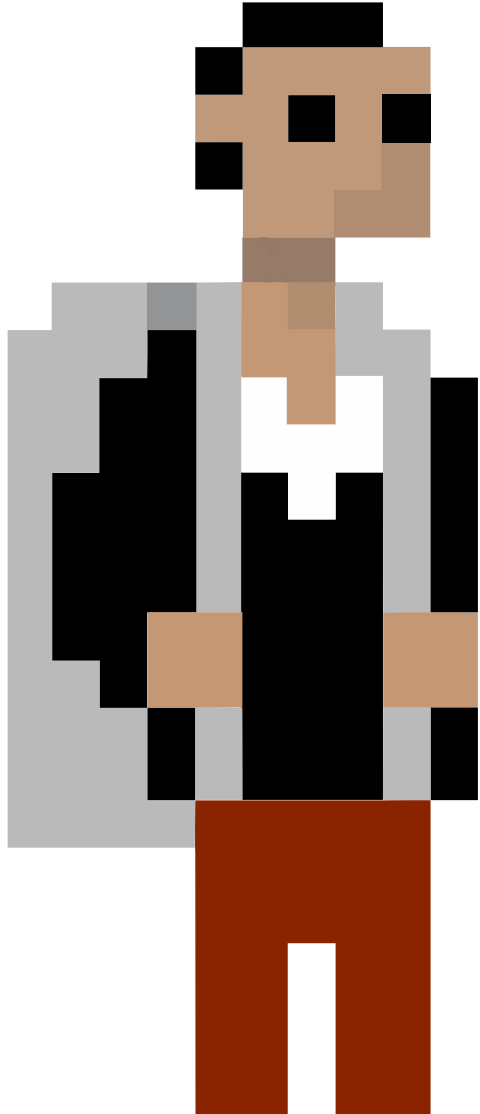
A man with a beard and glasses, wearing a blue shirt, is seated in the background, slightly out of focus. In the foreground, there is a large number of brown stuffed monkeys, with one in sharp focus in the center. The scene is indoors, with a window and some foliage visible in the background.

# Motz Codes Live

@JamesMontemagno  
motzcod.es

Pull to Refresh in Xamarin.Forms  
September 11, 2015

# Who is Motz?



James  
Montemagno  
Developer Evangelist, Xamarin

---

[james@xamarin.com](mailto:james@xamarin.com)

[motzcod.es](http://motzcod.es)

[@JamesMontemagno](https://twitter.com/JamesMontemagno)

Refresh that data!

```
public class MyViewModel : INotifyPropertyChanged
{
    bool isBusy;
    public bool IsBusy
    {
        get { return isBusy; }
        set
        {
            if (isBusy == value)
                return;

            isBusy = value;
            OnPropertyChanged ("IsBusy");
        }
    }
}

public event PropertyChangedEventHandler PropertyChanged;

public void OnPropertyChanged(string propertyName)
{
    if (PropertyChanged == null)
        return;

    PropertyChanged (this, new PropertyChangedEventArgs (propertyName));
}
}
```

```
ICommand refreshCommand;  
public ICommand RefreshCommand  
{  
    get  
    { return refreshCommand ??  
        (refreshCommand =  
            new Command (async () => await ExecuteRefreshCommand()));  
    }  
}
```

```
async Task ExecuteRefreshCommand()  
{  
    if (IsBusy)  
        return;  
  
    IsBusy = true;  
    //do stuff  
    IsBusy = false;  
}
```

A long time ago...

