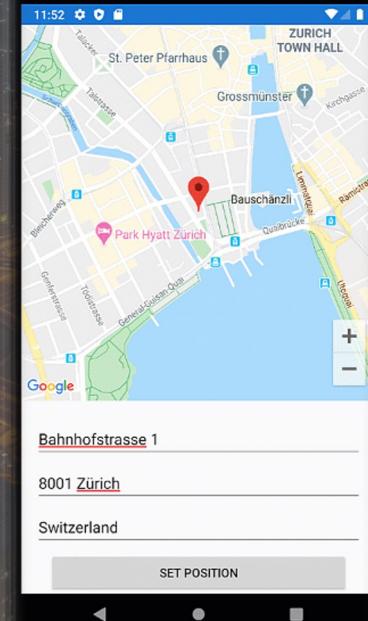


# Writing Reactive .NET MAUI apps

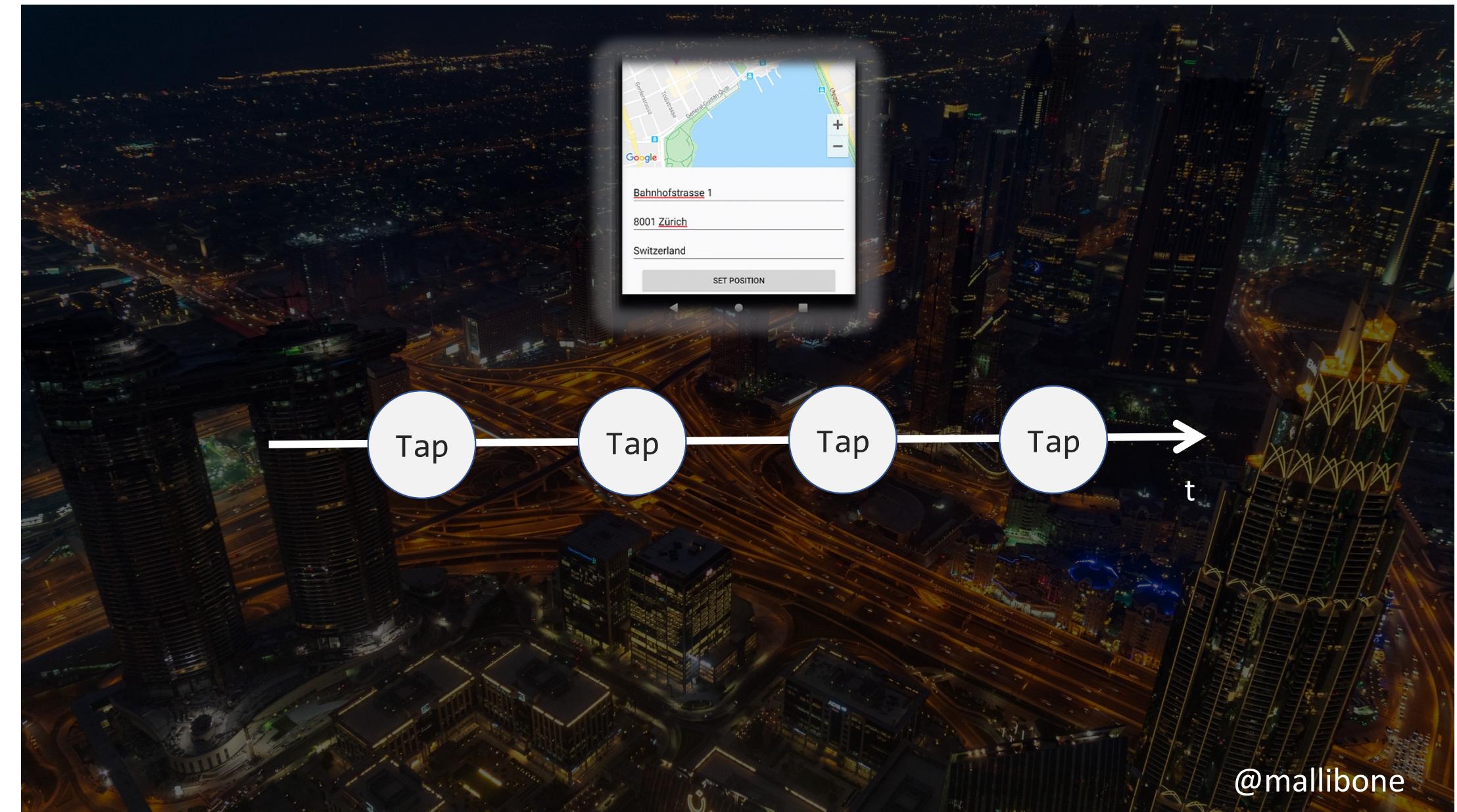
Mark Allibone  
Rey Technology

@mallibone



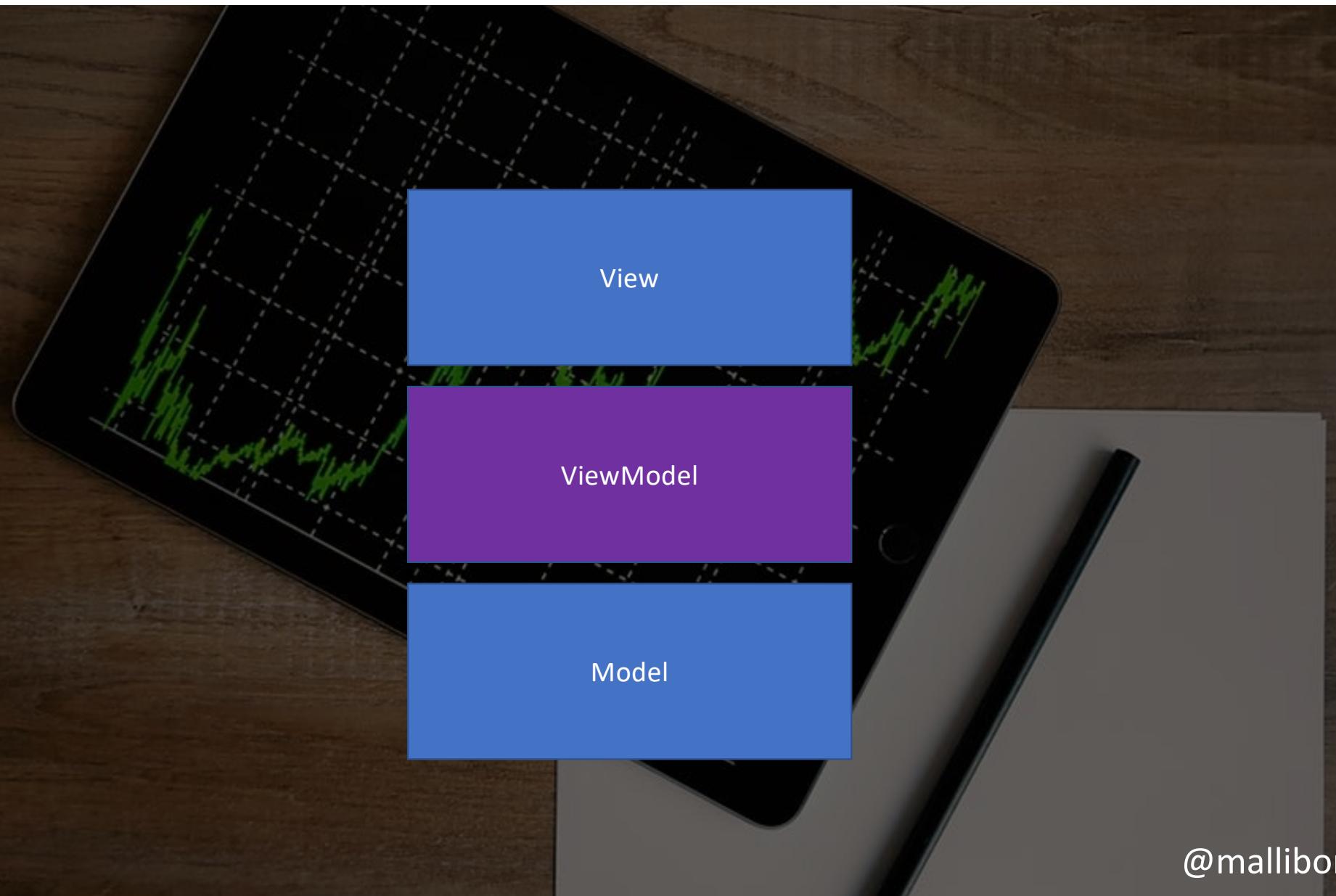


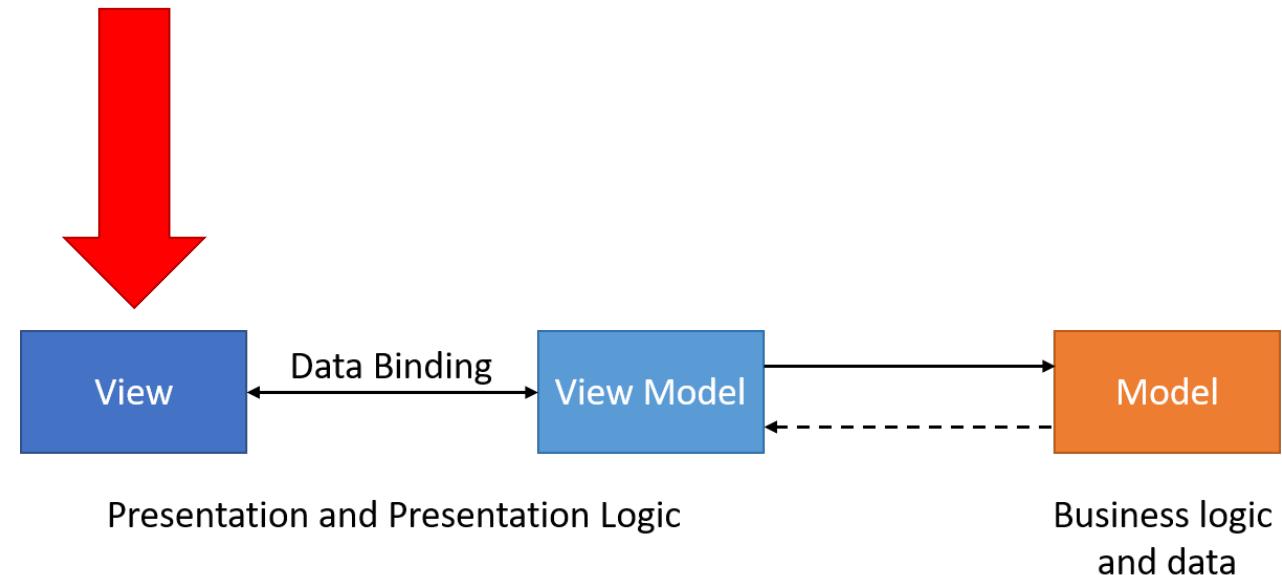
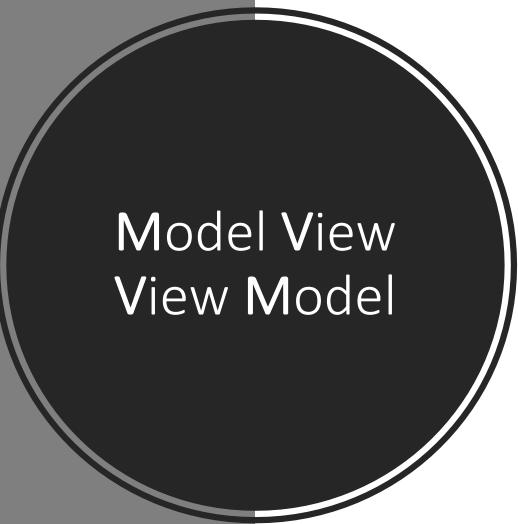
@mallibone

