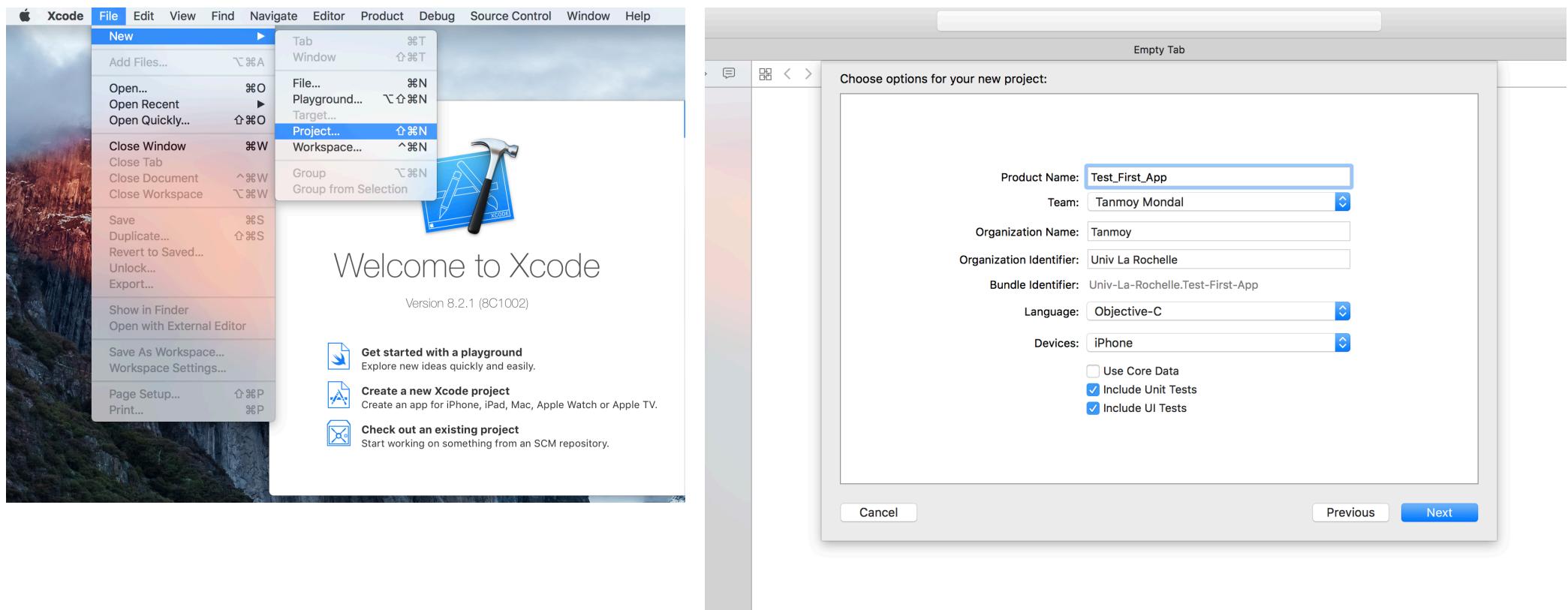


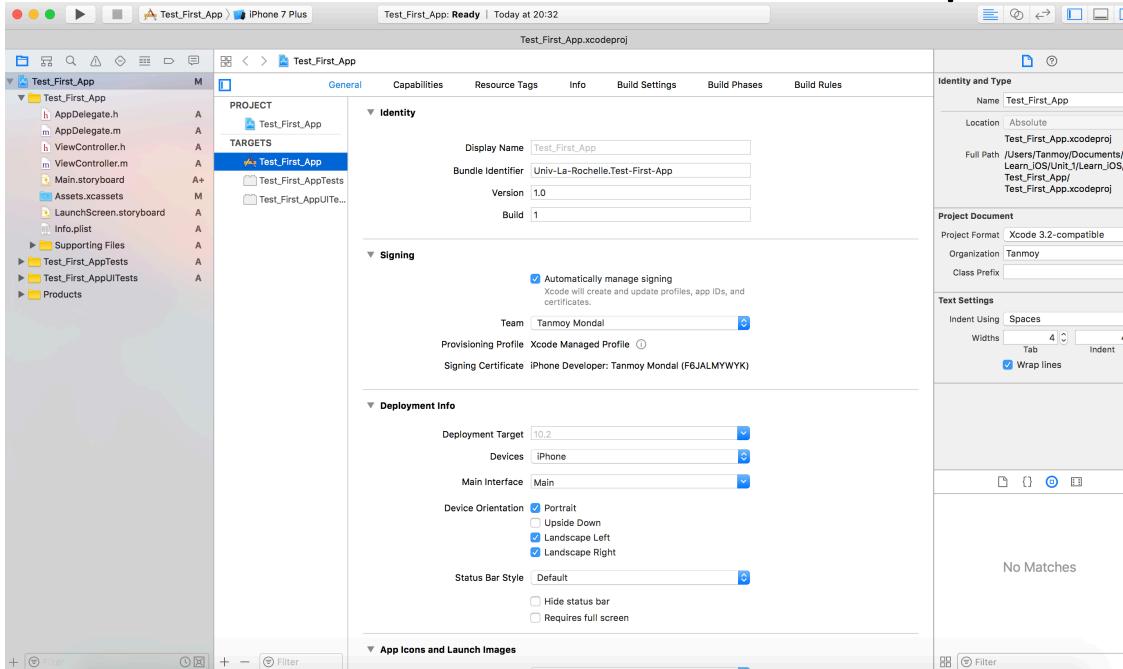
Création de la première application



Les diapositives sont faites à partir des matériaux à:

www.tutorialpoint.com

Création de la première application



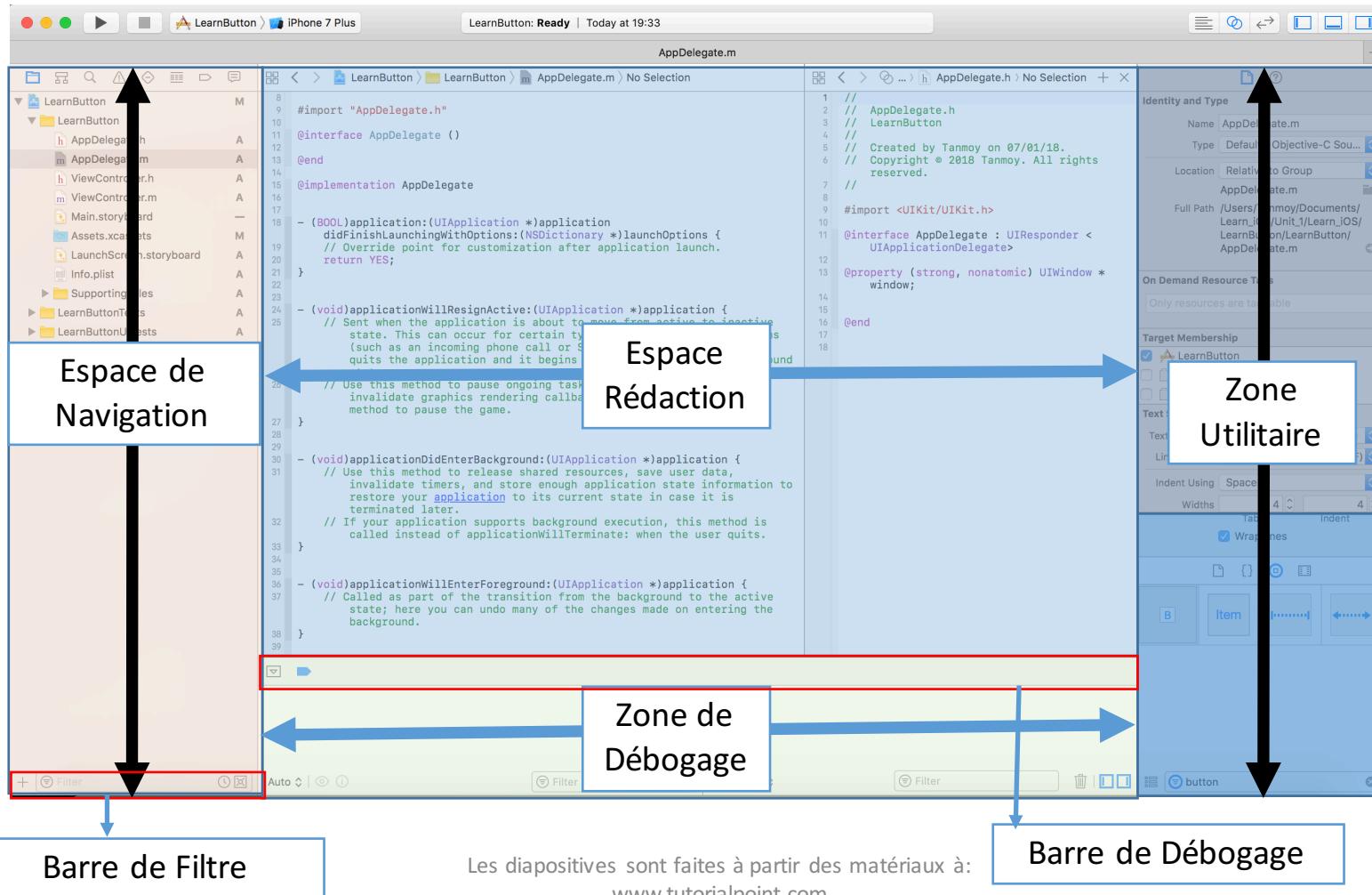
```
// AppDelegate.h
// Test_First_App
// Copyright © 2016 Tanmoy. All rights reserved.

#import <UIKit/UIKit.h>

@interface AppDelegate : UIResponder <UIApplicationDelegate>
@property (strong, nonatomic) UIWindow *window;
@end
```