In early releases of Xamarin.Forms

No pull to refresh ☹️

```
var refresh = new ToolbarItem
{
    Icon = "refresh.png",
    Text = "Refresh",
    Command = viewModel.RefreshCommand
};

ToolbarItems.Add(refresh);
```





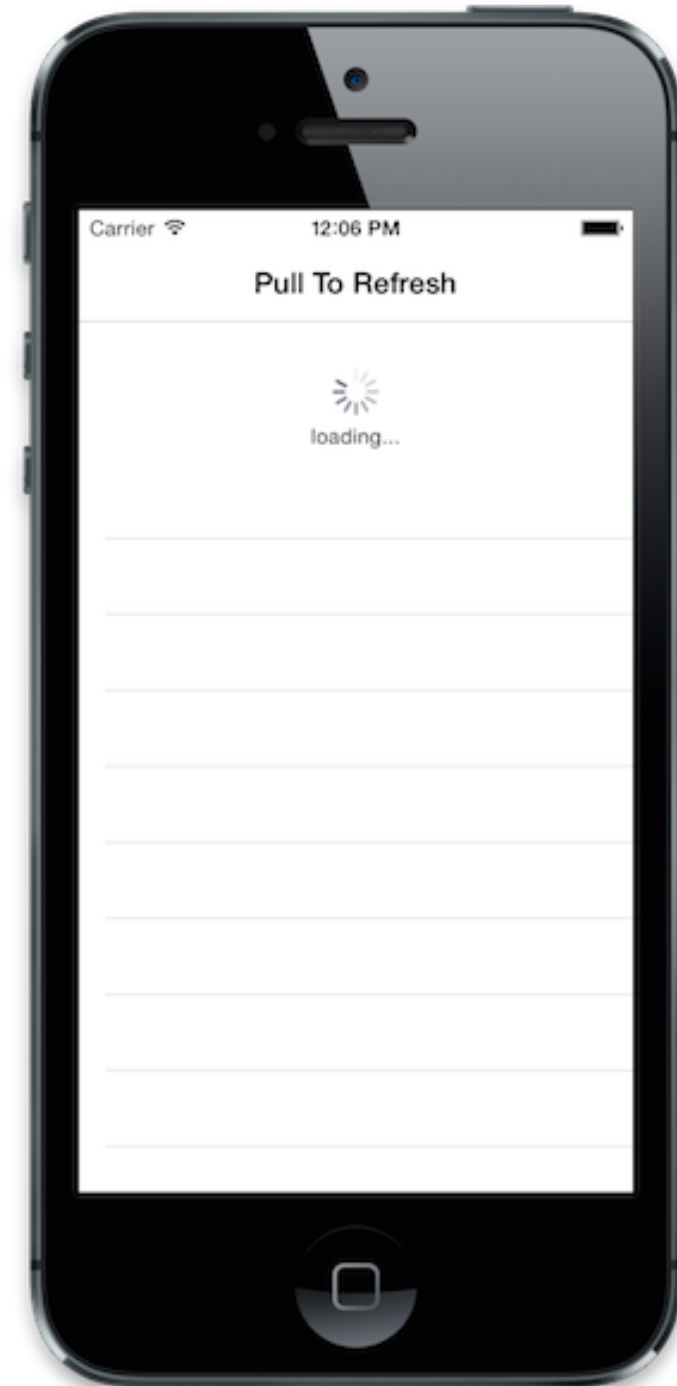
We can fix it!

We can custom render it!

# iOS PullToRefreshListView

- Adds UIRefreshControl
- Works with ListViewRenderer
- UIRefreshControl is a SubView of the UITableView