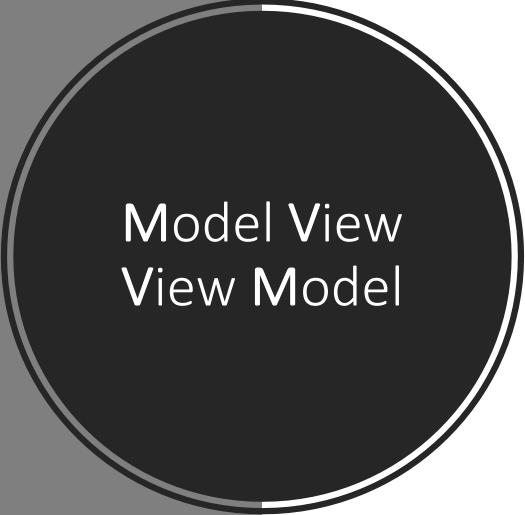




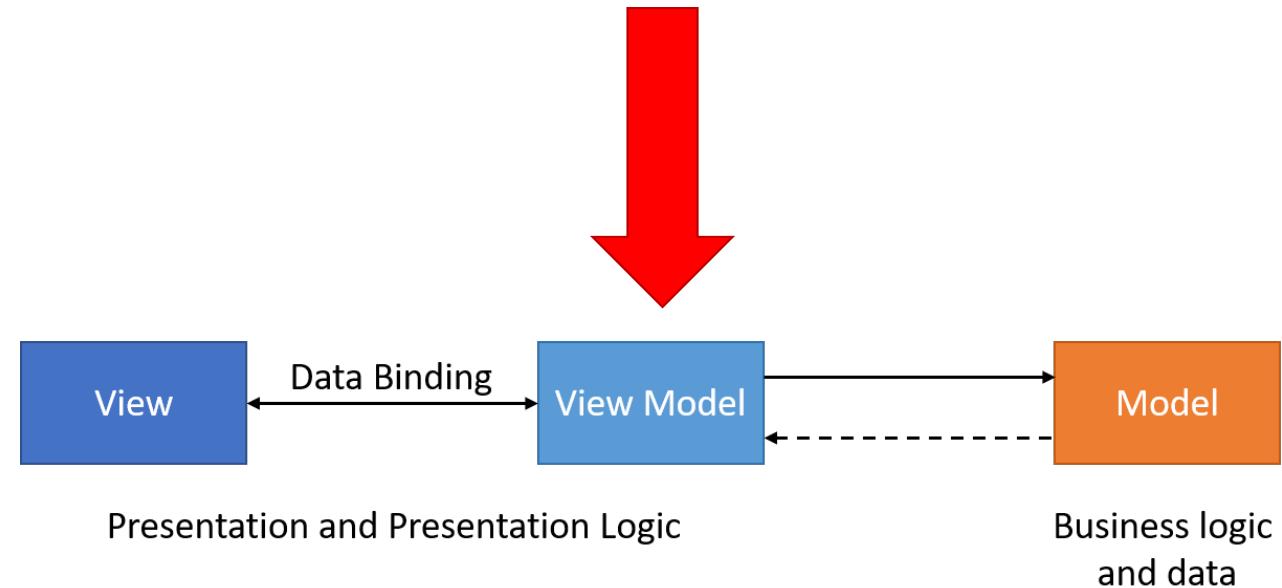
@mallibone

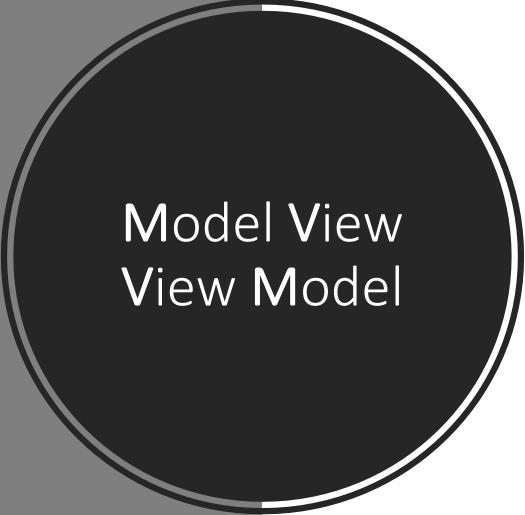




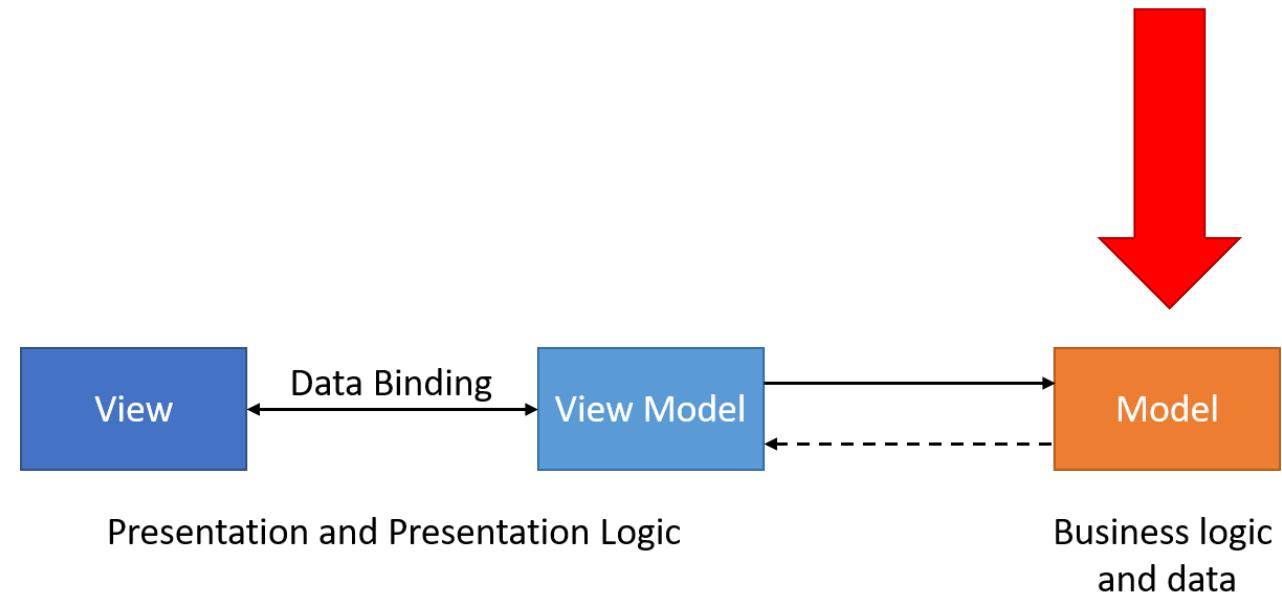


Model View  
View Model





Model View  
View Model

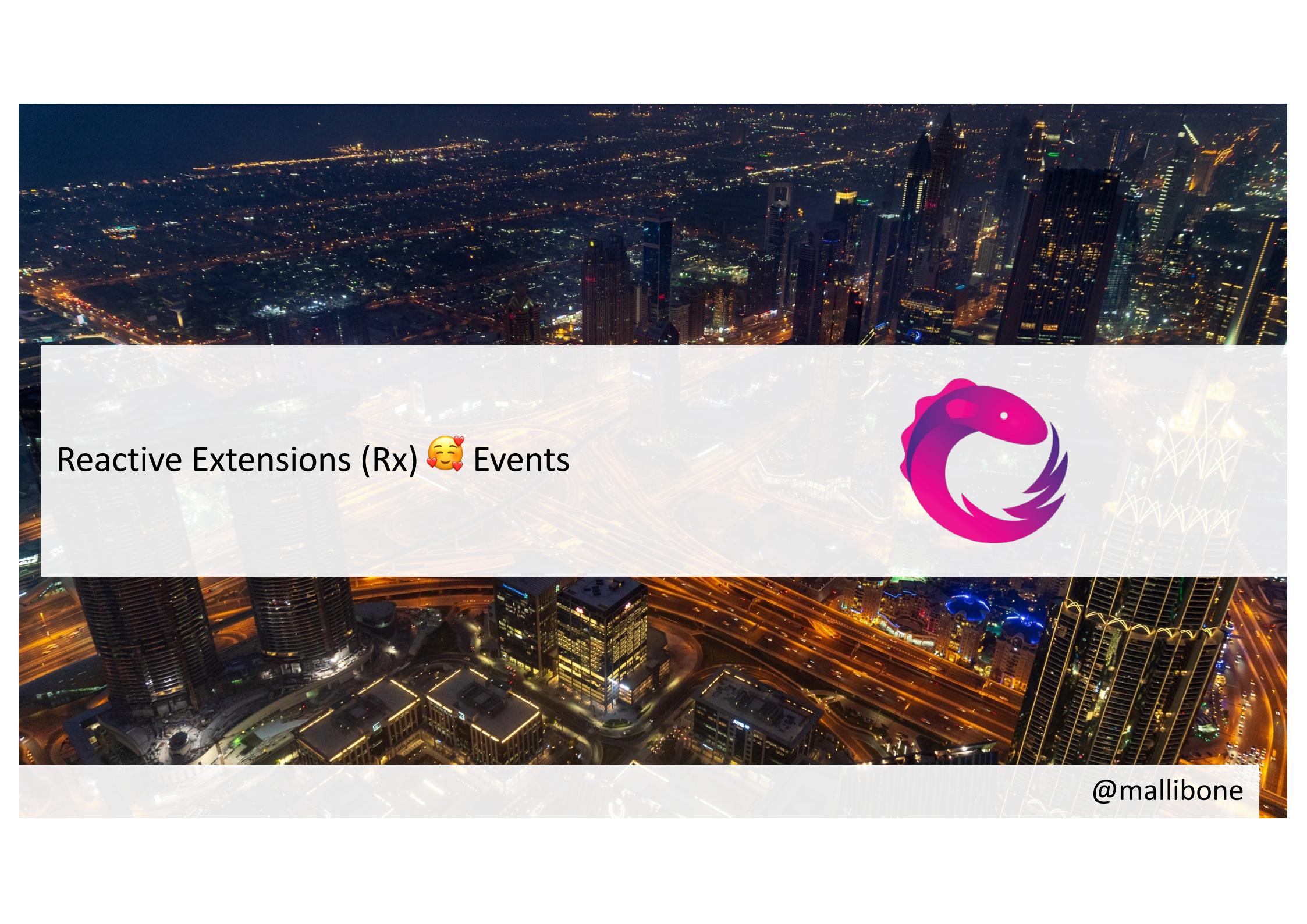


# EVENTS

# EVENTS EVERYWHERE

imgflip.com

@mallibone



Reactive Extensions (Rx) 😍 Events



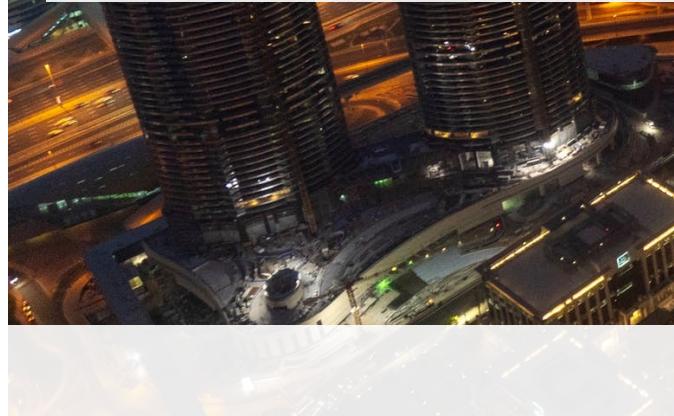
@mallibone



Rx = Observables + LINQ + Schedulers

Events\*