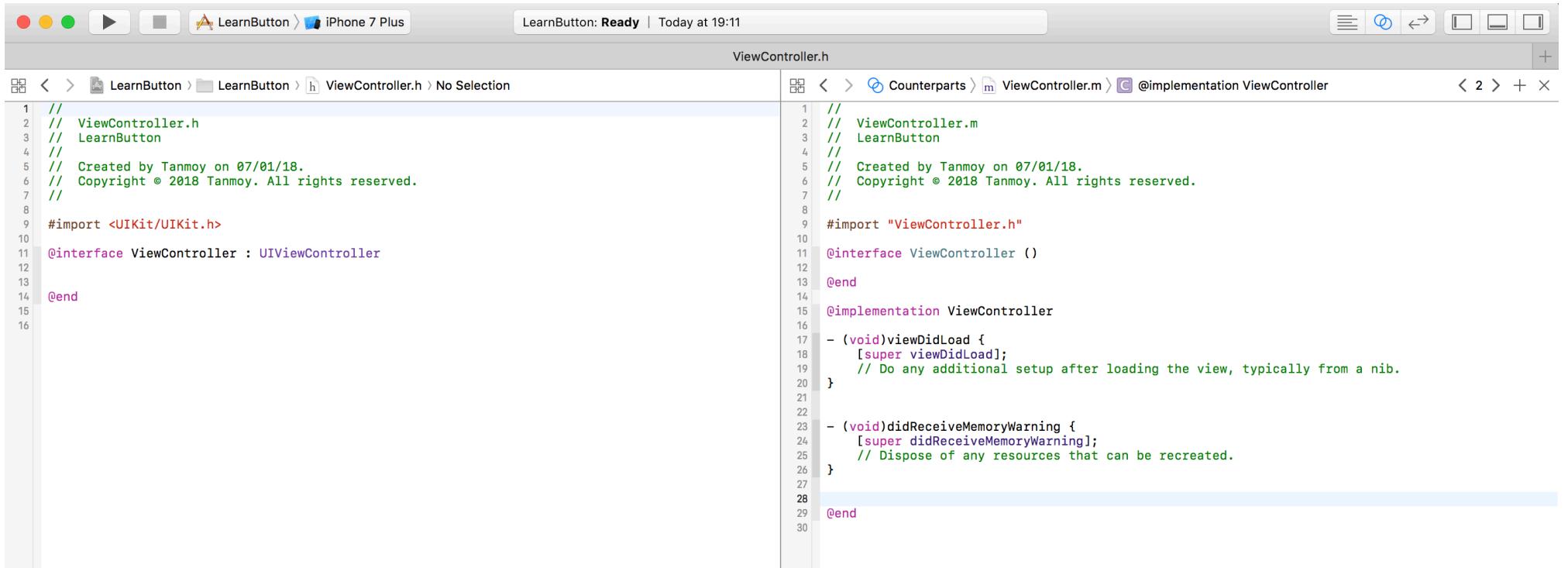
Les diapositives sont faites à partir des matériaux à:
www.tutorialpoint.com

Différent sections du Xcode ide



Création de la première application

Regardez les Fichier créé automatiquement dans le projet



The screenshot shows two Xcode code editors side-by-side. The left editor displays `ViewController.h` with the following content:

```
// ViewController.h
// LearnButton
//
// Created by Tanmoy on 07/01/18.
// Copyright © 2018 Tanmoy. All rights reserved.
//
#import <UIKit/UIKit.h>
@interface ViewController : UIViewController
@end
```

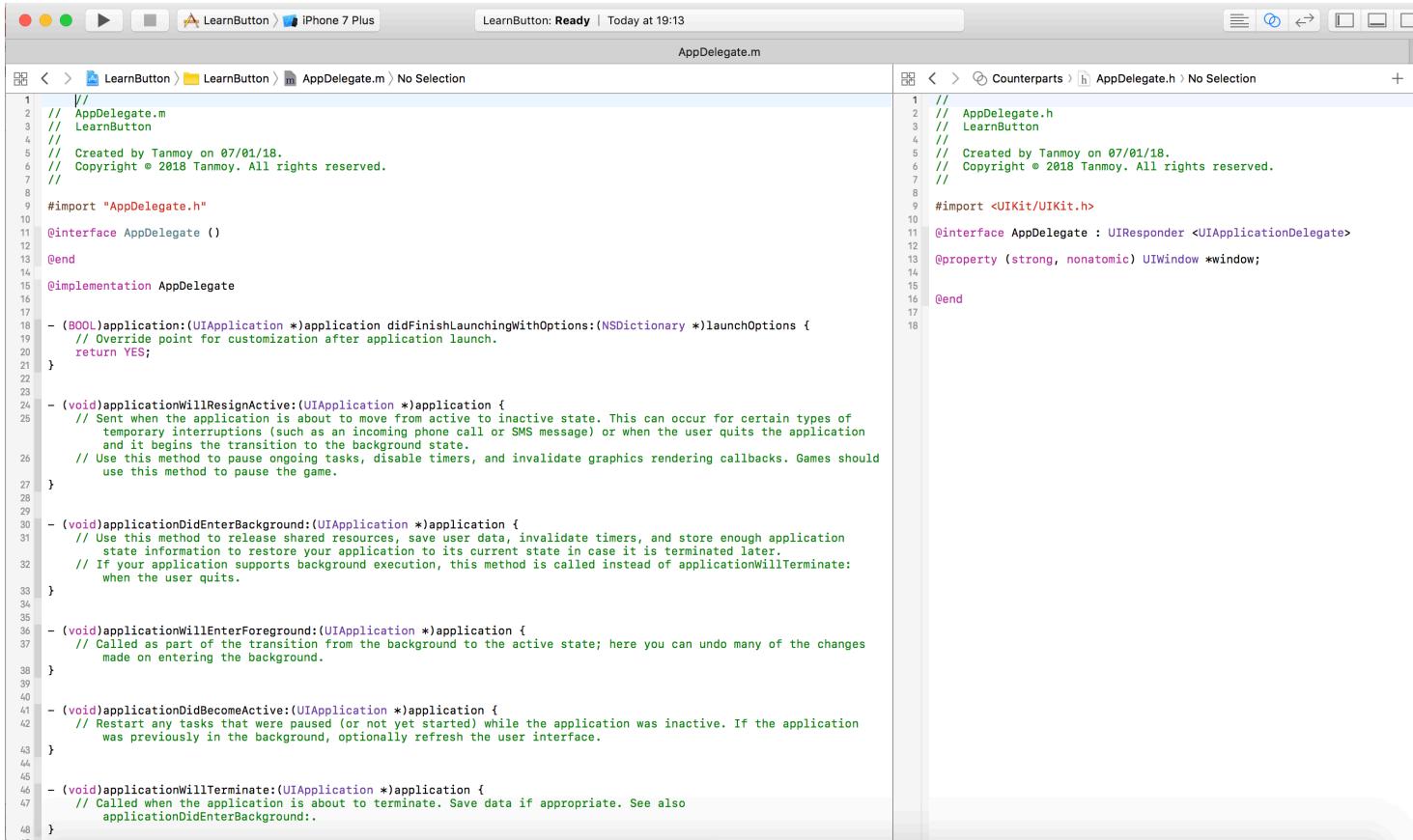
The right editor displays `ViewController.m` with the following content:

```
// ViewController.m
// LearnButton
//
// Created by Tanmoy on 07/01/18.
// Copyright © 2018 Tanmoy. All rights reserved.
//
#import "ViewController.h"
@interface ViewController ()
```

Both files contain identical boilerplate code for a UIViewController subclass.

