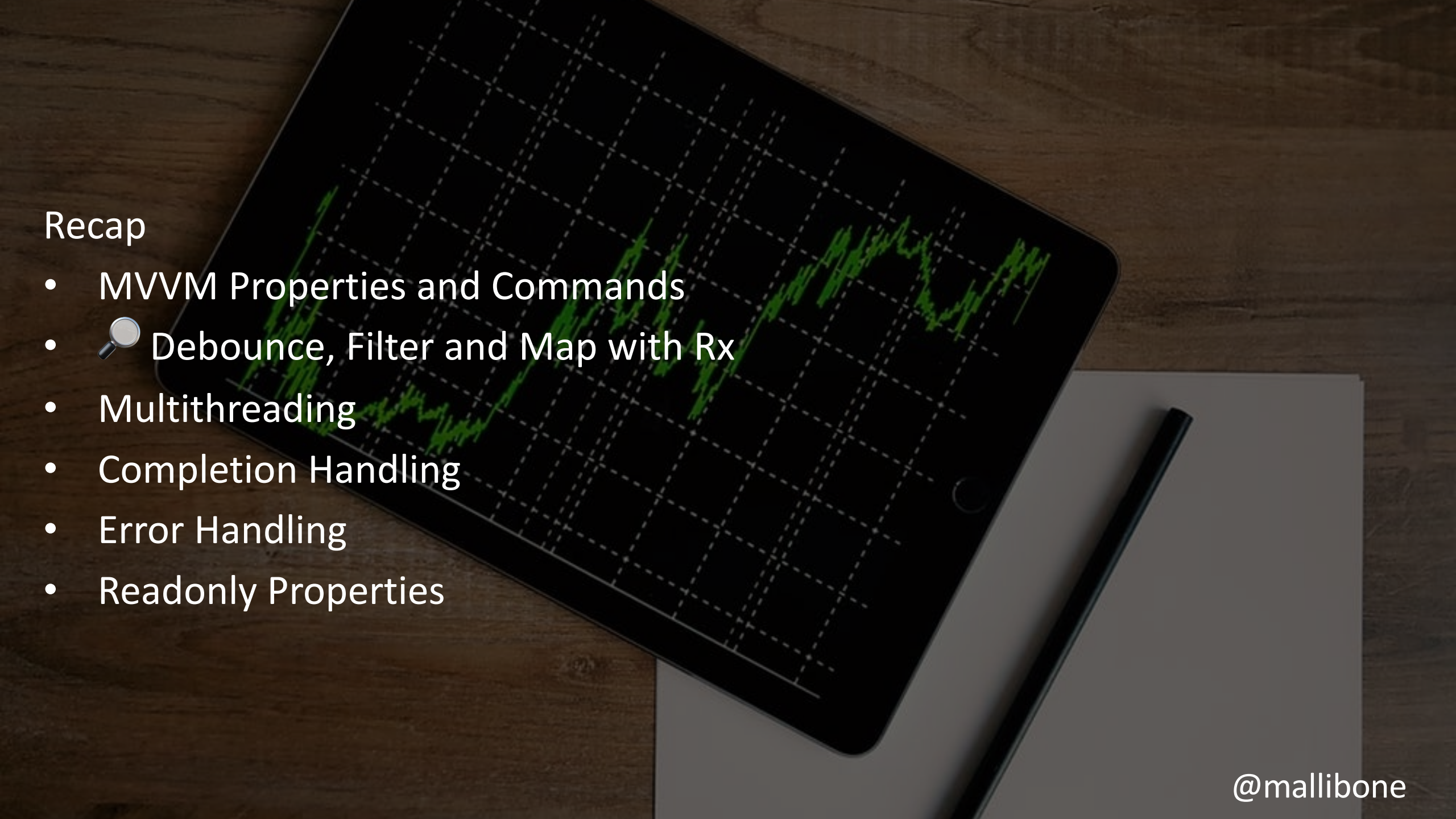
ViewModels 101

- Properties
- Commands
- Model (Stuff)