Timers/Threads/Tasks\*



@mallibone

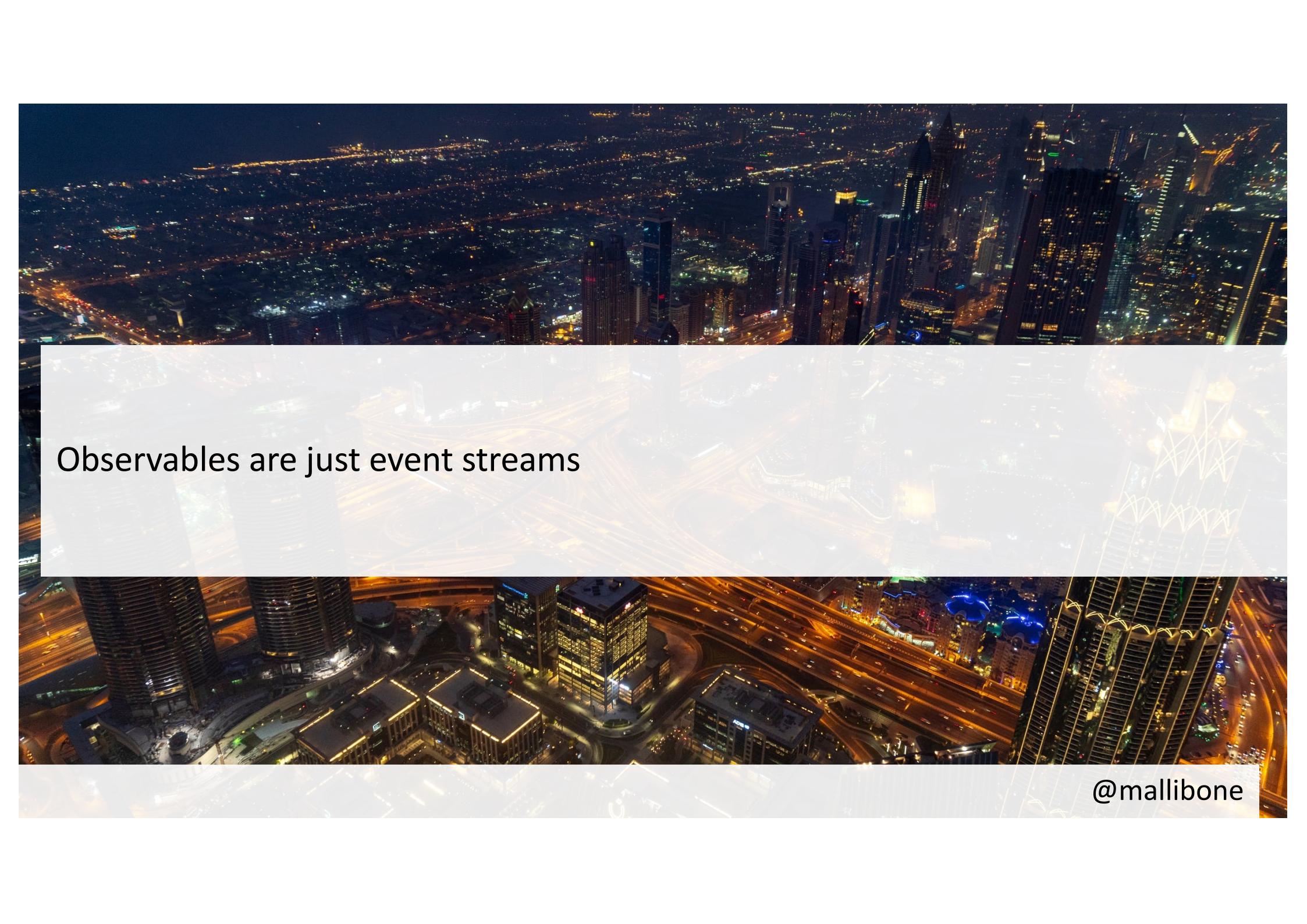
Rx101 – Demo01.cs

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

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

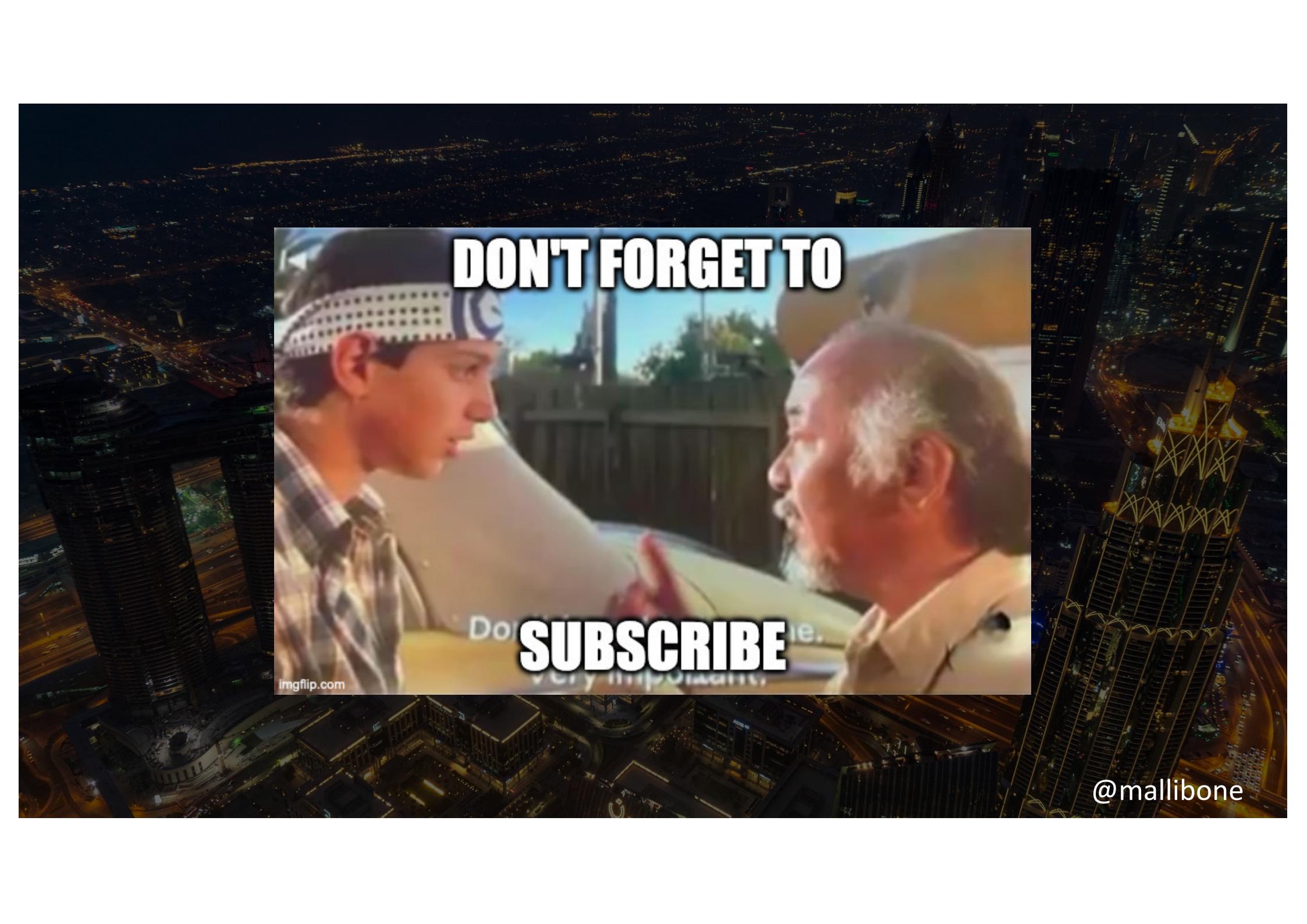
        private static void RunEventSample()
        {
            void EventSampleOnMeasurementChanged(object sender, MeasurementUpdate update) =>
                Console.WriteLine($"Temperature update {update.CurrentMeasurement}");