Création de la première application

Regardez les Fichier créé automatiquement dans le projet



The screenshot shows the Xcode interface with two files open in separate windows:

- AppDelegate.m**: This file contains the implementation of the UIApplicationDelegate protocol. It includes methods for application launch, background/foreground transitions, and termination. The code is well-commented, explaining the purpose of each method.
- AppDelegate.h**: This file contains the declaration of the UIApplicationDelegate protocol, defining the methods implemented in AppDelegate.m.

```
Code content for AppDelegate.m:
```

```
1 // AppDelegate.m
2 // LearnButton
3 //
4 // Created by Tanmoy on 07/01/18.
5 // Copyright © 2018 Tanmoy. All rights reserved.
6 //
7 //
8 #import "AppDelegate.h"
9
10 @interface AppDelegate : UIResponder
11
12 @property (strong, nonatomic) UIWindow *window;
13
14 @implementation AppDelegate
15
16
17 - (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
18     // Override point for customization after application launch.
19     return YES;
20 }
21
22
23
24 - (void)applicationWillResignActive:(UIApplication *)application {
25     // Sent when the application is about to move from active to inactive state. This can occur for certain types of
26     // temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application
27     // and it begins the transition to the background state.
28     // Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering callbacks. Games should
29     // use this method to pause the game.
30 }
31
32
33 - (void)applicationDidEnterBackground:(UIApplication *)application {
34     // Use this method to release shared resources, save user data, invalidate timers, and store enough application
35     // state information to restore your application to its current state in case it is terminated later.
36     // If your application supports background execution, this method is called instead of applicationWillTerminate:
37     // when the user quits.
38 }
39
40
41 - (void)applicationWillEnterForeground:(UIApplication *)application {
42     // Called as part of the transition from the background to the active state; here you can undo many of the changes
43     // made on entering the background.
44 }
45
46
47 - (void)applicationDidBecomeActive:(UIApplication *)application {
48     // Restart any tasks that were paused (or not yet started) while the application was inactive. If the application
49     // was previously in the background, optionally refresh the user interface.
50 }
51
52
53
54 - (void)applicationWillTerminate:(UIApplication *)application {
55     // Called when the application is about to terminate. Save data if appropriate. See also
56     // applicationDidEnterBackground:
57 }
```

```
Code content for AppDelegate.h:
```

```
1 // AppDelegate.h
2 // LearnButton
3 //
4 // Created by Tanmoy on 07/01/18.
5 // Copyright © 2018 Tanmoy. All rights reserved.
6 //
7 //
8 #import <UIKit/UIKit.h>
9
10 @interface AppDelegate : UIResponder
11
12 @property (strong, nonatomic) UIWindow *window;
```

Les diapositives sont faites à partir des matériaux à:

www.tutorialpoint.com

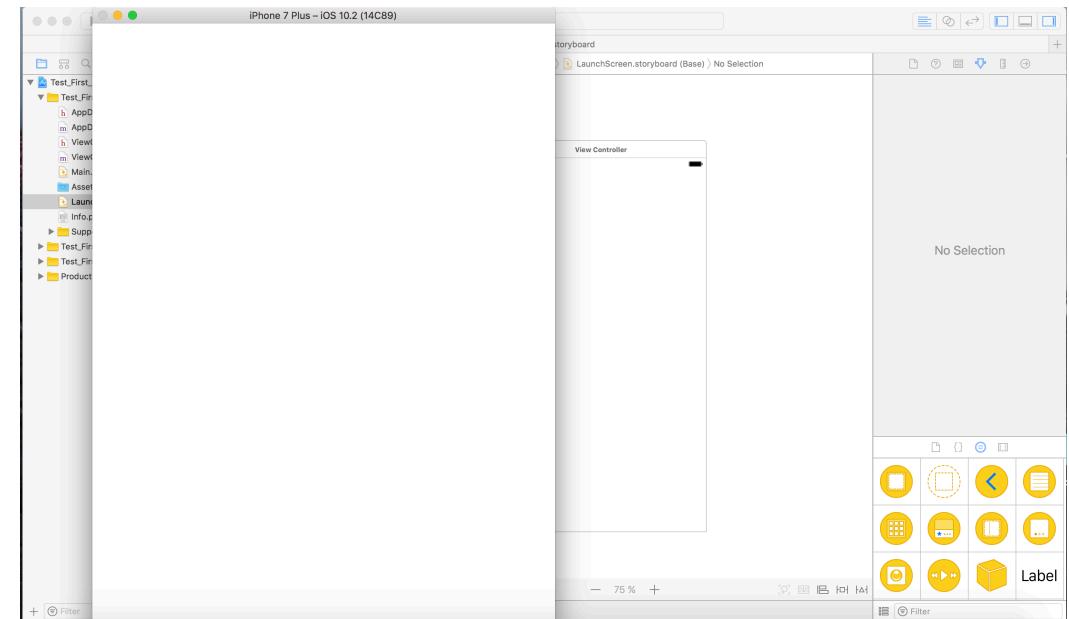
Création de la première application

Regardez les Fichier créé automatiquement dans le projet

The screenshot shows the Xcode interface with the title bar "Test_First_App" and "iPhone 7 Plus". The status bar indicates "Finished running Test_First_App on iPhone 7 Plus". The main area displays the file structure of the "Test_First_App" project, specifically the "Test_First_App" folder containing "AppDelegate.h", "AppDelegate.m", "ViewController.h", "ViewController.m", "Main.storyboard", "Assets.xcassets", "LaunchScreen.storyboard", "Info.plist", "Supporting Files", "Test_First_AppTests", "Test_First_AppUITests", and "Products". Below the file list is the content of "AppDelegate.h":

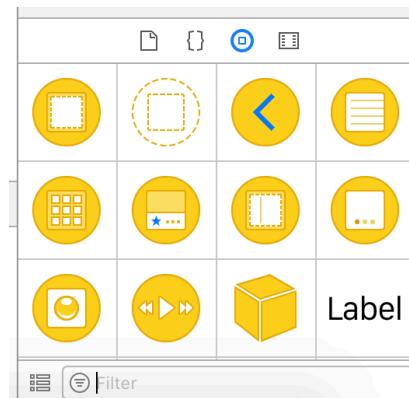
```
1 // AppDelegate.h
2 // Test_First_App
3 //
4 //
5 // Created by Tanmoy on 06/01/18.
6 // Copyright © 2018 Tanmoy. All rights reserved.
7 //
8
9 #import <UIKit/UIKit.h>
10
11 @interface AppDelegate : UIResponder <UIApplicationDelegate>
12
13 @property (strong, nonatomic) UIWindow *window;
14
15
16 @end
17
18
```

Lorsque vous allez compiler le projet



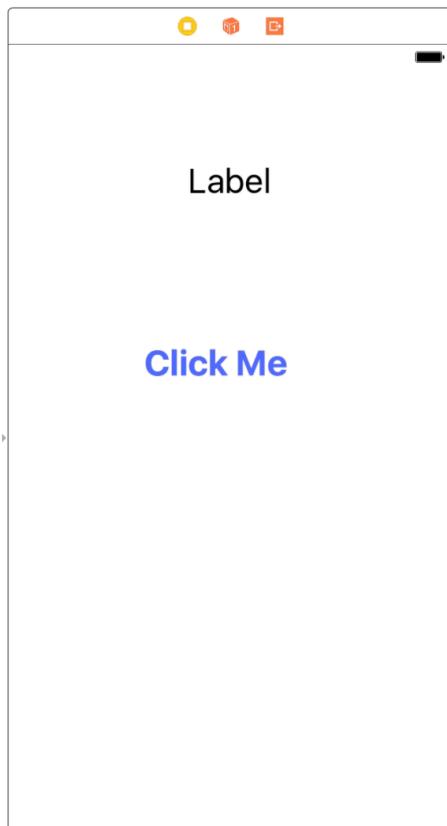
iOS – Actions et Outlets

- Les «Actions» et «Outlets» dans iOS sont appelées respectivement **ibActions** et **ibOutlets**
iOS – Actions et Outlets (Étapes impliquées)
 - Étape 1 - Utilisons notre première application iPhone
 - Étape 2 - Sélectionnez le fichier **ViewController. Xib** de la section Navigator
 - Étape 3 - Sélectionner les éléments de l'interface utilisateur dans le volet Bibliothèque



iOS – Actions et Outlets (Étapes impliquées)