@mallibone

Rx101 · 2 projects
Rx101
Dependencies
Helpers
EventSample.cs
MeasurementUpdate.cs
ObservableSample.cs
Demo01.cs
Demo02.cs
Demo03.cs
Demo04.cs
Program.cs
Rx101.Test
Dependencies
UnitTest1.cs
Scratches and Consoles

```
namespace Rx101
{
    1 usage  Mark Allibone *
    public static class Demo01
    {
        1 usage  Mark Allibone *
        public static void SimpleComparison()
        {
            Console.WriteLine("Simple Event comparison");
            RunEventSample();
            RunObservableSample();
        }

        1 usage  new *
        private static void RunObservableSample()
        {
            var observableSample = new ObservableSample();
            var measurmentChangedSubscription :IDisposable =
                observableSample.MeasurementChanged.Subscribe( onNext: update =>
                    Console.WriteLine($"Temperature update {update.CurrentMeasurement}"));
            observableSample.NewMeasurementReading( temperature: 24.0f);
            measurmentChangedSubscription.Dispose();

        }

        1 usage  new *
        private static void RunEventSample()
        {
            void EventSampleOnMeasurementChanged(object sender, MeasurementUpdate update) =>
                Console.WriteLine($"Temperature update {update.CurrentMeasurement}");

            var eventSample = new EventSample();
            eventSample.MeasurementChanged += EventSampleOnMeasurementChanged;
            eventSample.NewMeasruementReading( measurement: 22.0f);
            eventSample.MeasurementChanged -= EventSampleOnMeasurementChanged;
        }
    }
}
```



Recap

- MVVM Properties and Commands
- 🔎 Debounce, Filter and Map with Rx
- Multithreading
- Completion Handling
- Error Handling
- Readonly Properties