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



Observables are just event streams

@mallibone



**DON'T FORGET TO**

**SUBSCRIBE**

imgflip.com

@mallibone



```
var observableSample = new ObservableSample();

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

@mallibone



**ONE DOES NOT FORGET TO**

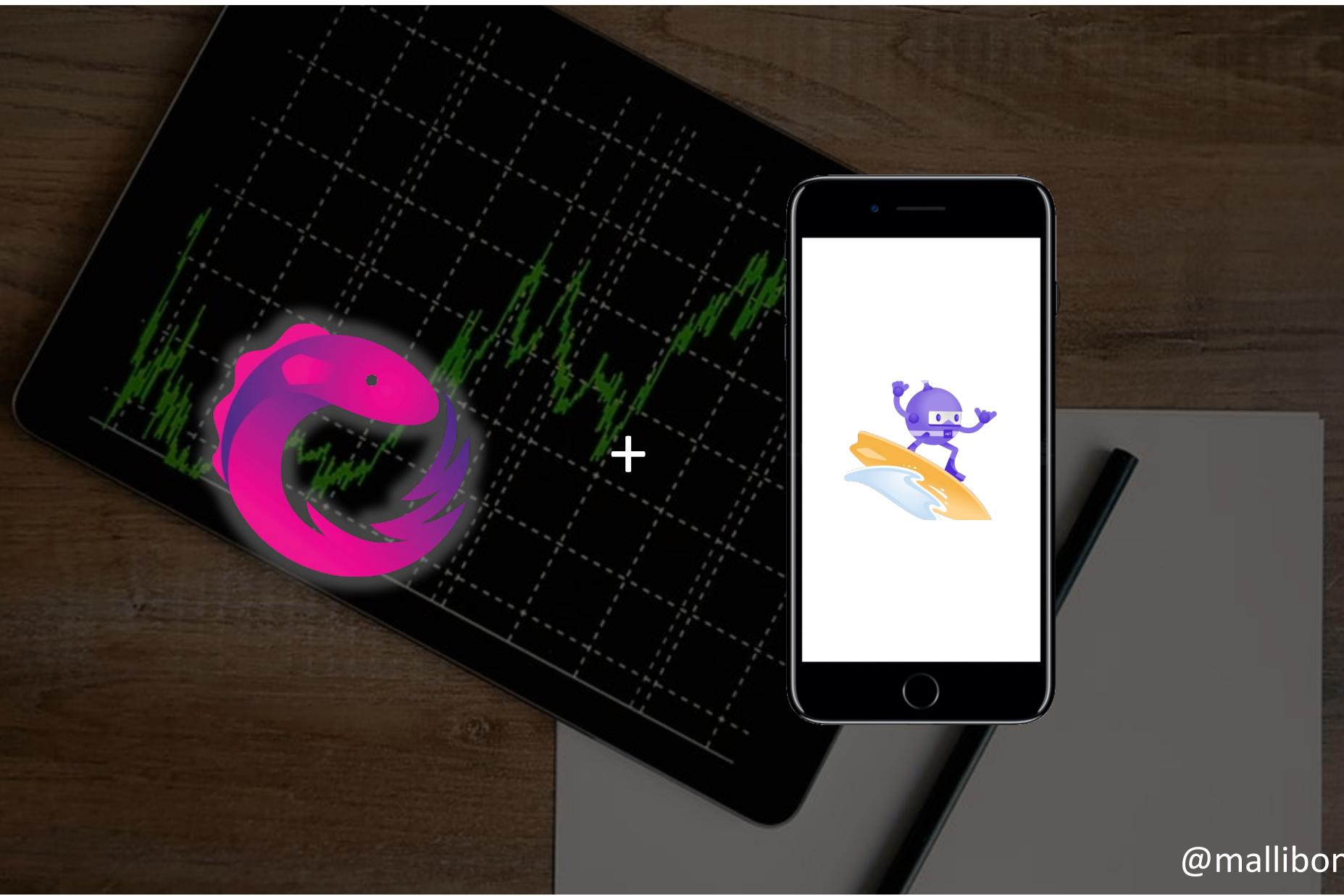
**DISPOSE**

imgflip.com

@mallibone



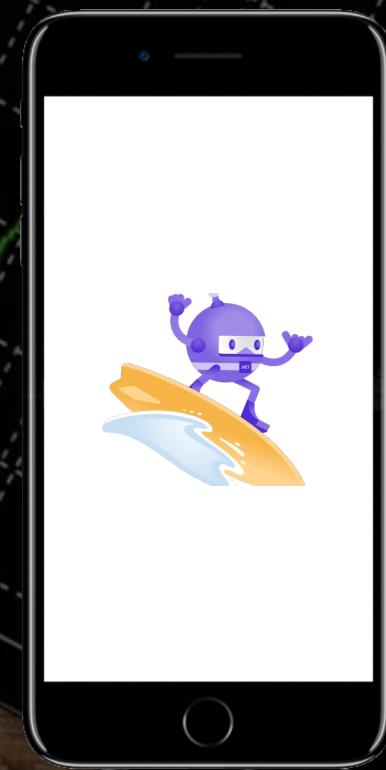
@mallibone



@mallibone



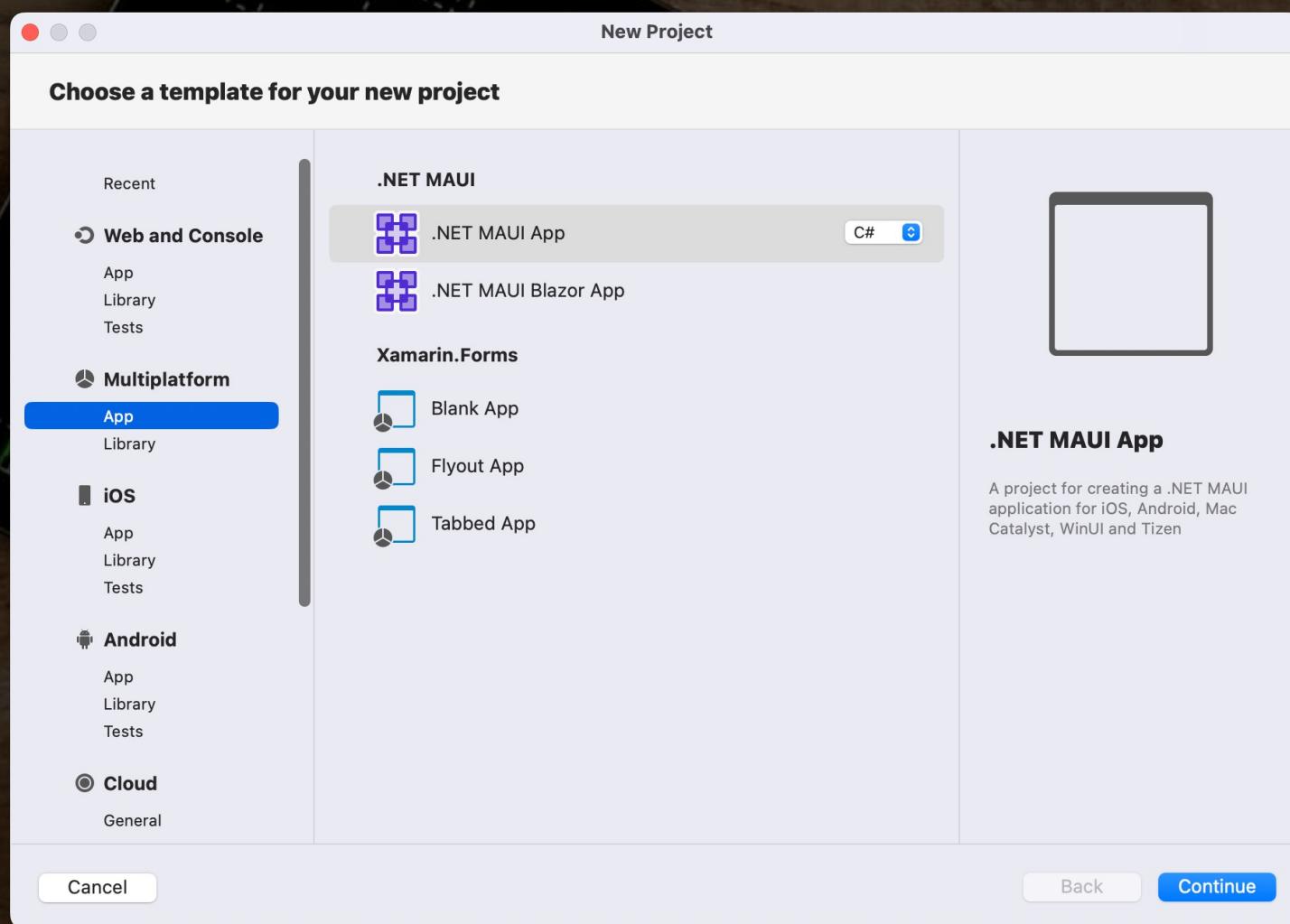
+



=

A large, stylized blue letter 'R' followed by a smaller white 'ü' character.

@mallibone



@mallibone

Debug > My Mac (MacCatalyst)

MainPage.xaml.cs MainPage.xaml

Gnabber

Solution

Gnabber

- Connected Services
- Dependencies
- Platforms
- Properties
- Resources
- App.xaml
- AppShell.xaml
- MainPage.xaml

  - MainPage.xaml.cs

- MauiProgram.cs

<?xml version="1.0" encoding="utf-8" ?>  
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"  
              xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"  
              x:Class="Gnabber.MainPage"  
  
    <ScrollView>  
        <VerticalStackLayout  
            Spacing="25"  
            Padding="30, 0"  
10     </VerticalStackLayout>  
11    </ScrollView>  
12    </ContentPage>

NuGet Packages – Gnabber

Reactive