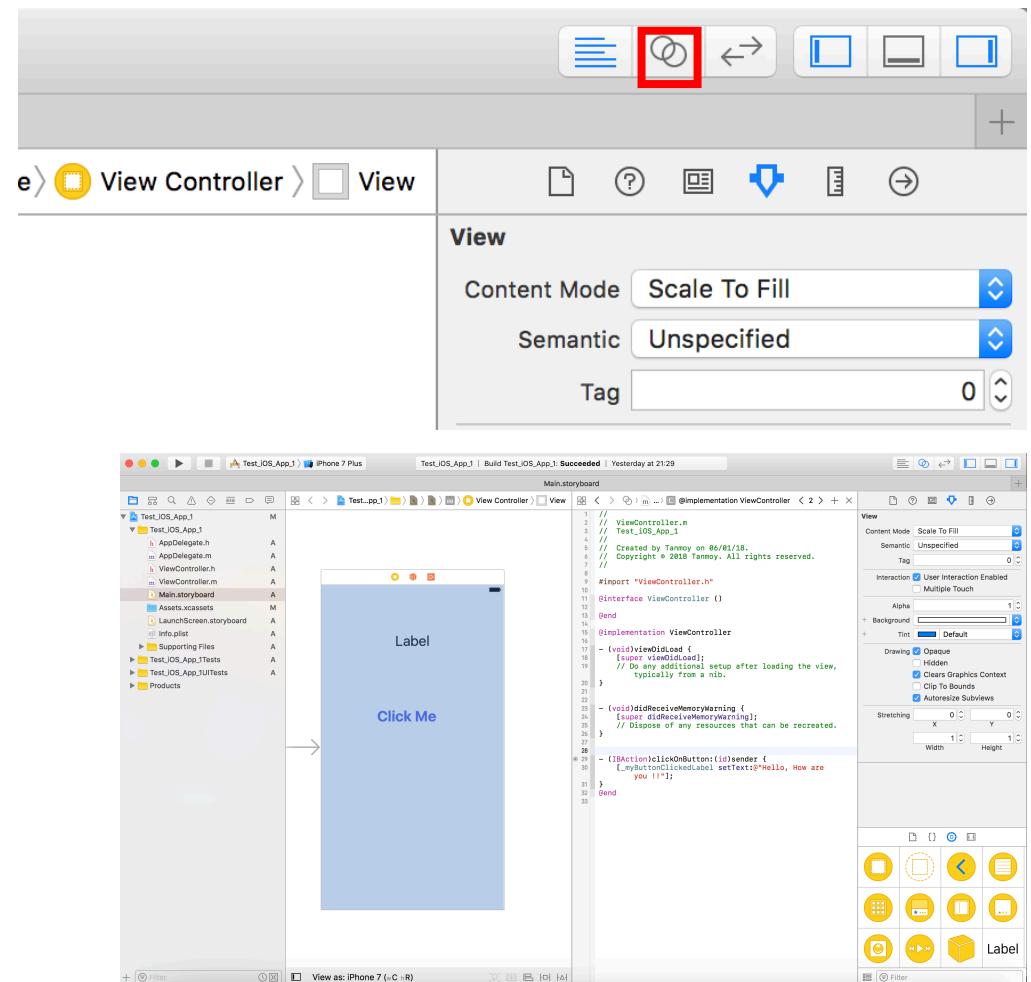
- Étape 4 - Vous pouvez **glisser-déposer** les éléments de l'interface utilisateur
- Étape 5 - Ajoutons une **étiquette et un bouton** rond rectangulaire à notre vue



Les diapositives sont faites à partir des matériaux à:
www.tutorialpoint.com

iOS – Actions et Outlets (Étapes impliquées)

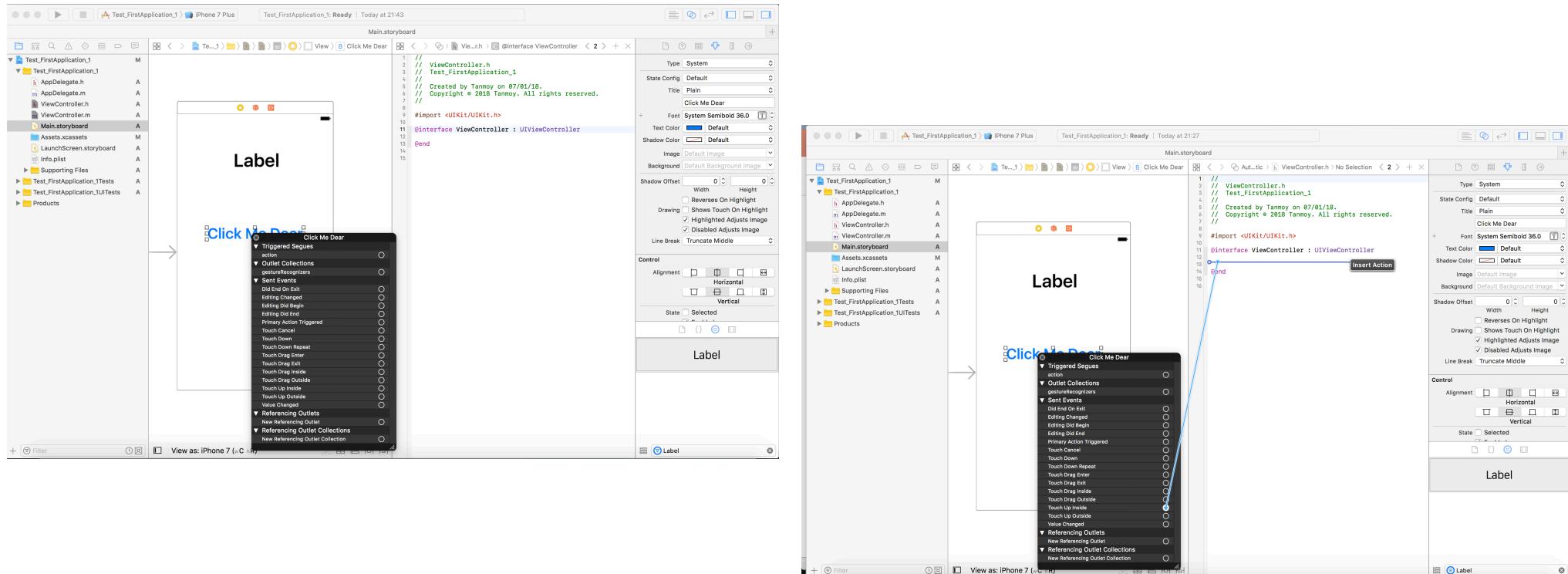
- Étape 6 – Cliquez sur le buttons dans la barre d'outils situé dans le coin supérieur droit
- Étape 7 - Deux fenêtres dans **notre zone d'édition** au centre, l'une est le fichier **ViewController. xib** et l'autre est **ViewController. h**



Les diapositives sont faites à partir des matériaux à:

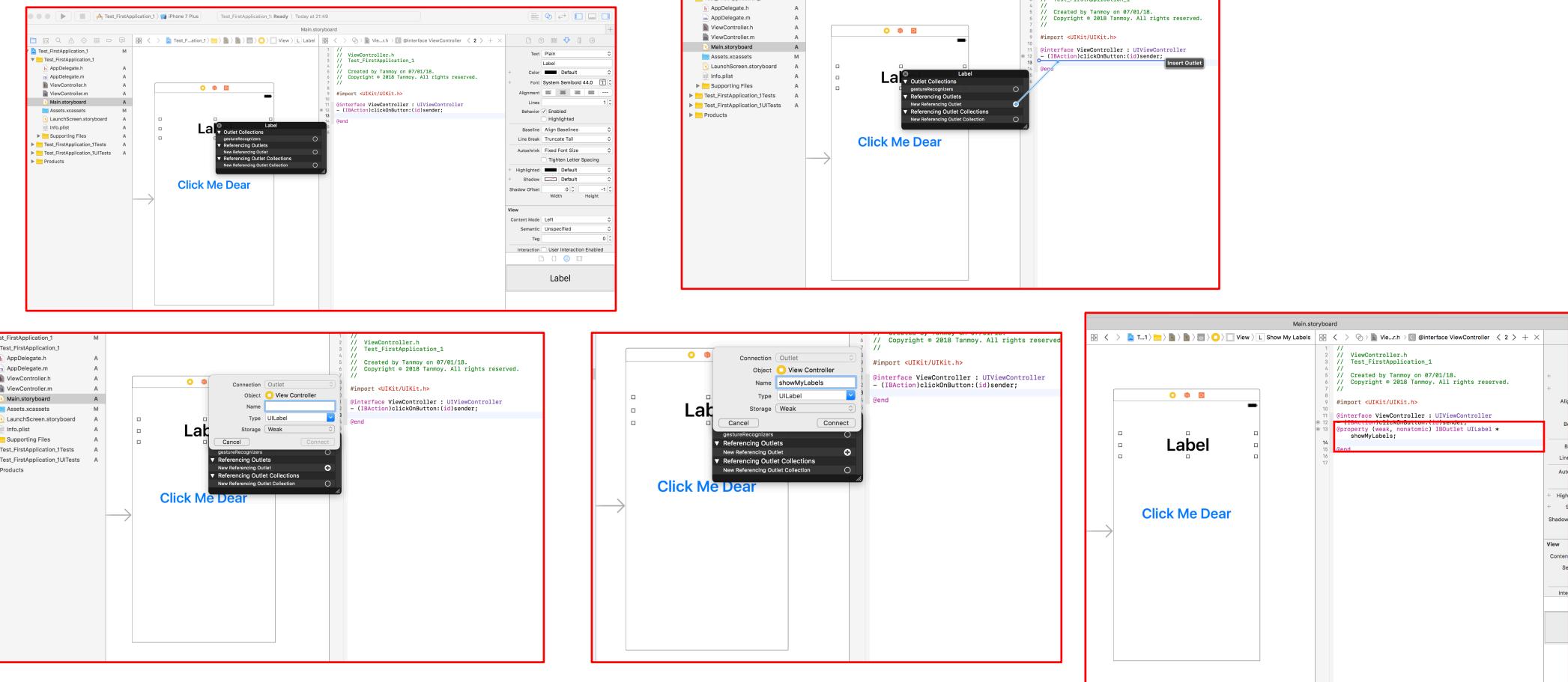
www.tutorialpoint.com

Ajout de l'action dans le bouton



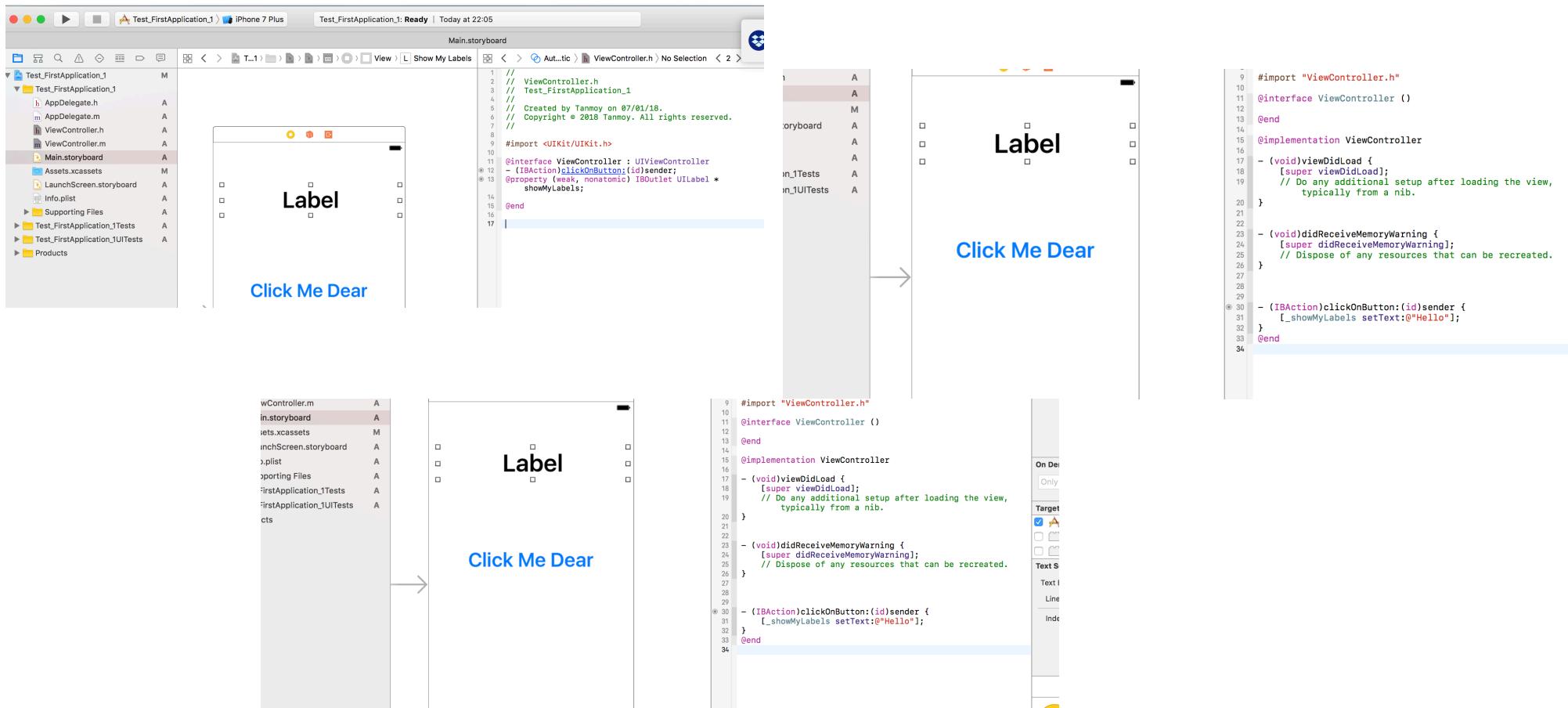
Les diapositives sont faites à partir des matériaux à:
www.tutorialpoint.com

Ajout de l'action dans le bouton



Les diapositives sont faites à partir des matériaux à:
www.tutorialpoint.com

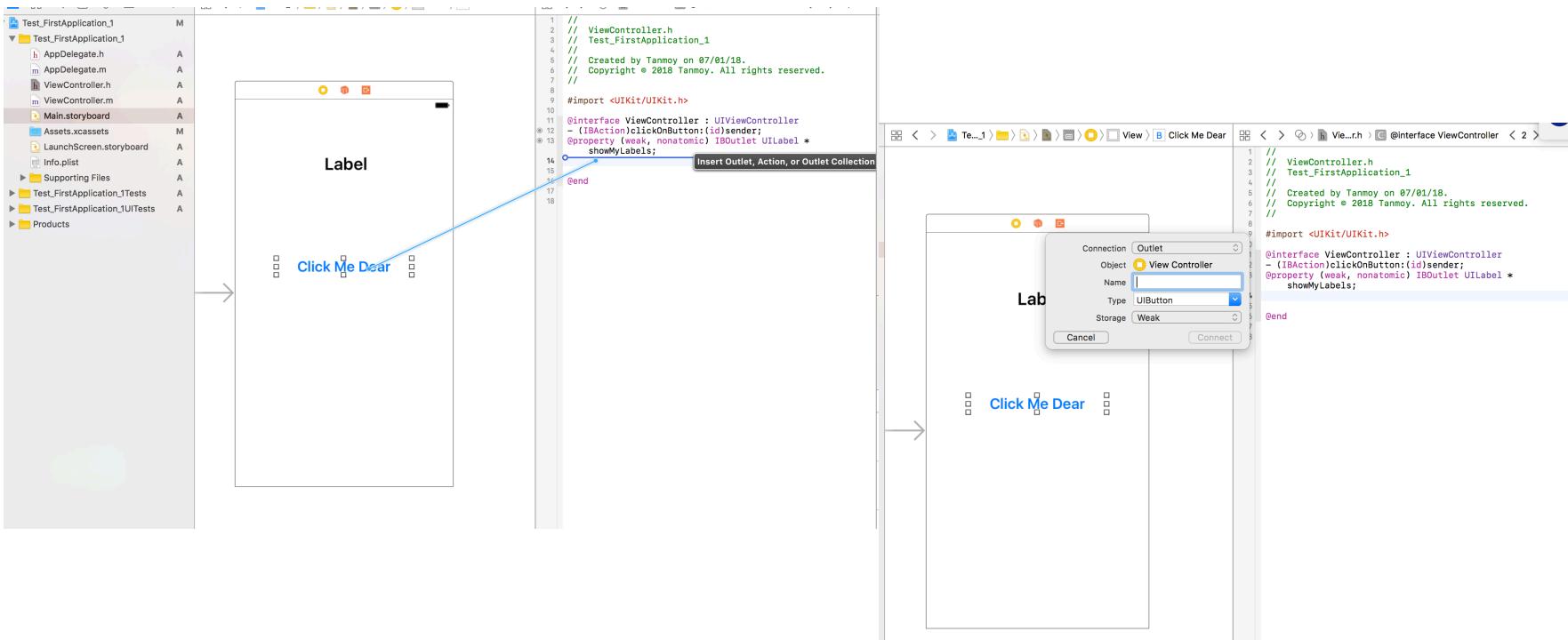
Ajout de l'action dans le bouton



Les diapositives sont faites à partir des matériaux à:

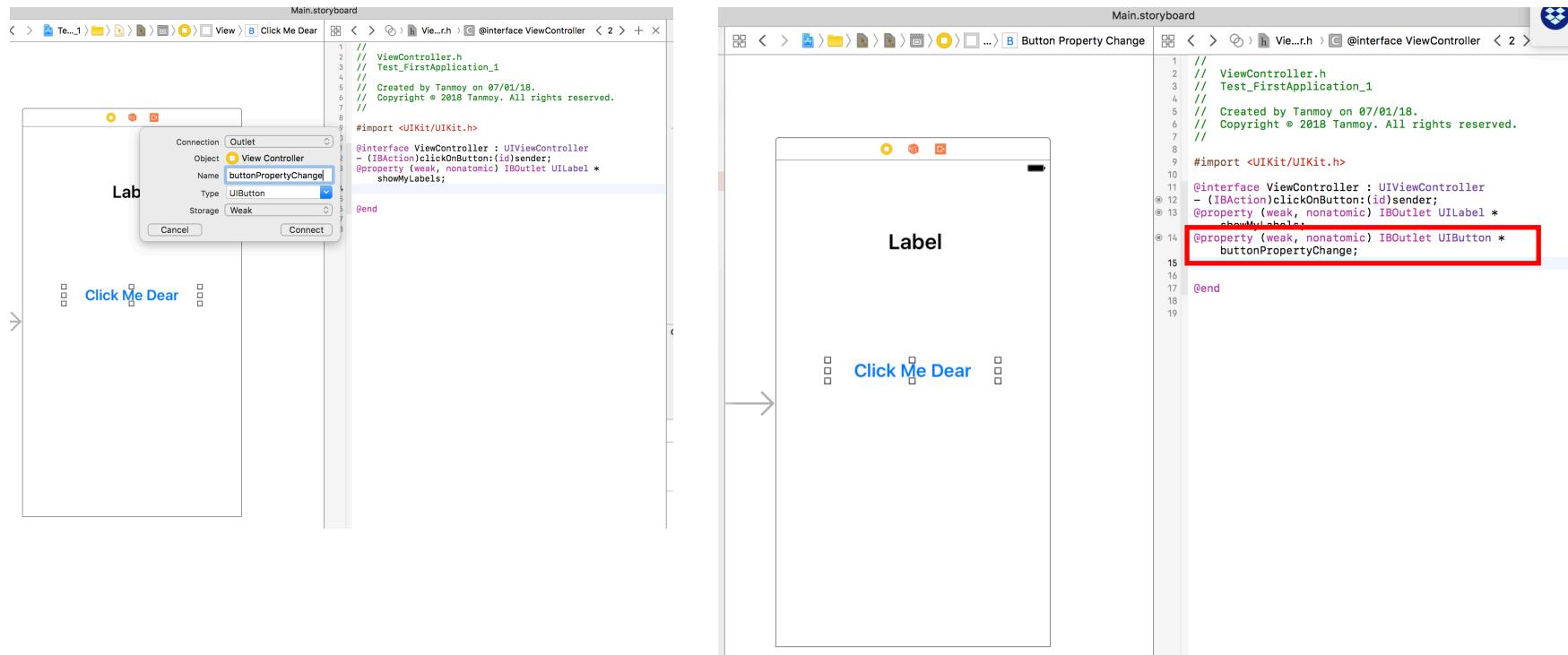
www.tutorialpoint.com

Changé de propriétés du bouton



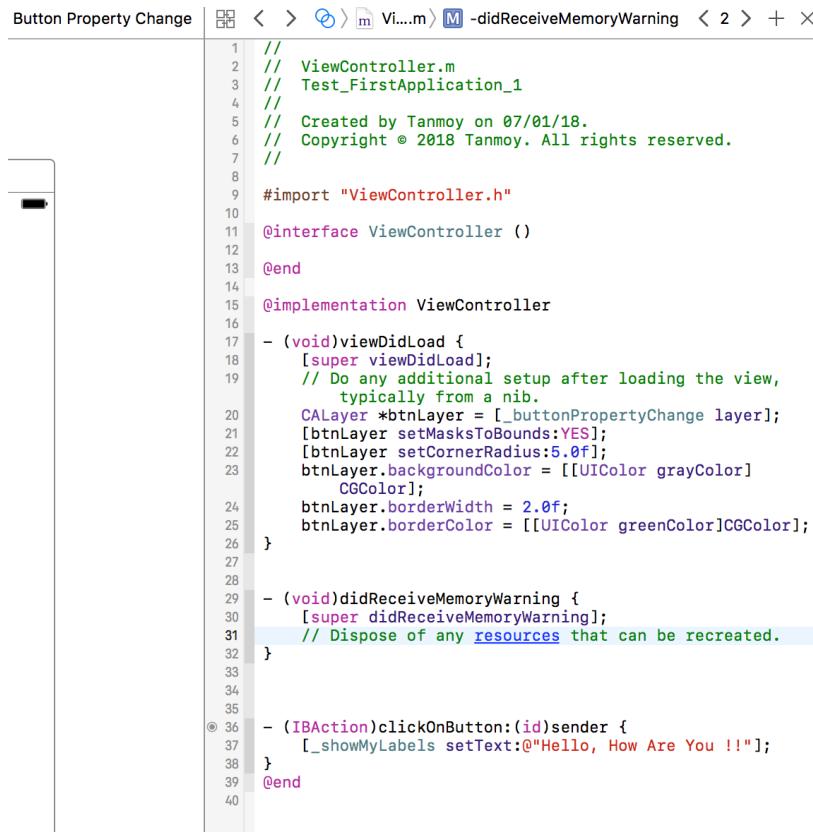
Les diapositives sont faites à partir des matériaux à:
www.tutorialpoint.com

Changé de propriétés du bouton



Les diapositives sont faites à partir des matériaux à:
www.tutorialpoint.com

Changé de propriétés du bouton



The screenshot shows the Xcode code editor with the file "Button Property Change.m" open. The code implements a custom button layer for a "ViewController". It includes methods for viewDidLoad, didReceiveMemoryWarning, and clickOnButton. The didReceiveMemoryWarning method contains a comment about disposing of resources.

```
// ViewController.m
// Test_FirstApplication_1
//
// Created by Tanmoy on 07/01/18.
// Copyright © 2018 Tanmoy. All rights reserved.

#import "ViewController.h"

@interface ViewController : UIViewController

@end

@implementation ViewController

- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view,
    // typically from a nib.
    CALayer *btnLayer = [_buttonPropertyChange layer];
    [btnLayer setMasksToBounds:YES];
    [btnLayer setCornerRadius:5.0f];
    btnLayer.backgroundColor = [[UIColor grayColor] CGColor];
    btnLayer.borderWidth = 2.0f;
    btnLayer.borderColor = [[UIColor greenColor] CGColor];
}

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

- (IBAction)clickOnButton:(id)sender {
    [_showMyLabels setText:@"Hello, How Are You !!"];
}

@end
```

Delegates

- **Delegate** est simplement un **moyen de communication** entre les objets des applications iOS
- La **Delegate** est un **moyen simple** de relier des objets et de communiquer entre eux
- **Delegates** permet à un objet d'envoyer un message à un autre objet lorsqu'un événement se produit

Delegates Exemple

- L'objet **Vincent** appelle un objet **Nicolas** pour effectuer une action
- Lorsque l'action terminée, l'objet **Vincent** doit savoir que **Nicolas** à terminé la tâche
- Maintenant, **Vincent** peut prendre les mesures nécessaires