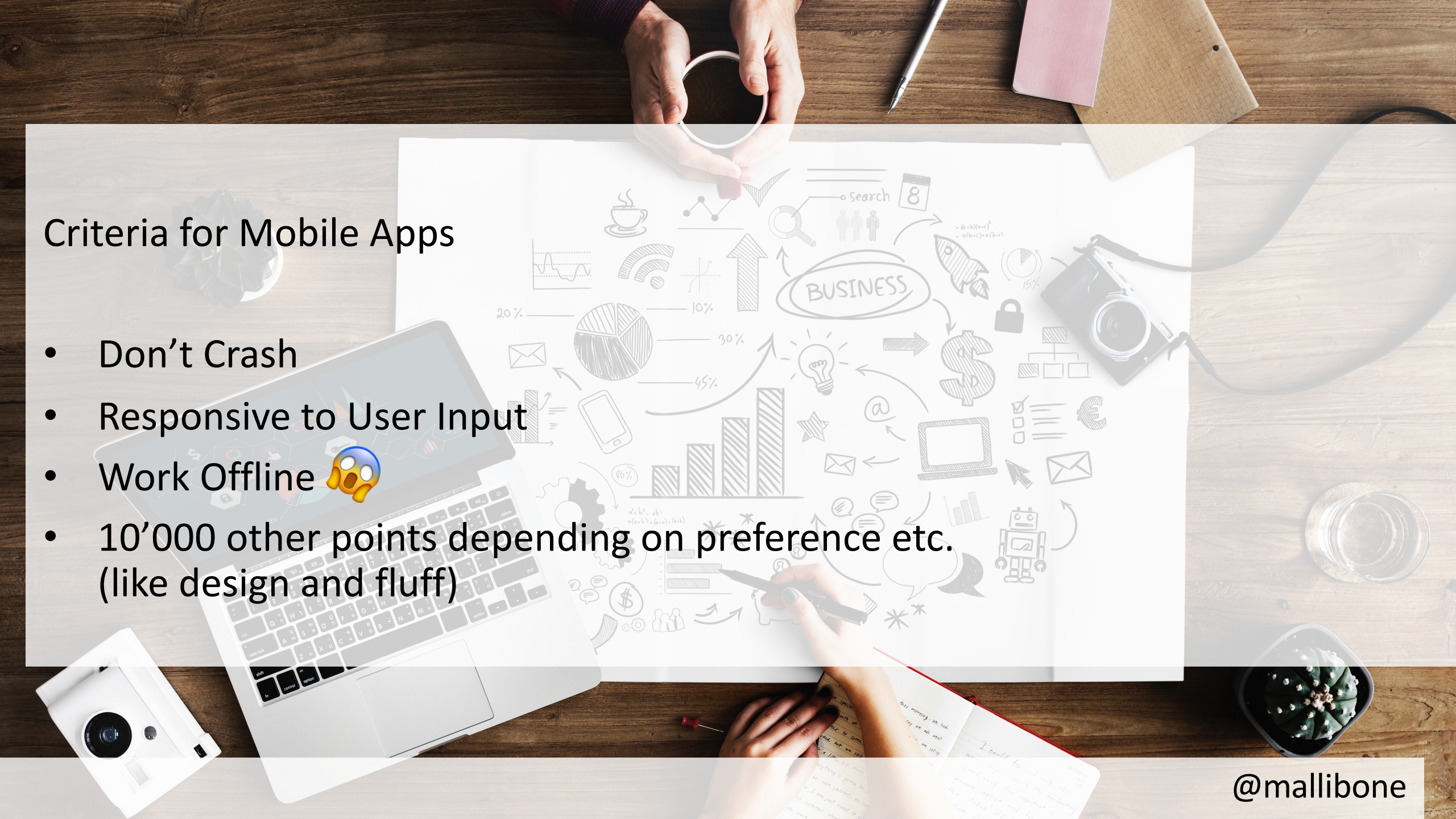
@mallibone


Rui
@mallibone


Validation
Navigation
Event Extensions
Auto Persistance
= Rx + MVVM + “a ton of helpers™
View Lifecycle
Dynamic Collections
DI Container
User Confirmation
@mallibone