Browse Installed Updates

ReactiveUI 18.4.20

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.

ReactiveUI.Fody 18.4.20

Fody extension to generate RaisePropertyChanged notifications for properties and ObservableAsPropertyHelper properties.

ReactiveUI.Maui 18.4.1

Contains the ReactiveUI platform specific extensions for Microsoft Maui

ID: ReactiveUI  
Author: .NET Foundation and Contributors  
Published: 01.02.2023  
Downloads:  
Dependencies:  
DynamicData (>= 7.12.11)  
Splat (>= 14.6.8)

Package source: nuget.org

Uninstall Package

Source

Toolbox

Search

Text Snippets

commentb

Layouts

- AbsoluteLayout
- ContentView
- FlexLayout
- Frame
- Grid
- RelativeLayout
- ScrollView
- StackLayout

Controls

- ActivityIndicator
- BoxView

Properties Document Outline

Property Filter

Misc

File Format: Visual Studio 2012+  
File Path: /Users/malli...bber.csproj  
Name: Gnabber  
Root Directory: /Users/malli...er/Gnabber

Ln 1, Col 1 Spaces LF

3 packages successfully added

Errors Build Output Tasks Vim Output Package Console

@mamijone

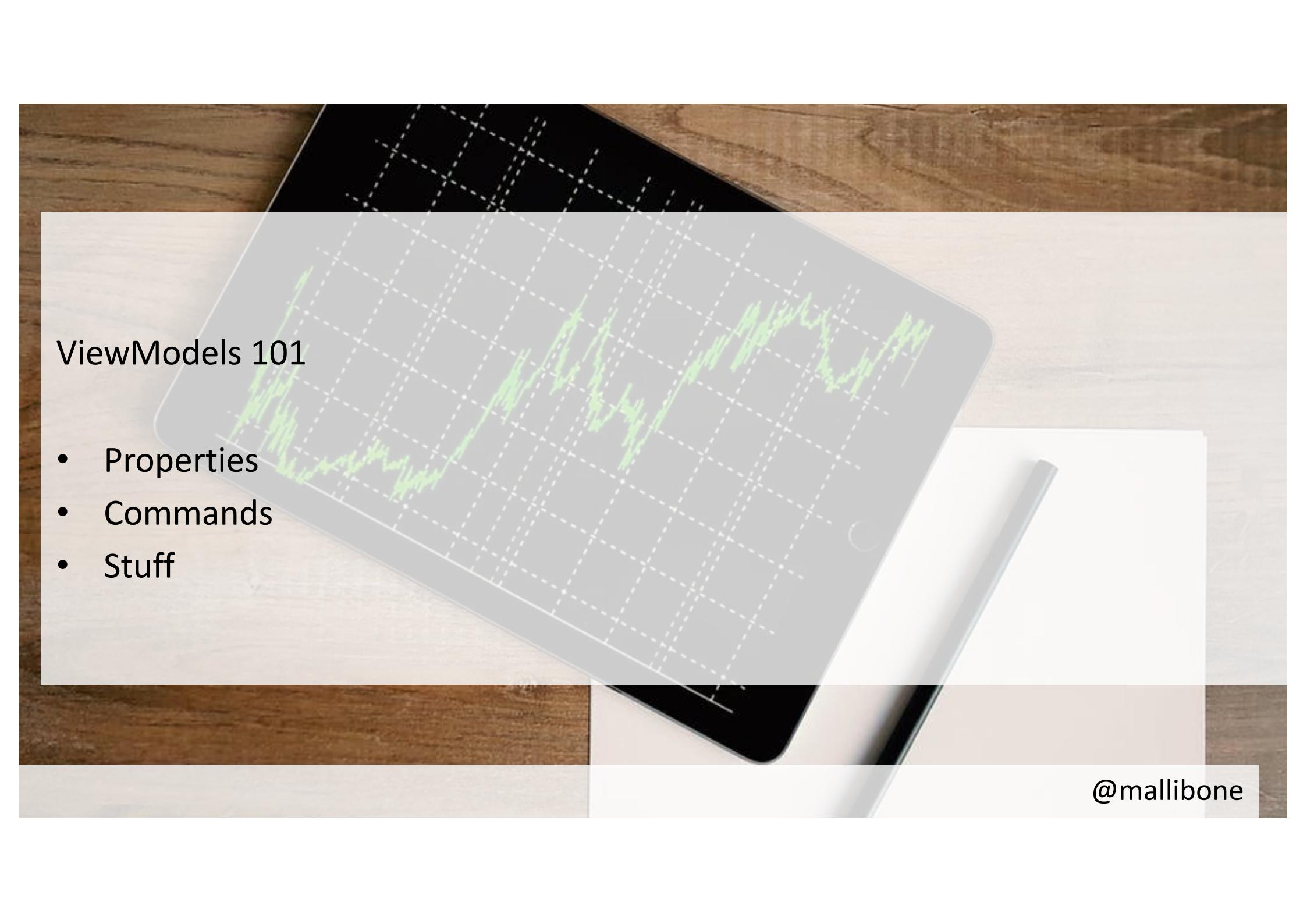
Rx101 – Demo01.cs

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

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

        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;
        }
    }
}
```



