# Navigation in Xamarin

When there is a need to have multiple pages within a Xamarin Forms app the NavigationPage class can be used to add hierarchical navigation (https://docs.microsoft.com/en-us/xamarin/xamarin-forms/app-fundamentals/navigation/hierarchical).

To use a Navigation Page, assign MainPage to a new instance of Navigation Page in the App constructor. This makes that page the “Root” or first page of the Navigation Stack.

MainPage = new NavigationPage( new SomeContentPage());

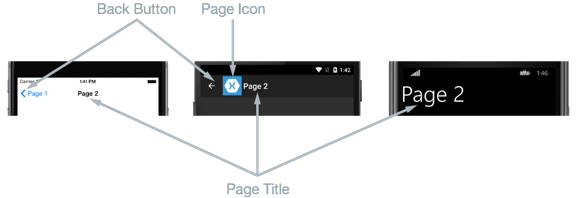
The argument to the constructor is the first page that will go onto the stack. In this case it is a class that inherits from ContentPage called SomeContentPage()

Once the navigation page is active, the pages displayed within it can call various methods to access the navigation stack. Like a stack data structure, the navigation stack uses pushes and pops to add and remove pages from the stack.





Having multiple items on the stack will cause the interface to show a platform dependent bar at the top for navigating backwards on the stack that might look similar to this:



## Switching Between Pages in Code

The page’s Navigation property has several methods that allow us to work with the stack. These are called like so:

Navigation.MethodNeeded(new SomeContentPage());

**PushAsync:**

Open a new instance of a Page.

The new page will be pushed onto the navigation stack.

**PopAsync:**

Close the current instance of the Page currently open.

The page will be popped from the navigation stack. \*\*Do not “Pop” modal pages, see below\*\*

**PushModalAsync:**

Open a new instance of a Page presented modally.

The modal page will slide in upwards.

**PopModalAsync:**

Close the current modal page.

[Xamarin.Forms Modal Pages - Xamarin | Microsoft Docs](https://docs.microsoft.com/en-us/xamarin/xamarin-forms/app-fundamentals/navigation/modal)

The page will slide out downwards. \*\*Do not “PopModal” non-modal pages\*\*

**PopToRootAsync:**

Close all but the root page.

All other pages will be removed from the navigation stack.

Also note that when a modal page is open, you should not try to pop additional pages.

The Page’s Title property, if set, will appear as the title in the navigation bar at top.

## Adding Additional XAML Pages

In Visual Studio you can add additional XAML pages by right clicking the shared project, choosing Add-New Item, navigate to the Xamarin Forms category and choosing Content Page.

Once multiple page files are in use, it is suggested that tasks such as wiring buttons to event handlers or populating the page with views be done within that page’s code behind file. Do this work after the call to the Initialize Component method.

## Passing Data to Pages

There are several approaches to passing data to other pages. The simplest is to simply add constructor parameters to the Pages that take in data. Then when that page is created the relevant data is passed in.

For example to pass an int to a Page first override the constructor to take an int parameter, then when popping that page onto the stack pass the relevant number into the constructor:

Navigation.PopAsync(new SomePage(100));

## Toolbar Items

A page may have a toolbar that displays at the top of the app. Links can be added to the toolbar that function effectively like buttons. The code to add a simple button to the toolbar looks like this:

ToolbarItem tb = new ToolbarItem { Text="Test"};

tb.Clicked += (sender, e) => { MainPage.Navigation.PushAsync(new Page1()); };

MainPage.ToolbarItems.Add(tb);

In the above code the toolbar button is just opening a page on the stack, but it could do other things.

# OnAppearing and OnDisappearing

OnAppearing allows the content page to react to it being displayed, perhaps when being returned to after navigating to another page. It is inherited from Page and can be overridden. Mainly useful if there is something on the second page that affects the first (perhaps the second page modifies a list that is shown on the first, so you want the list to update.)

Typically this is overridden to reinitialize data in the page that may have changed since the page was originally opened.