Criteria for Mobile Apps

@mallibone



Criteria for Mobile Apps

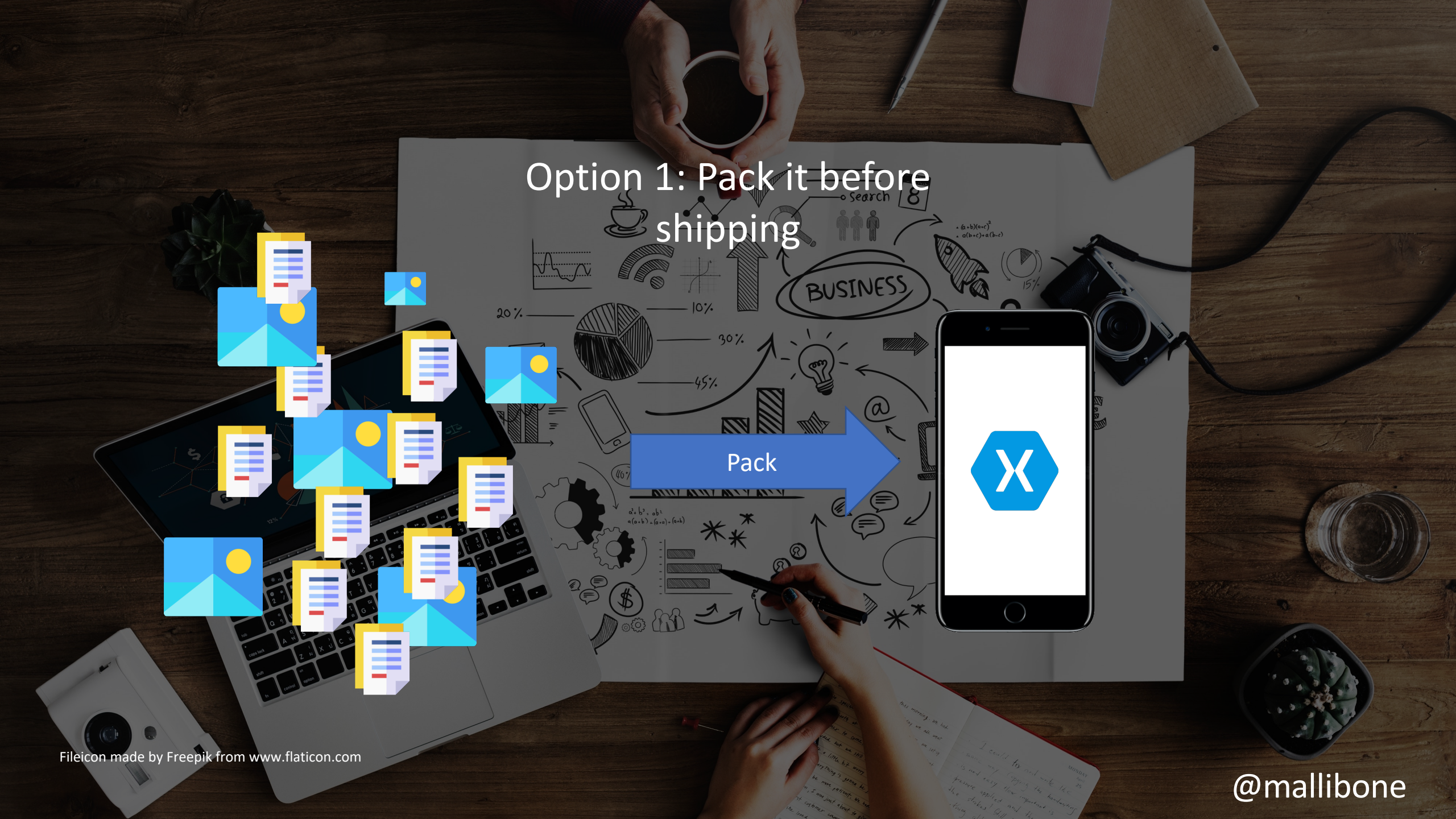
- Don’t Crash
- Responsive to User Input
- Work Offline 😱
- 10’000 other points depending on preference etc.
  (like design and fluff)