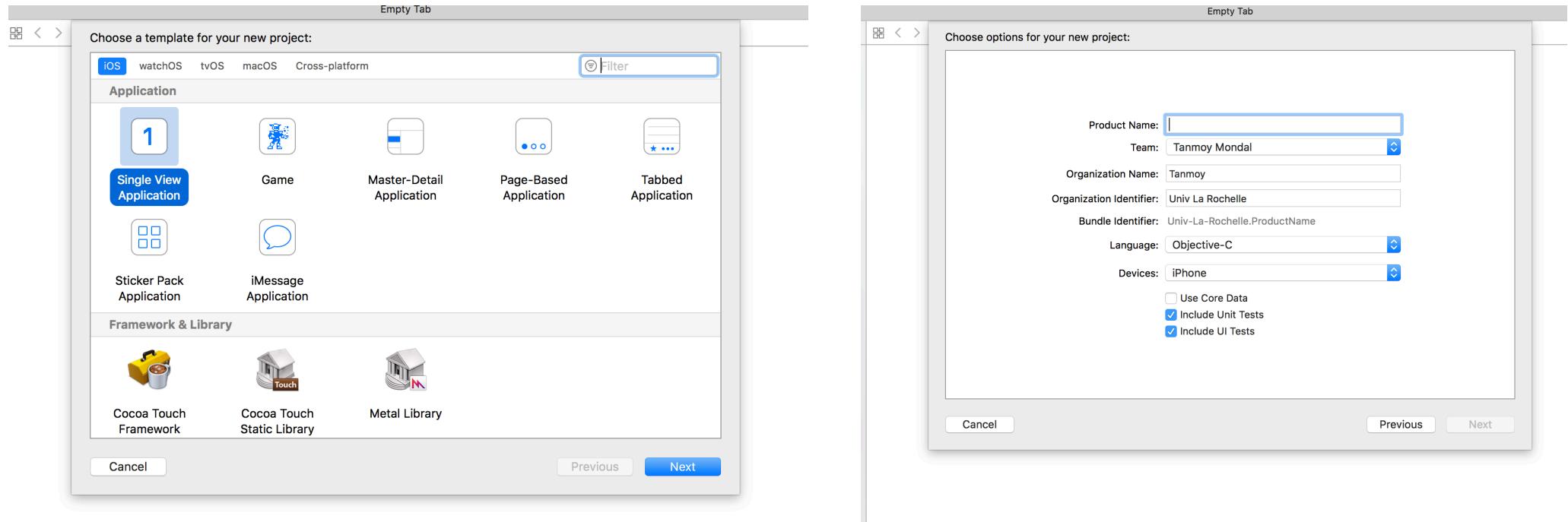
Delegates Exemple

Les concepts importante dans l'exemple sont

- Vincent est un “delegate objet” de Nicolas
- Nicolas aura une référence de Vincent
- Vincent va implemente les “delegate méthodes” de Nicolas
- Nicolas va signaler vincent par les “delegate méthodes”