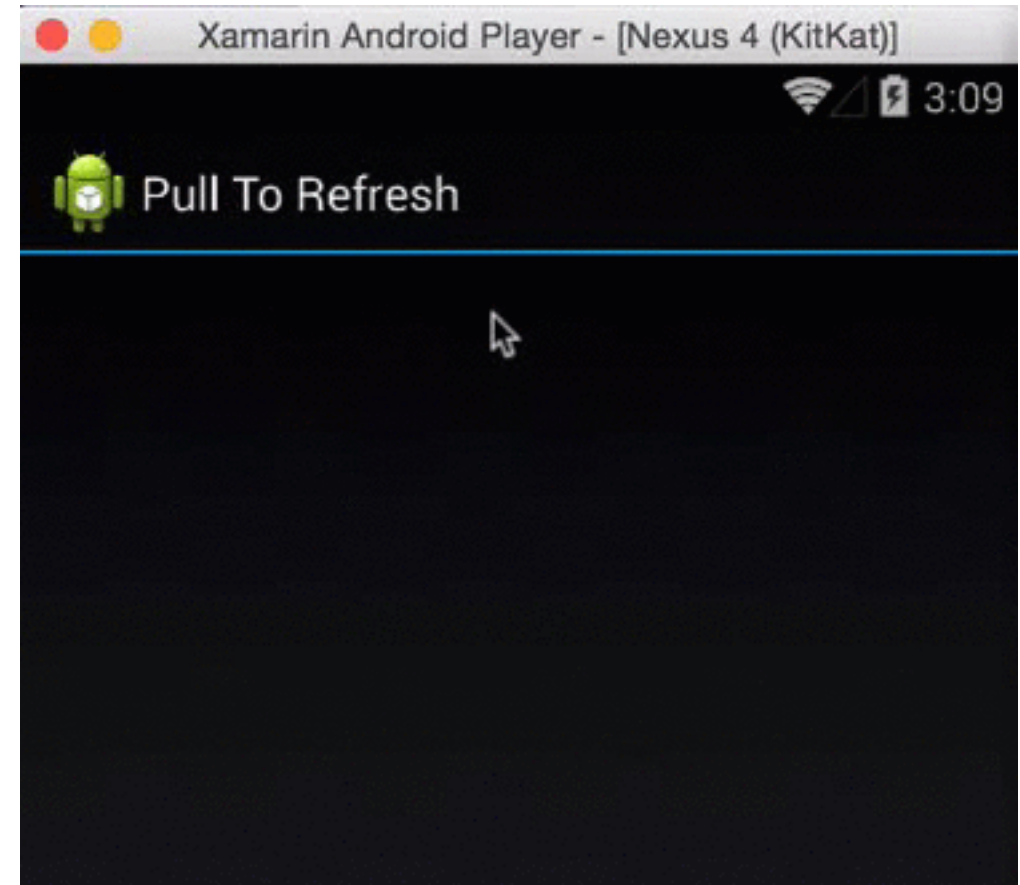
<http://motzcod.es/post/87917979362/pull-to-refresh-for-xamarinforms-ios>



# Android PullToRefreshContentView

- Adds SwipeRefreshLayout
- Based on ContentView
- Uses IVisualElementRenderer
- SwipeRefreshLayout is the base layout and bundles sub-views (opposite of iOS)

<http://motzcod.es/post/103224921992/pullswipe-to-refresh-for-xamarinforms-android>