@mallibone

How to build an App that works offline?

@mallibone

Option 1: Pack it before shipping



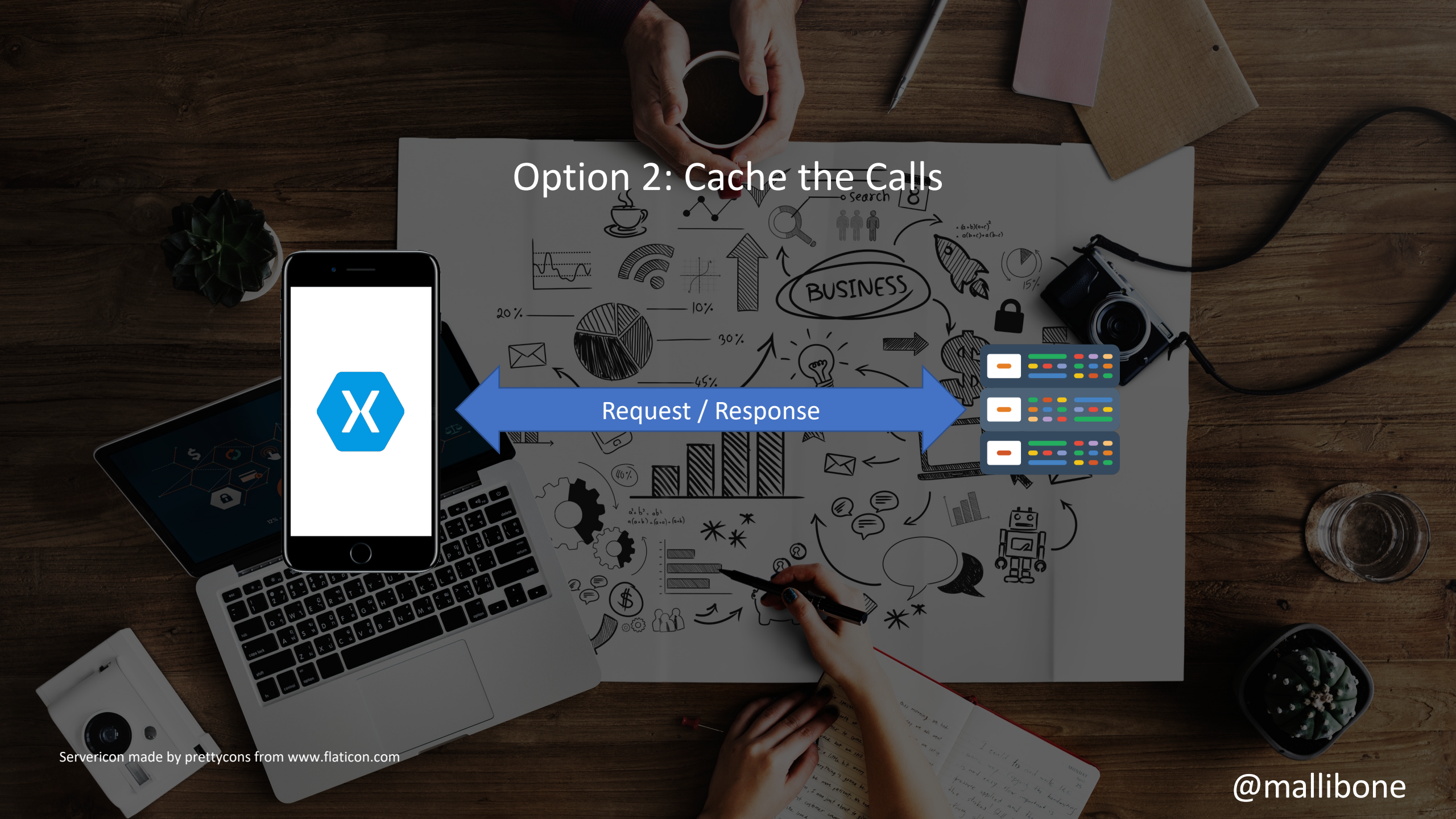
Fileicon made by Freepik from www.flaticon.com