## ViewModels 101

- Properties
- Commands
- Stuff

@mallibone



## ViewModels 101

- **Properties**
- Commands
- Stuff

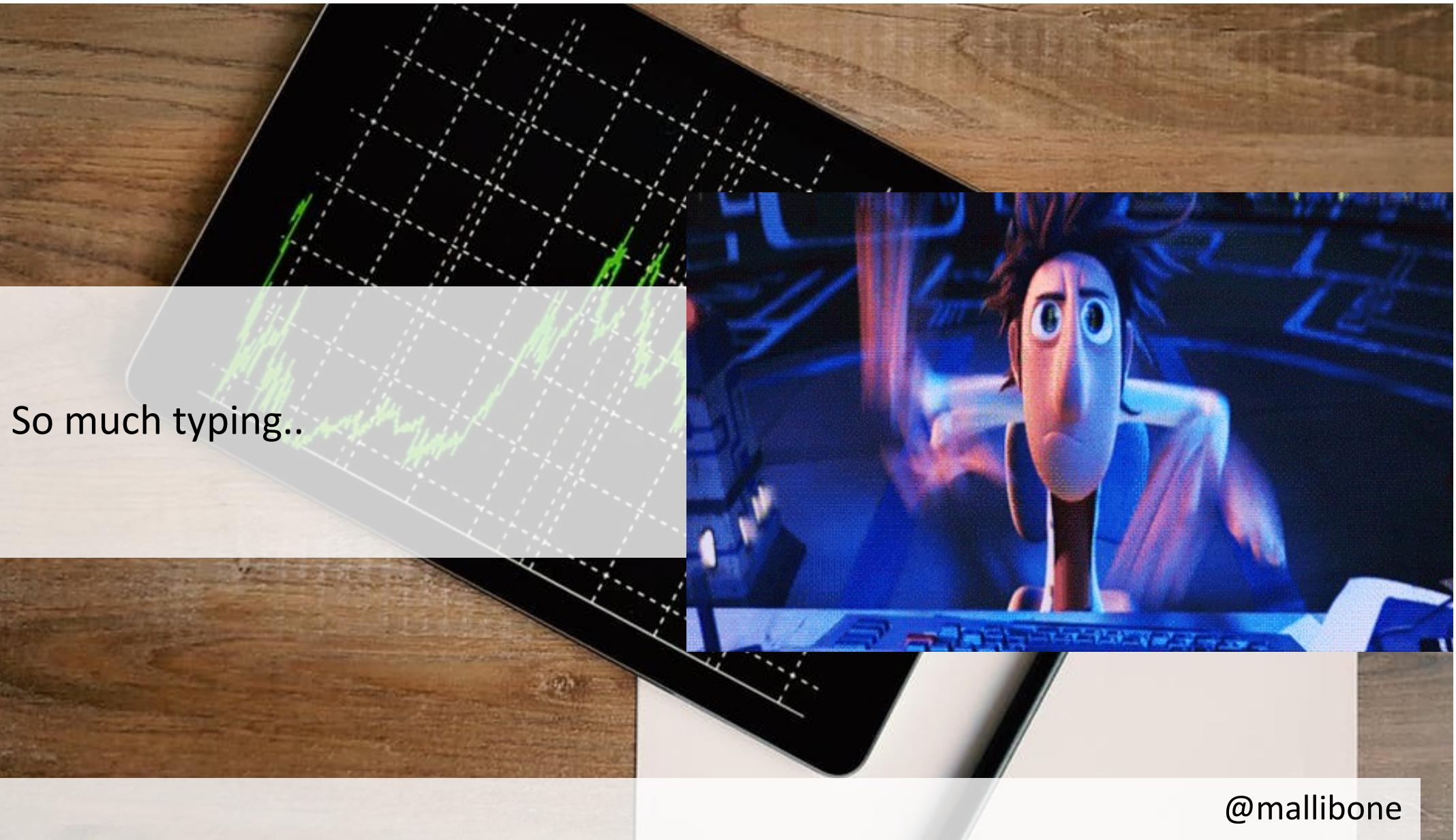
@mallibone



```
private int _searchEntry;

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

@mallibone



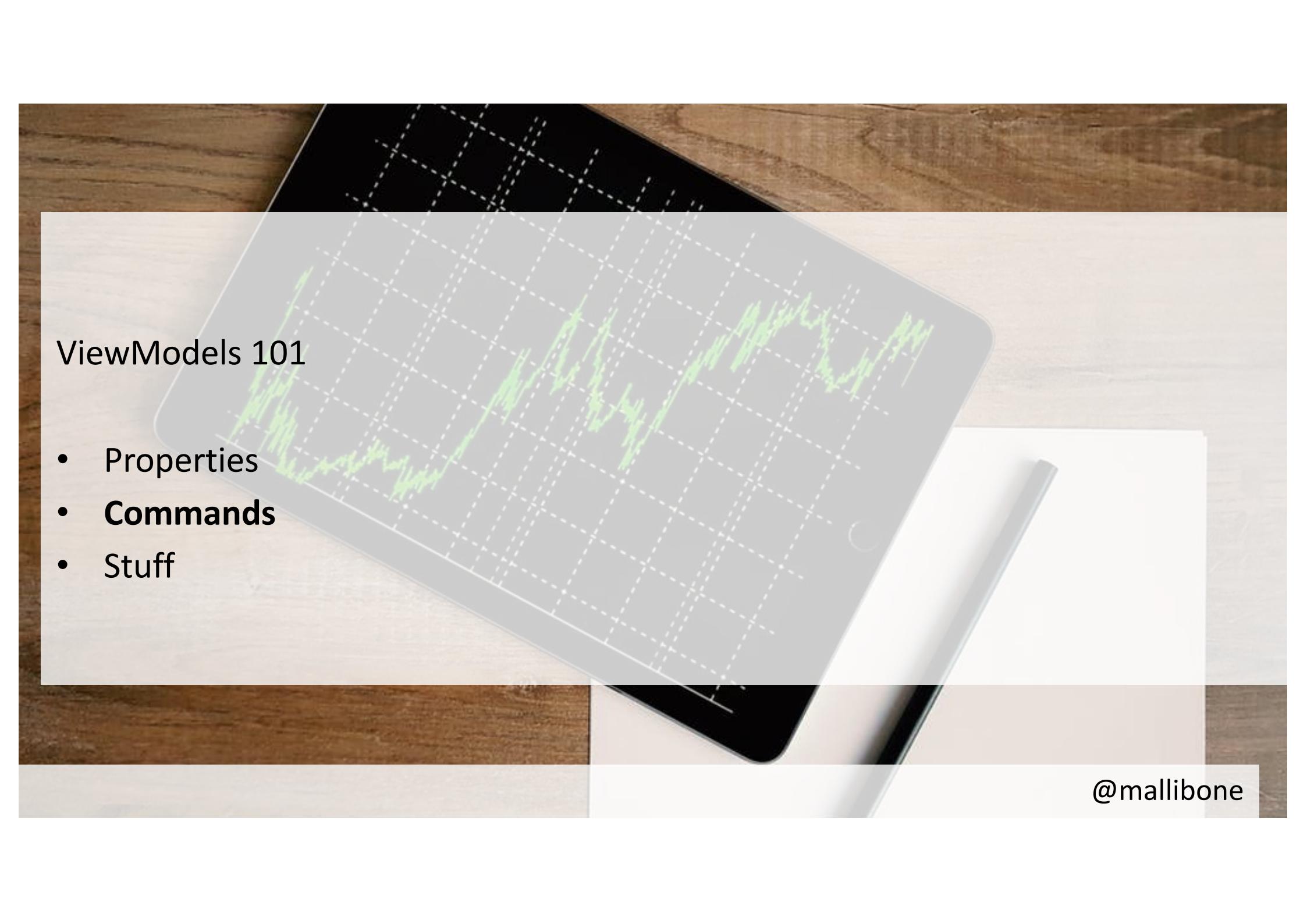
So much typing..

@mallibone



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

@mallibone



## ViewModels 101

- Properties
- **Commands**
- Stuff

@mallibone

## Synchronous Commands

```
// Your standard ICommand
```

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

```
// Can be subscribed to
```

```
ReactiveCommand<Unit, string> syncCommand = ReactiveCommand.Create(() => "Sync");
```

@mallibone

## Synchronous Commands



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

@mallibone

## Synchronous Commands



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

@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

# 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};

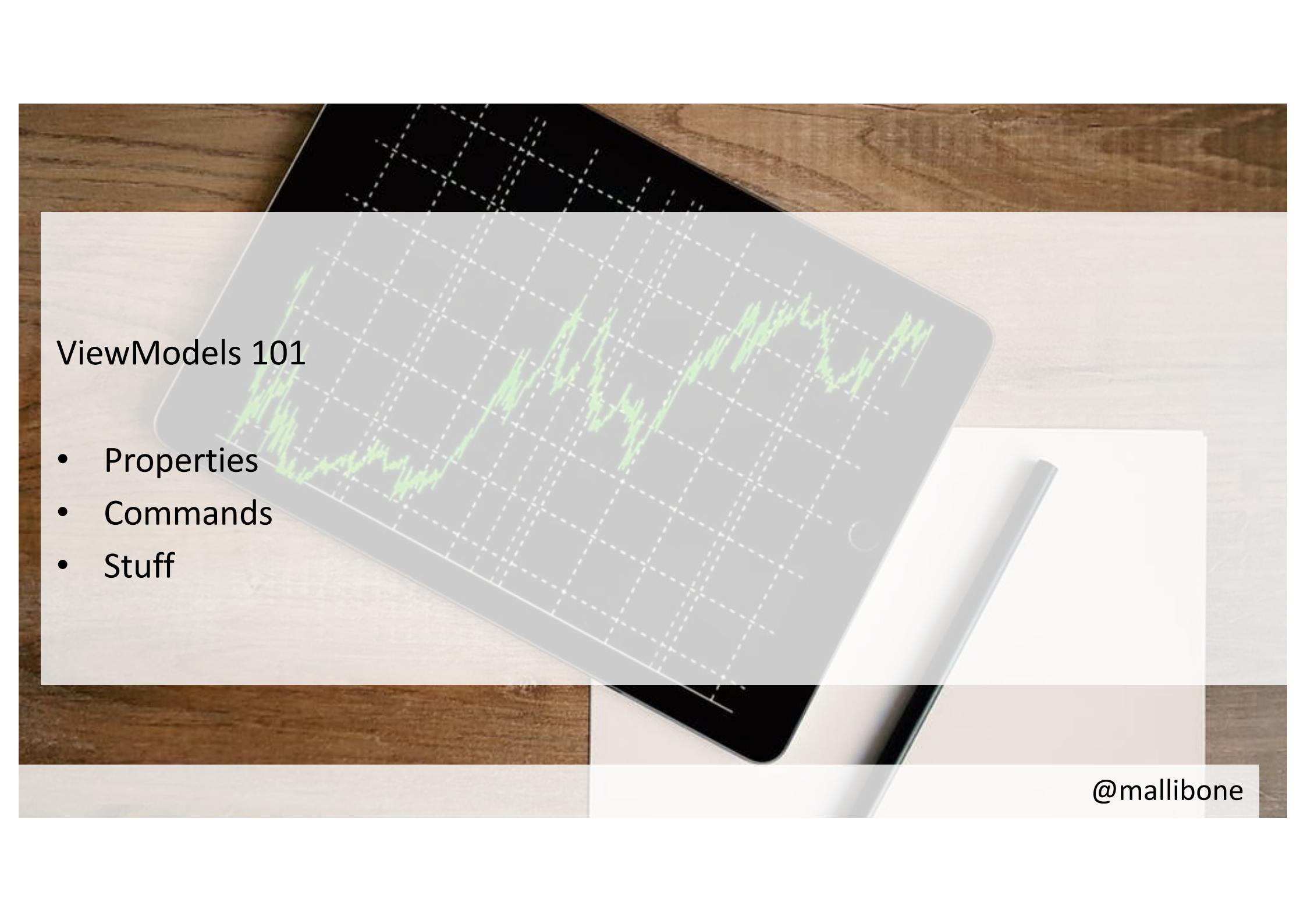
// Combined Commands
CombinedReactiveCommand<Unit, string> combinedCommand = ReactiveCommand.CreateCombined(commands);
```

## Combined Commands

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

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

@mallibone



## ViewModels 101

- Properties
- Commands
- Stuff

@mallibone



@mallibone



= Rx + MVVM + “a ton of helpers™”

