

Use of Toolbar

If we want to manipulate something based on our current view we can use toolbar.

Example would be the email app with an inbox item having options to delete, make favourite, reply and so on. It is shown below.



Important Properties

- barStyle
- items

Add a Custom Method addToolbar

```
-(void)addToolbar
{
    UIBarButtonItem *spaceItem = [[UIBarButtonItem alloc]
initWithBarButtonSystemItem:UIBarButtonSystemItemFlexibleSpace
target:nil action:nil];
    UIBarButtonItem *customItem1 = [[UIBarButtonItem alloc]
initWithTitle:@"Tool1" style:UIBarButtonItemStyleBordered
target:self action:@selector(toolBarItem1:)];
    UIBarButtonItem *customItem2 = [[UIBarButtonItem alloc]
initWithTitle:@"Tool2" style:UIBarButtonItemStyleDone
target:self action:@selector(toolBarItem2:)];
    NSArray *toolbarItems = [NSArray arrayWithObjects:
customItem1, spaceItem, customItem2, nil];
    UIToolbar *toolbar = [[UIToolbar alloc] initWithFrame:
CGRectMake(0, 366+54, 320, 50)];
    [toolbar setBarStyle:UIBarStyleBlackOpaque];
    [self.view addSubview:toolbar];
    [toolbar setItems:toolbarItems];
}
```

For knowing the action performed, we add a **UILabel** in our **ViewController.xib** and create an **IBOutlet** for the UILabel and name it as **label**.

We also need to add two methods in order to execute actions for toolbar items as shown below.

```
-(IBAction)toolBarItem1:(id)sender{
    [label setText:@"Tool 1 Selected"];
}

-(IBAction)toolBarItem2:(id)sender{
    [label setText:@"Tool 2 Selected"];
}
```

Update viewDidLoad in ViewController.m as follows –

```
-(void)viewDidLoad
{
    [super viewDidLoad];
    // The method hideStatusBar called after 2 seconds
    [self addToolbar];
    // Do any additional setup after loading the view, typically from a nib.
}
```

Output

When we run the application, we'll get the following output –



Click tool1 and tool2 bar buttons and we get the following –