@mallibone


COULD YOU SHOW ME THE CURRENT WEATHER
THAT WOULD BE REALLY GREAT
imgflip.com

@mallibone

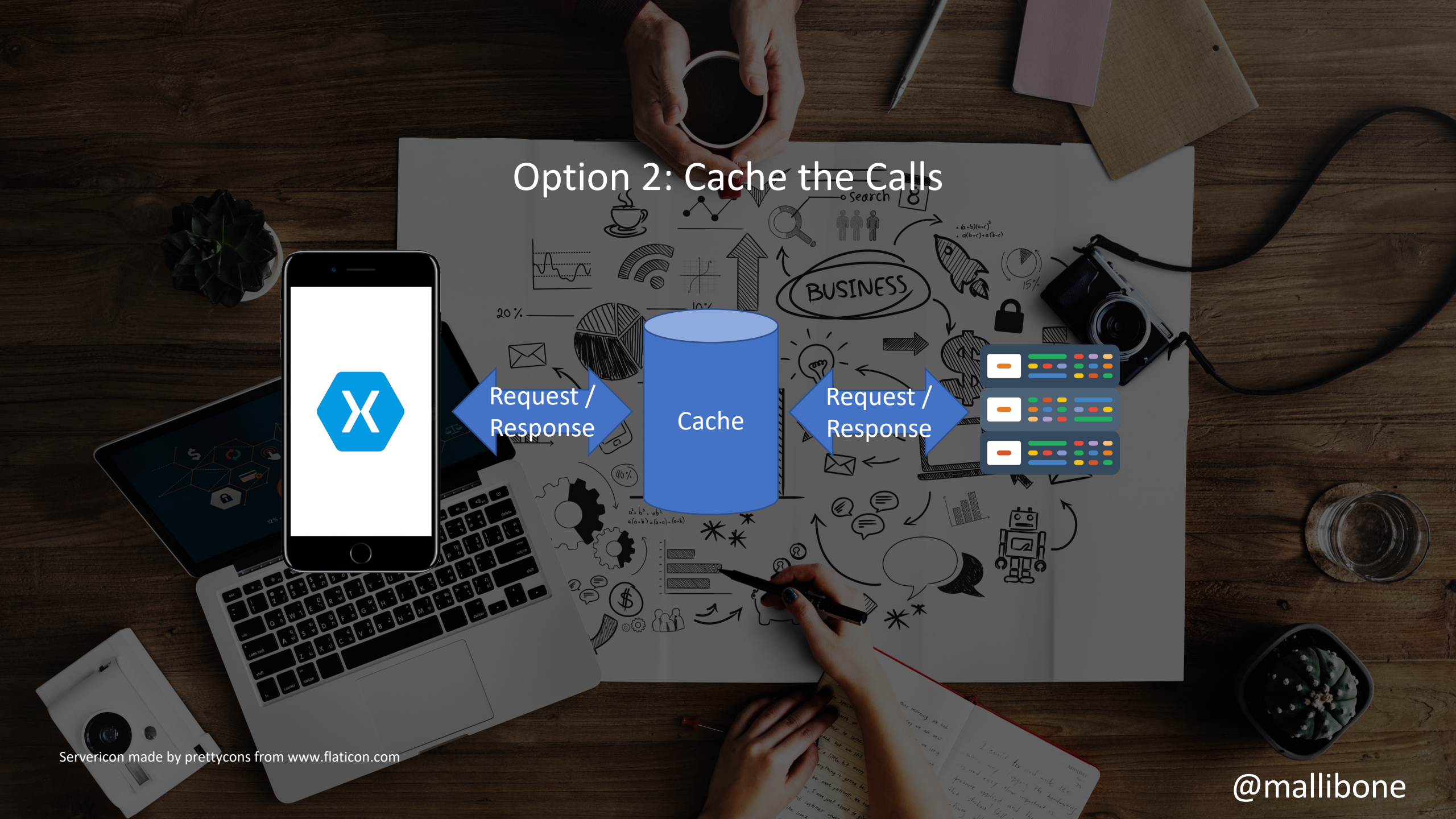
Option 2: Cache the Calls