Dynamic Collections

Validation

Navigation

Event Extensions

Auto Persistence

View Lifecycle

DI Container

User Confirmation

@mallibone



## Criteria for Mobile Apps

@mallibone

## Criteria for awesome Mobile Apps

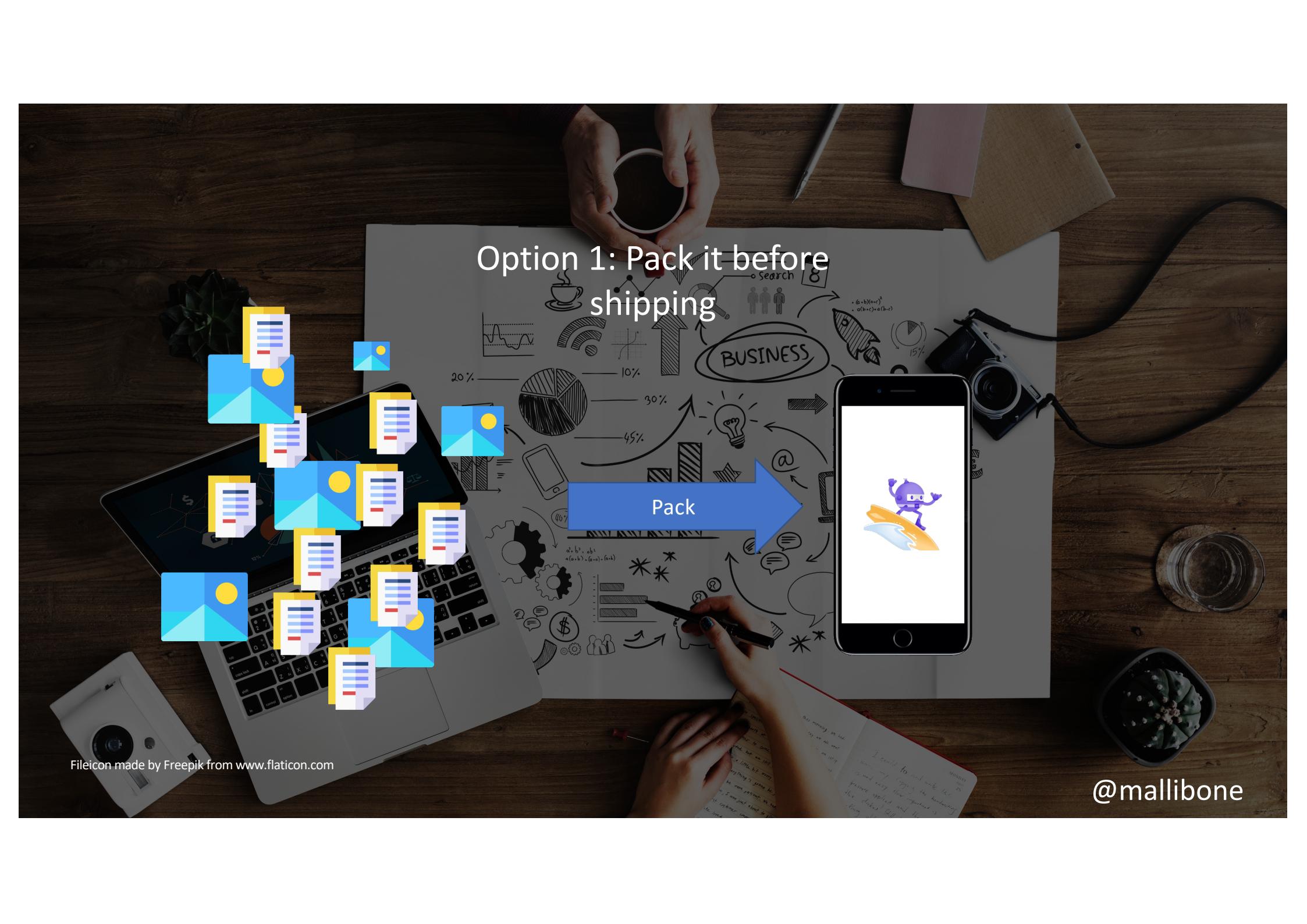
- Don't Crash
- Responsive
- Work Offline 🤯
- 10'000 other points depending on preference etc.  
(like design and fluff)

@mallibone



# How to build an App that works offline?

@mallibone



Option 1: Pack it before  
shipping.

Pack

Fileicon made by Freepik from [www.flaticon.com](http://www.flaticon.com)

@mallibone

**COULD YOU SHOW ME THE CURRENT WEATHER**

**THAT WOULD BE REALLY GREAT**

imgflip.com

@mallibone

## Option 2: Cache the Calls

Request / Response

Servericon made by prettycons from [www.flaticon.com](http://www.flaticon.com)

@mallibone

## Option 2: Cache the Calls

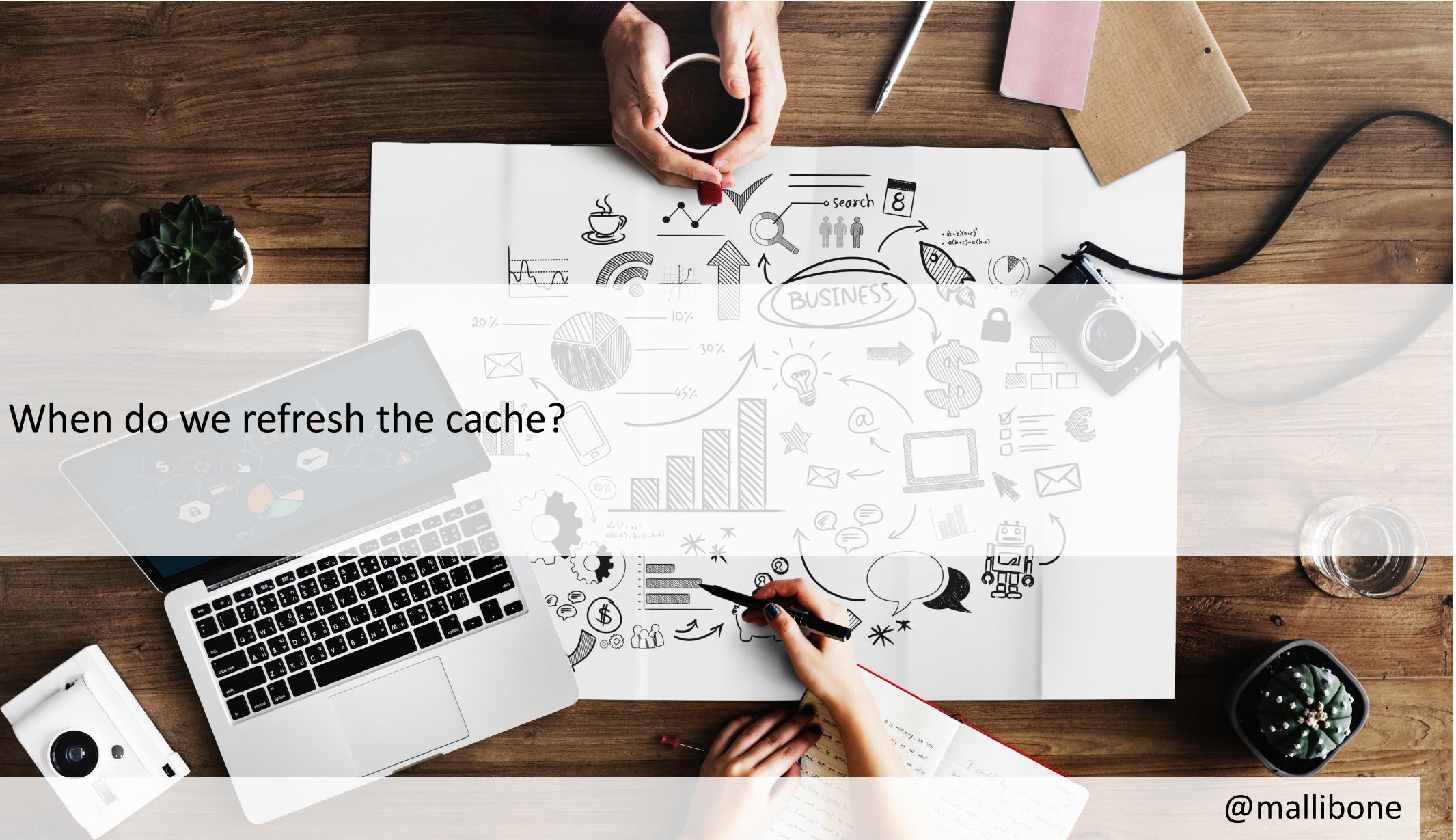
Request /  
Response

Cache

Request /  
Response

Servericon made by prettycons from [www.flaticon.com](http://www.flaticon.com)