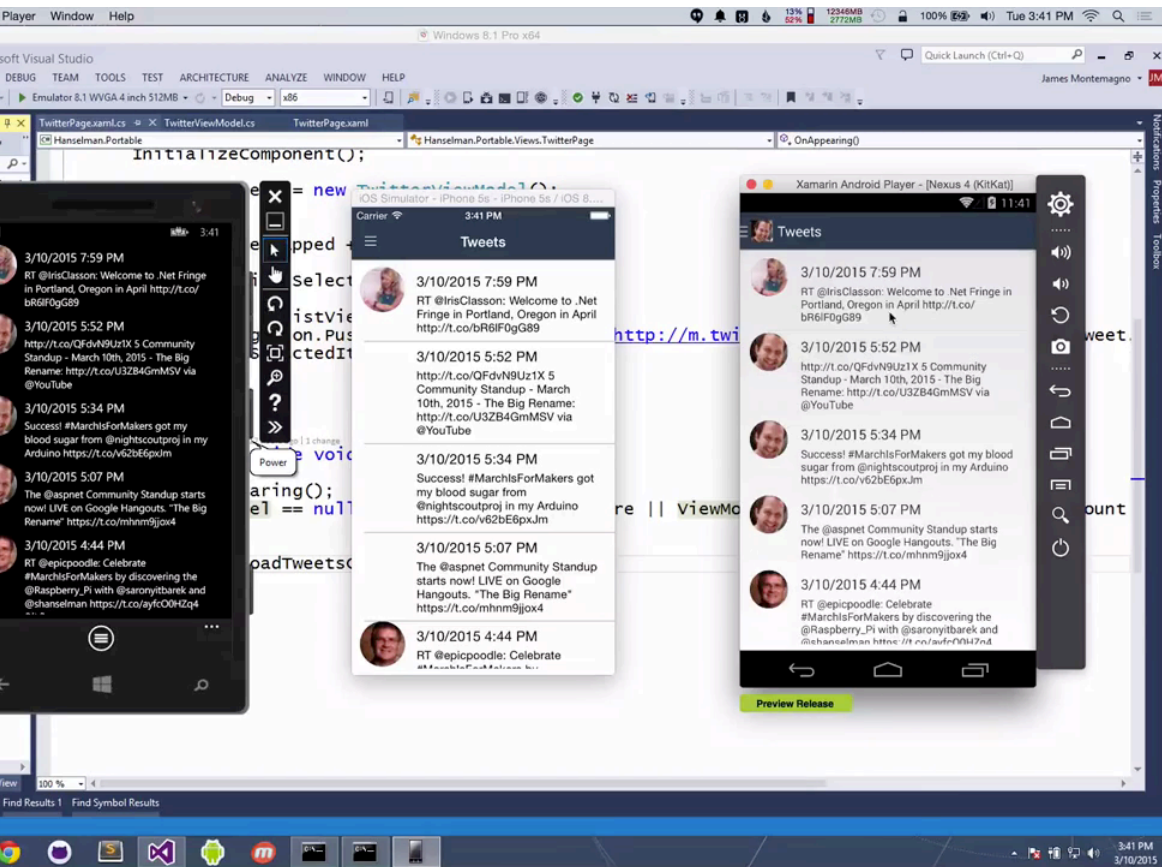
Only good for  
Xamarin.Forms 1.0.X – 1.3.X!

Don't use with 1.4.X!

Game Changer:  
Xamarin.Forms 1.4.X

# Pull to Refresh in 1.4.X

- Built in ListView
- Uses Native Controls
- Supports iOS, Android, and Windows Phone!



# The API

```
//Enable/Disable all pull to refresh  
public bool IsPullToRefreshEnabled { get; set; } = false;
```

```
//Is the spinner currently shown  
public bool IsRefreshing { get; set; } = false;
```

```
//The method/command to trigger when pulled  
public ICommand RefreshCommand { get; set; } = null;
```

```
//Manual Events to trigger/subscribe  
public void BeginRefresh ();  
public void EndRefresh ();  
public event EventHandler Refreshing;
```

# The XAML

```
<ListView x:Name="listview"  
    ItemsSource="{Binding Items}"  
    HasUnevenRows="True"  
    IsPullToRefreshEnabled="True"  
    RefreshCommand="{Binding RefreshCommand}"  
    IsRefreshing="{Binding IsBusy, Mode=OneWay}">
```



Let's take a look!

What about other views?  
ScrollView?