Servericon made by prettycons from www.flaticon.com

@mallibone

Option 2: Cache the Calls



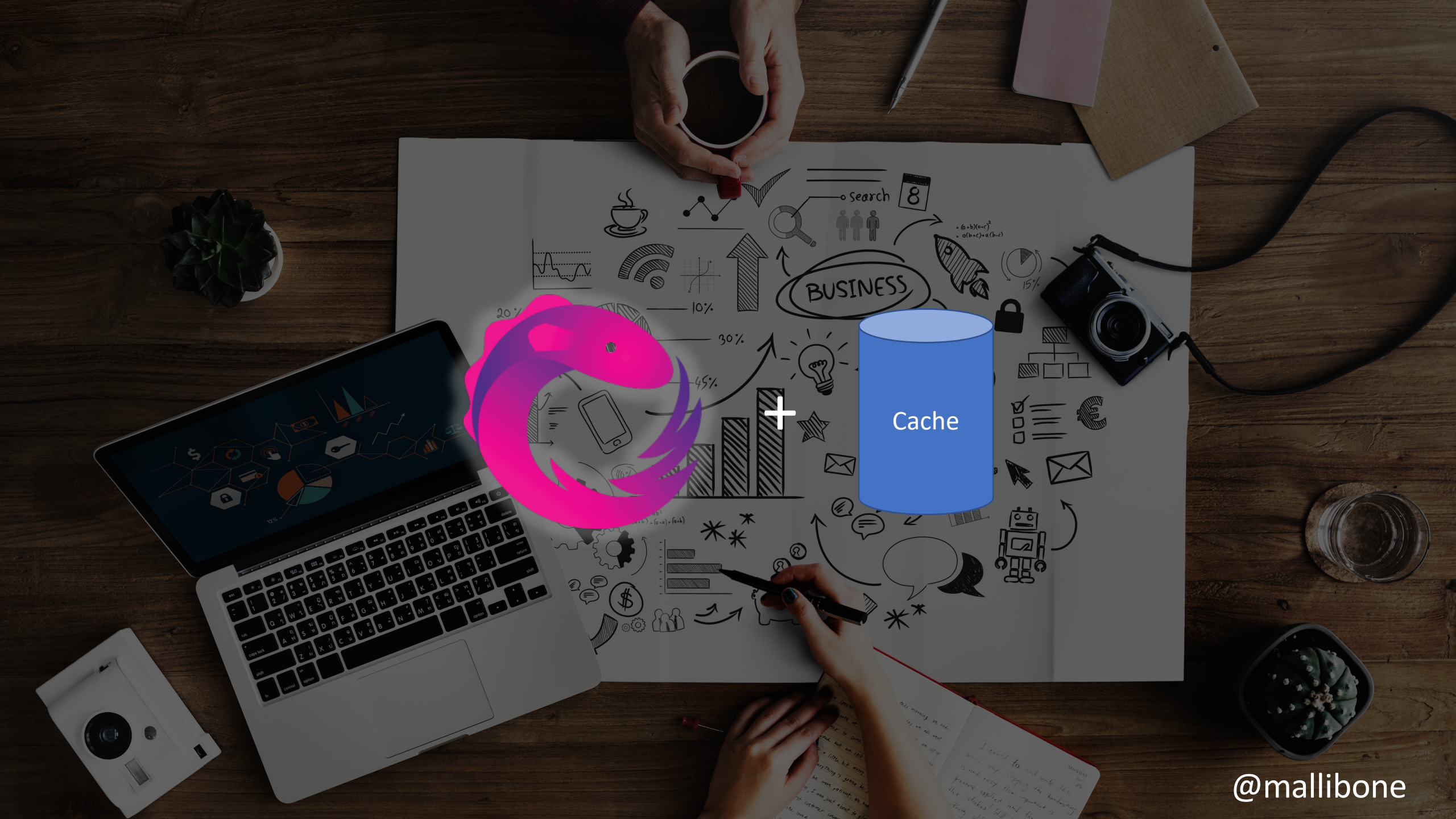
Servericon made by prettycons from www.flaticon.com