@mallibone

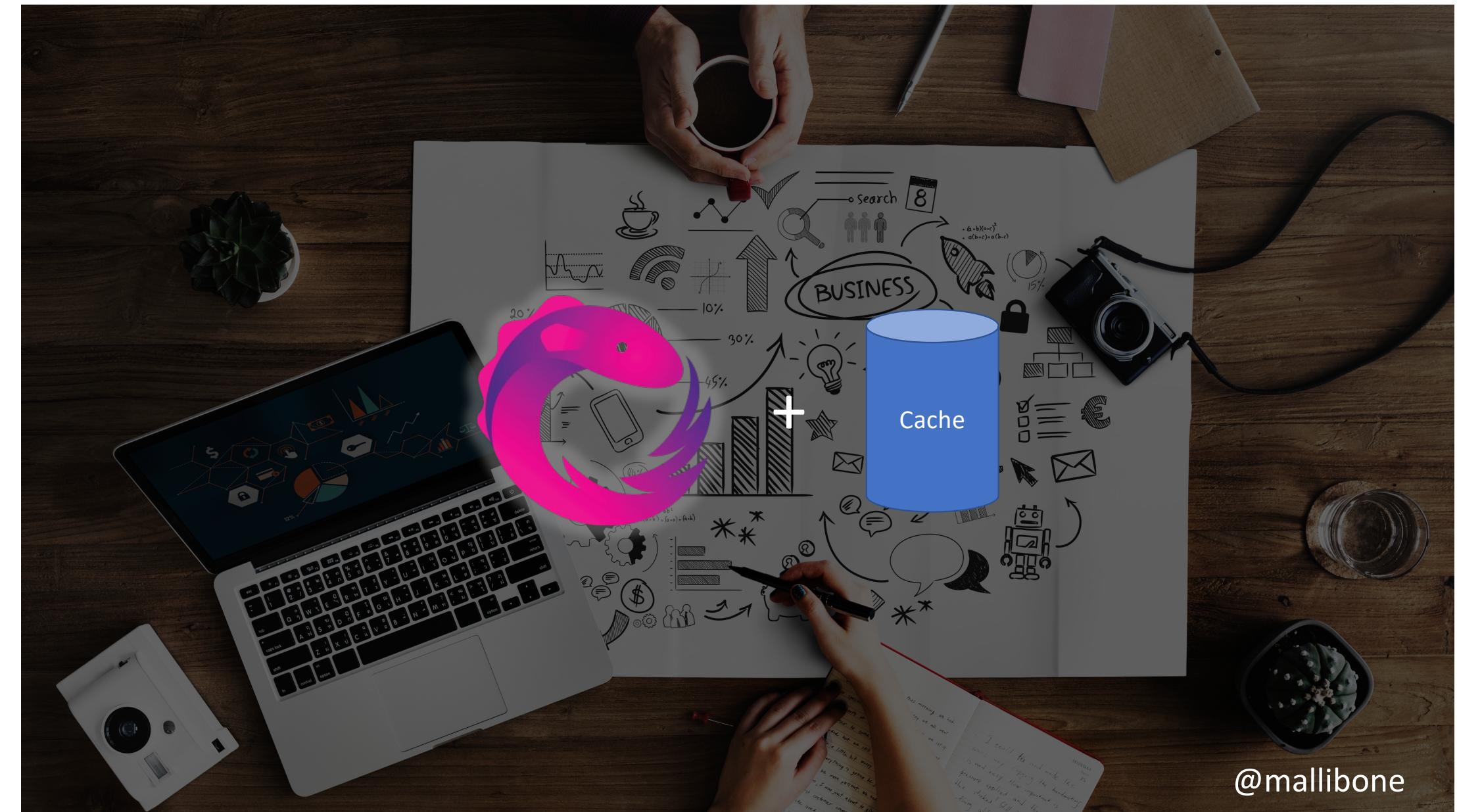


When do we refresh the cache?

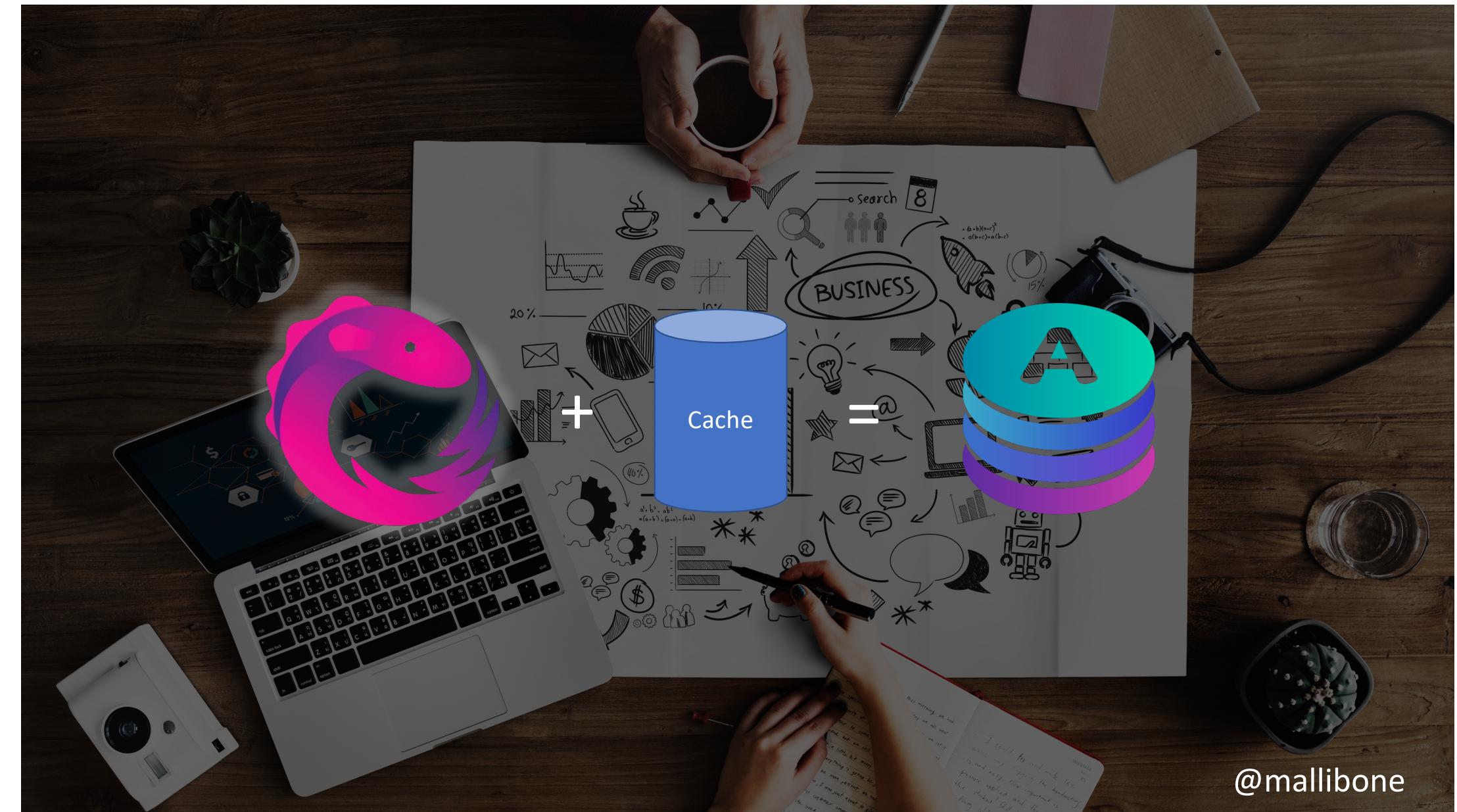
@mallibone



@mallibone



@mallibone



@mallibone

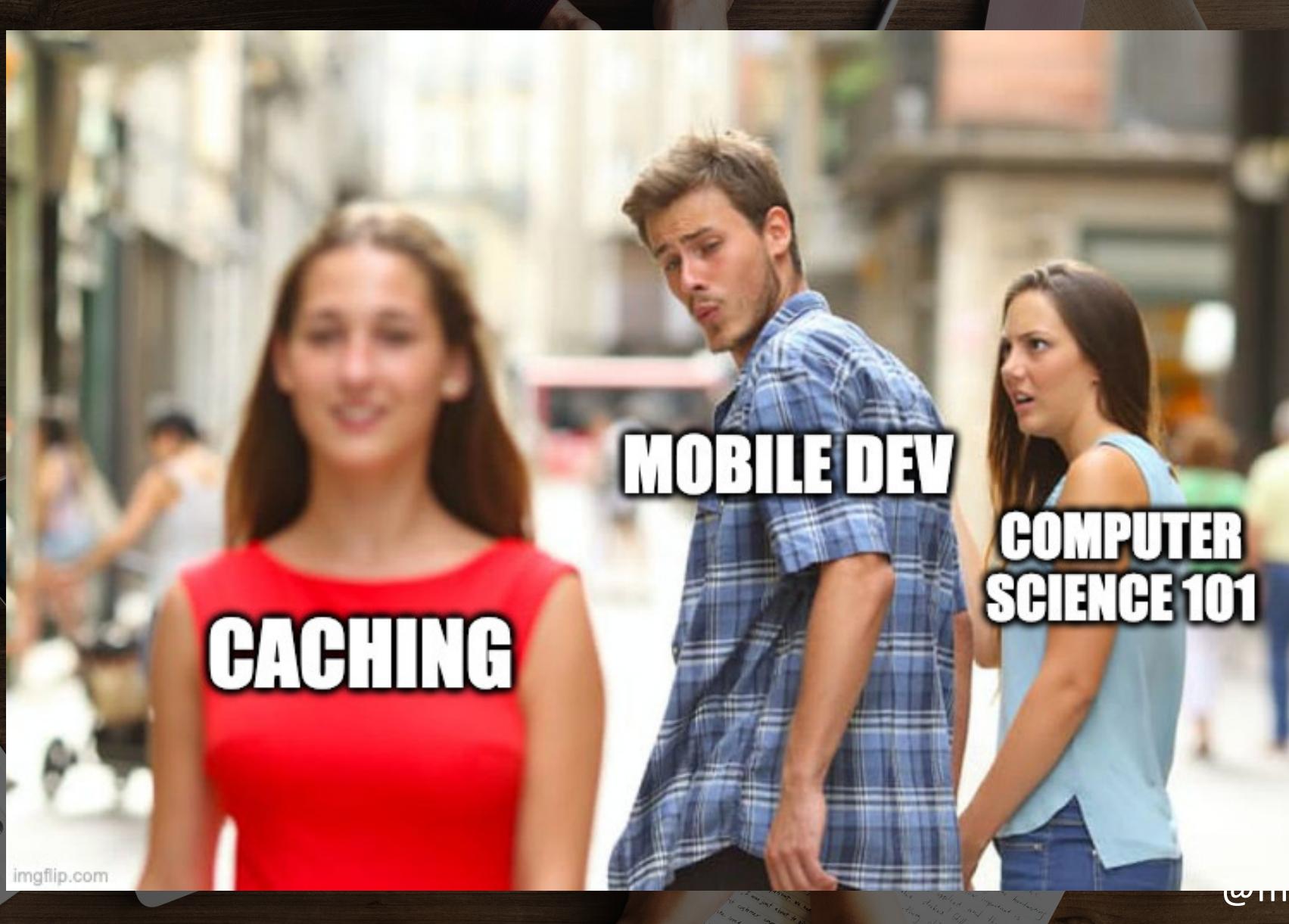
Rx101 – Demo01.cs

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

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

        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;
        }
    }
}
```



imgflip.com

@mallibone



@mallibone

## Recap

- MVVM Properties and Commands
- 🔎 Debounce, Filter and Map with Rx
- Multithreading
- Observable vs Task
- Seamless Offline experience with Akavache

@mallibone

# Takeaways



# Takeaways

- Reactive Extensions (Rx)
  - Observables
  - LINQ
  - Schedulers
- Rx + UI → Reactive UI
- Caching with Akavache
- Don't forget to Subscribe and Dispose



*Thank you for your time!*



Mark Allibone



@mallibone



Rey Technology



<https://mallibone.com>



<https://nullpointers.io>



<http://reactivex.io/>



<https://www.reactiveui.net/>