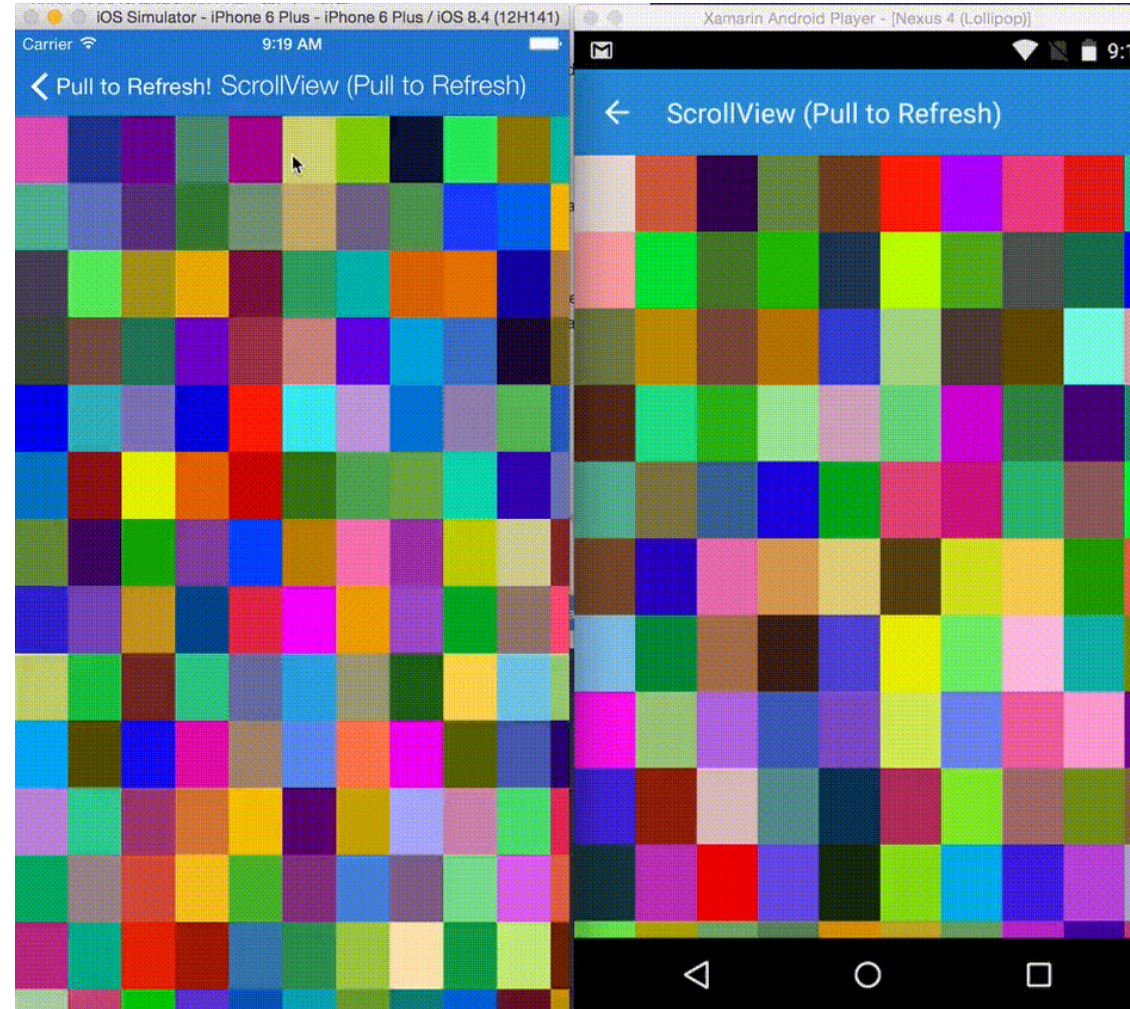
We can fix it!

We can custom render it!

# PullToRefreshLayout

- Supports
  - iOS & Android
- Works with
  - ListView & ScrollView
- Uses Native Controls
  - UIRefreshControl & SwipeRefreshLayout
- Based on ContentView

<http://motzcod.es/post/128274430137/pull-to-refresh-anyish-xamarinforms-view>



# The API

//Enable/Disable all pull to refresh

```
public bool IsPullToRefreshEnabled { get; set; } = true;
```

//Is the spinner currently shown

```
public bool IsRefreshing { get; set; } = false;
```

//The method/command to trigger when pulled

```
public ICommand RefreshCommand { get; set; } = null;
```

//Spinner color

```
public Color RefreshColor { get; set; } = Color.Default;
```

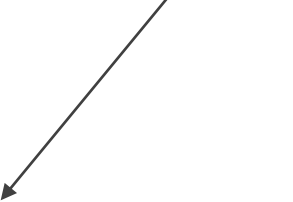
//Spinner background color

```
public Color RefreshBackgroundColor { get; set; } = Color.Default;
```