@mallibone


MOBILE DEV
CACHING
COMPUTER SCIENCE 101
imgflip.com


When do we refresh the cache?
@mallibone


Search
BUSINESS
Cache
@mallibone

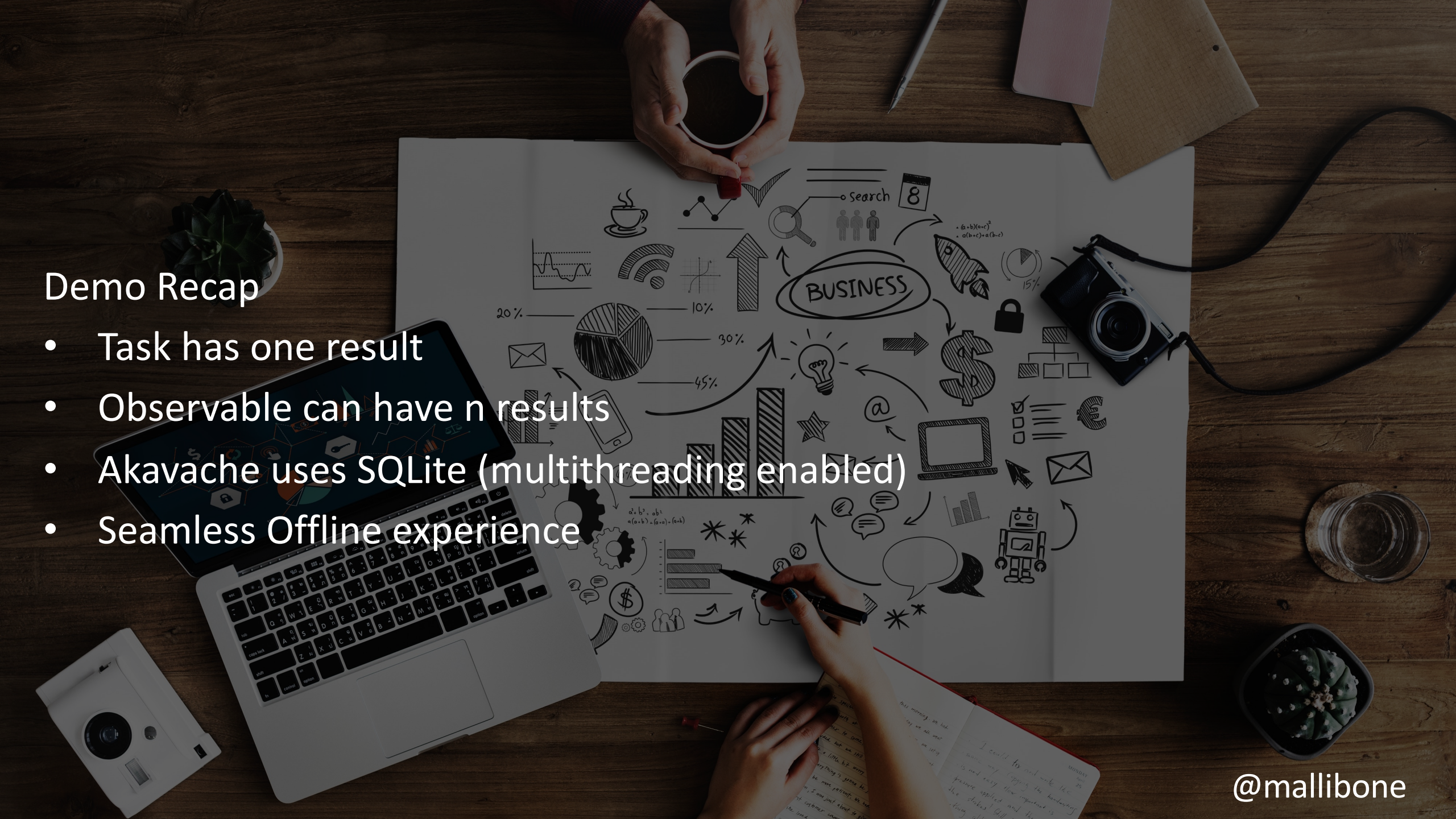

Search
BUSINESS
Cache
@mallibone


Search
BUSINESS
Cache
@mallibone

Rx101 · 2 projects
Rx101
Dependencies
Helpers
EventSample.cs
MeasurementUpdate.cs
ObservableSample.cs
Demo01.cs
Demo02.cs
Demo03.cs
Demo04.cs
Program.cs
Rx101.Test
Dependencies
UnitTest1.cs
Scratches and Consoles

```
namespace Rx101
{
    1 usage  Mark Allibone *
    public static class Demo01
    {
        1 usage  Mark Allibone *
        public static void SimpleComparison()
        {
            Console.WriteLine("Simple Event comparison");
            RunEventSample();
            RunObservableSample();
        }

        1 usage  new *
        private static void RunObservableSample()
        {
            var observableSample = new ObservableSample();
            var measurmentChangedSubscription :IDisposable =
                observableSample.MeasurementChanged.Subscribe( onNext: update =>
                    Console.WriteLine($"Temperature update {update.CurrentMeasurement}"));
            observableSample.NewMeasurementReading( temperature: 24.0f);
            measurmentChangedSubscription.Dispose();

        }

        1 usage  new *
        private static void RunEventSample()
        {
            void EventSampleOnMeasurementChanged(object sender, MeasurementUpdate update) =>
                Console.WriteLine($"Temperature update {update.CurrentMeasurement}");

            var eventSample = new EventSample();
            eventSample.MeasurementChanged += EventSampleOnMeasurementChanged;
            eventSample.NewMeasruementReading( measurement: 22.0f);
            eventSample.MeasurementChanged -= EventSampleOnMeasurementChanged;
        }
    }
}
```

@mallibone



Demo Recap

- Task has one result
- Observable can have n results
- Akavache uses SQLite (multithreading enabled)
- Seamless Offline experience

@mallibone

Takeaways

Takeaways

- Reactive Extensions (Rx) helps you with everyday development
- Reactive UI brings Rx to the UI through MVVM
- Rx can be used for seemless caching with Akavache
- Don’t forget to `Subscribe` and `Dispose`

REY TECHNOLOGY

Thank you for your time!

Mark Allibone @mallibone

Rey Technology

FEATURED COMMUNITY BLOG https://mallibone.com

https://nullpointers.io

http://reactivex.io/

https://www.reactiveui.net/

MVP Microsoft® Most Valuable Professional