# The XAML

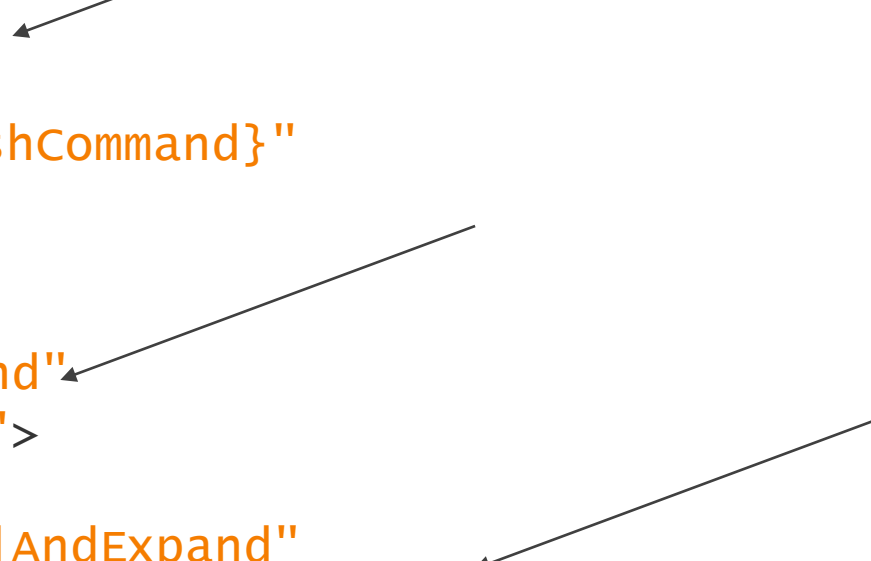
```
<?xml version="1.0" encoding="UTF-8"?>
<ContentPage
  xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  x:Class="RefreshSample.Views.ScrollableViewXamlPage"
  xmlns:controls="clr-namespace:Refractored.XamForms.PullToRefresh;assembly=Refractored.XamForms.PullToRefresh"
  Title="Xaml Scroll">

</ContentPage>
```



# The XAML

```
<controls:PullToRefreshLayout
    IsPullToRefreshEnabled="True"
    RefreshCommand="{Binding RefreshCommand}"
    IsRefreshing="{Binding IsBusy}"
    RefreshColor="Blue">
    <ScrollView
        HorizontalOptions="FillAndExpand"
        VerticalOptions="FillAndExpand">
        <StackLayout
            HorizontalOptions="FillAndExpand"
            VerticalOptions="FillAndExpand">
            <BoxView HeightRequest="200" WidthRequest="100" BackgroundColor="Blue"/>
            <BoxView HeightRequest="200" WidthRequest="100" BackgroundColor="Red"/>
            <BoxView HeightRequest="200" WidthRequest="100" BackgroundColor="Yellow"/>
            <BoxView HeightRequest="200" WidthRequest="100" BackgroundColor="Purple"/>
            <BoxView HeightRequest="200" WidthRequest="100" BackgroundColor="Maroon"/>
        </StackLayout>
    </ScrollView>
</controls:PullToRefreshLayout>
```



The diagram illustrates the hierarchical structure of the XAML code. Three arrows point to the opening tags of the nested elements: the first arrow points to `<controls:PullToRefreshLayout`, the second arrow points to `<ScrollView`, and the third arrow points to `<StackLayout`. This visualizes how these elements are nested within each other in the UI tree.

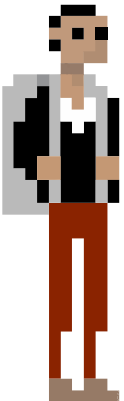
Let's take a look!



# Resources

- Pull to Refresh in Xamarin.Forms ListView
  - <http://motzcod.es/post/113280718807/official-pull-to-refresh-in-xamarinforms-140>
- PullToRefreshLayout
  - <https://github.com/jamesmontemagno/Xamarin.Forms-Awesome-Controls>
  - <https://www.nuget.org/packages/Refractored.XamForms.PullToRefresh/>
- Motz Codes Live:
  - <https://www.youtube.com/user/jamesmontemagno>
  - <https://github.com/jamesmontemagno/MotzCodesLive>

# Thank you. Questions?



James  
Montemagno  
Developer Evangelist, Xamarin

---

[james@xamarin.com](mailto:james@xamarin.com)

[motzcod.es](http://motzcod.es)

[@JamesMontemagno](https://twitter.com/JamesMontemagno)