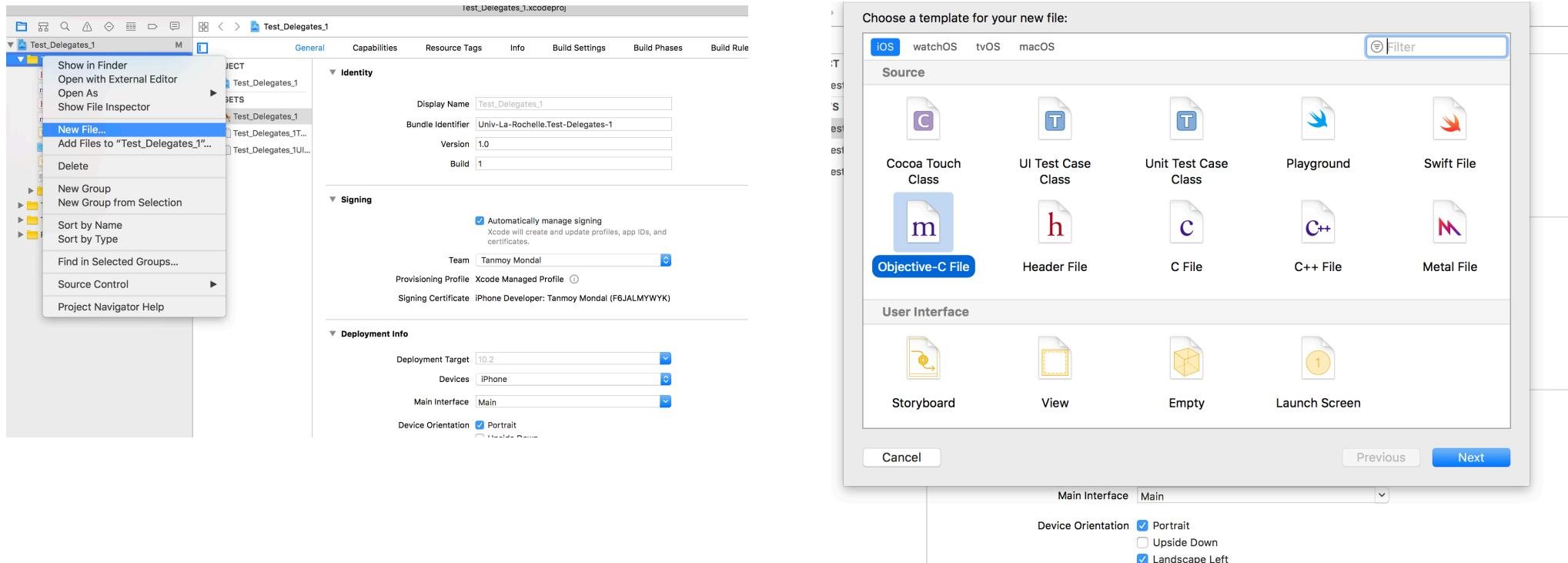
Delegates Exemple

- Crée un « single view application» projet



Delegates Exemple

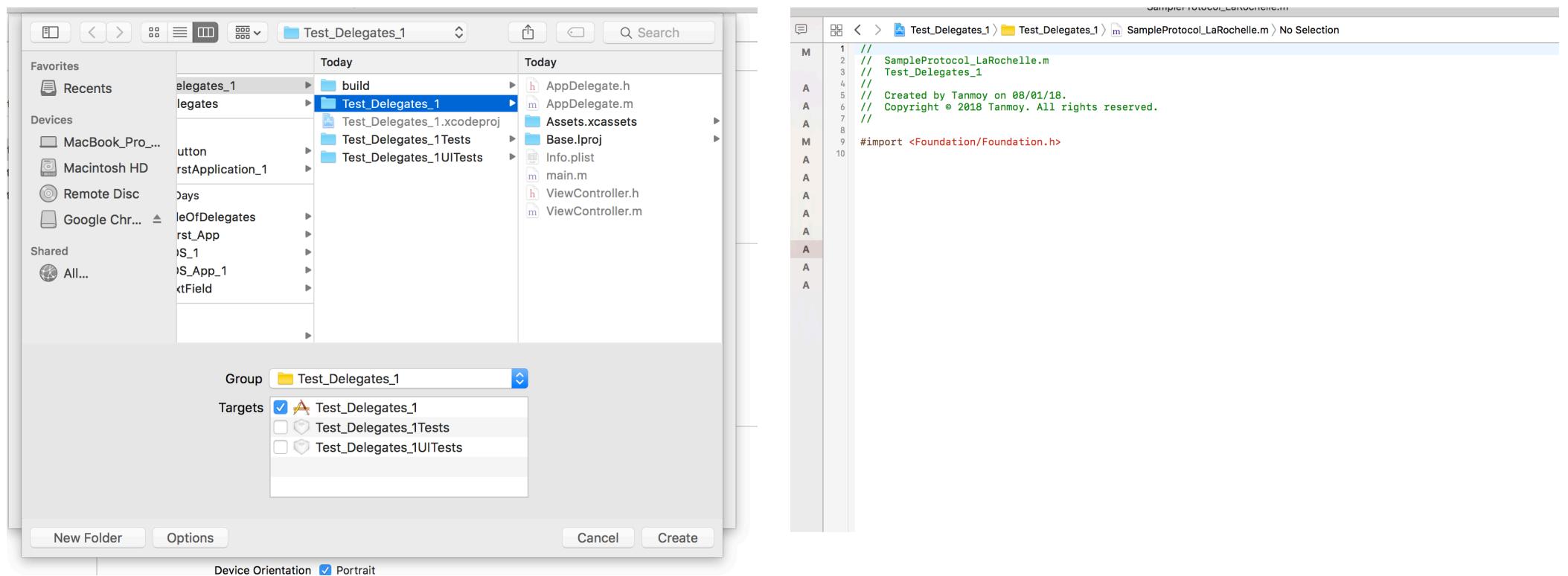
- Ajouter un fichier dans le projet



Les diapositives sont faites à partir des matériaux à:
www.tutorialpoint.com

Delegates Exemple

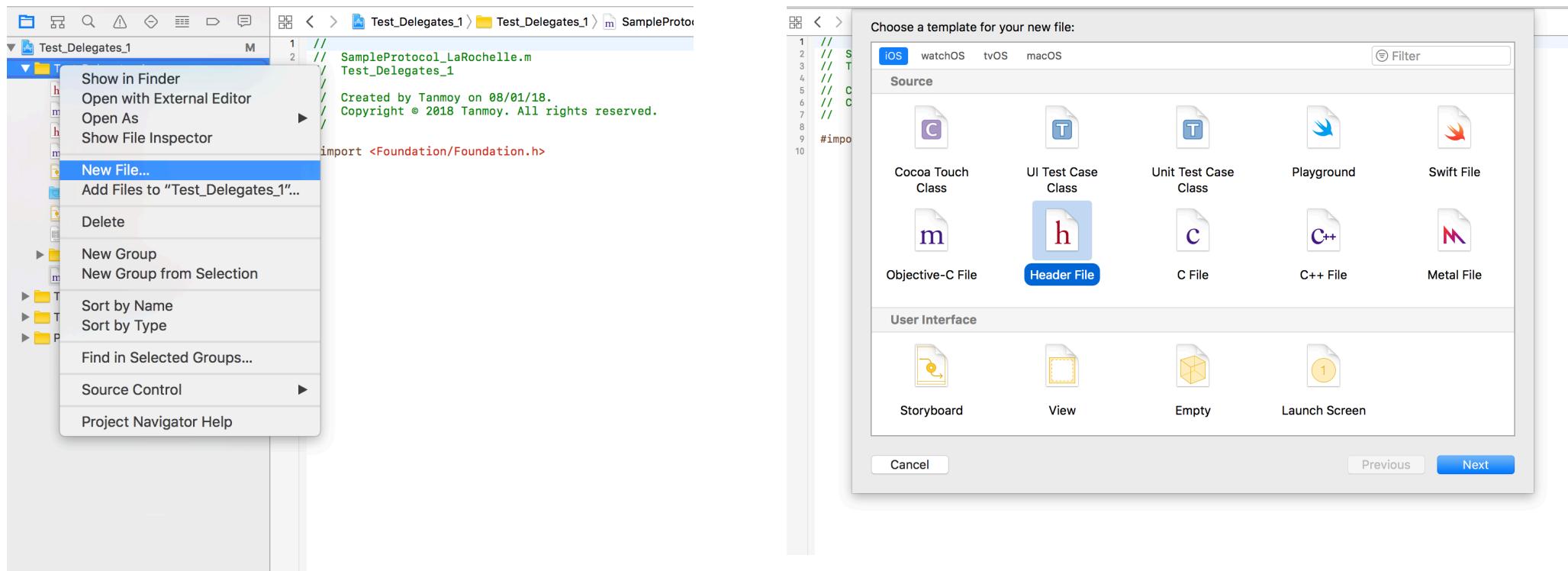
- Ajouter un fichier dans le projet



Les diapositives sont faites à partir des matériaux à:
www.tutorialpoint.com

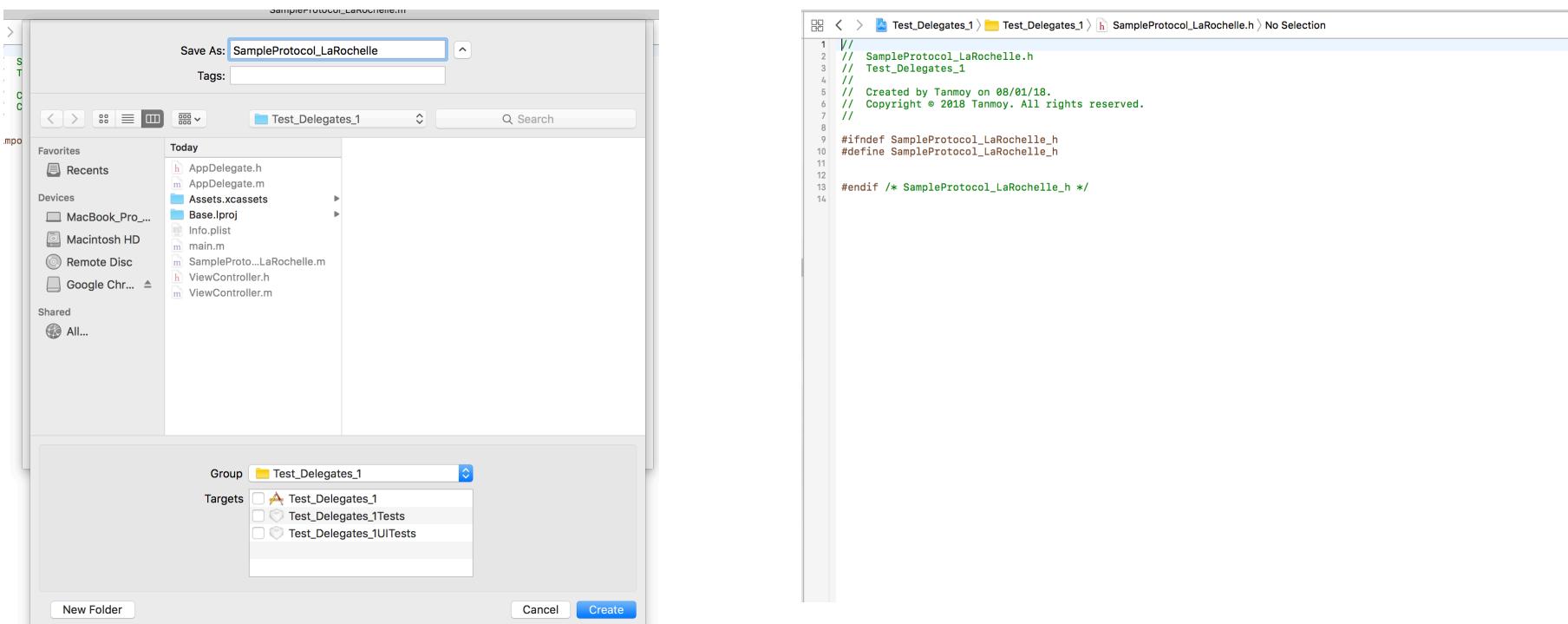
Delegates Exemple

- Crée un « header file» par le même nom comme le dernier fichier



Delegates Exemple

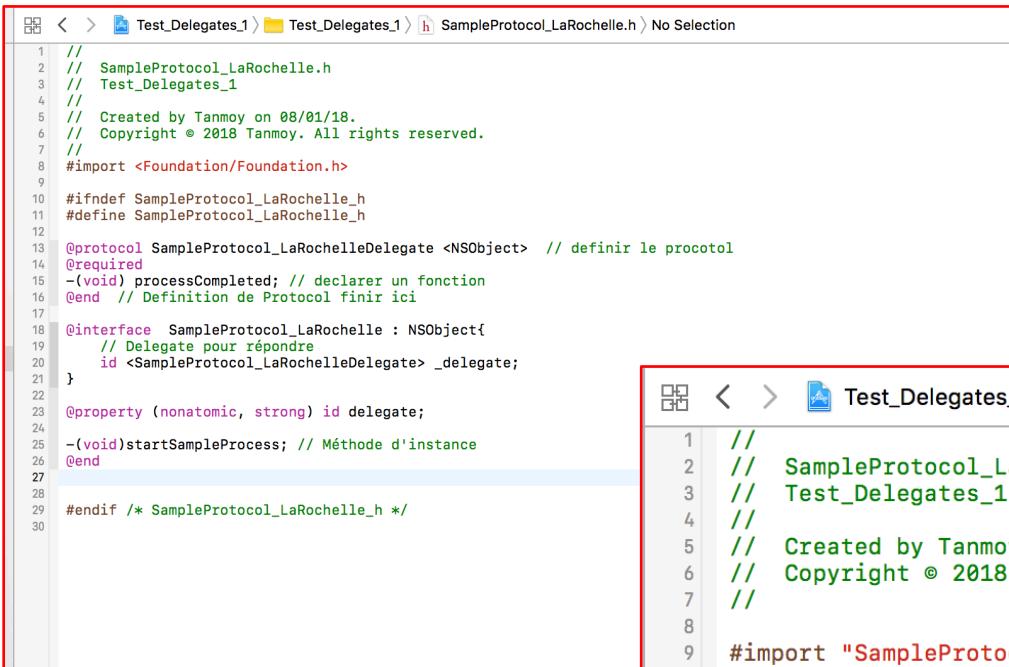
- Crée un « header file» par le même nom comme le dernier fichier



Les diapositives sont faites à partir des matériaux à:
www.tutorialpoint.com

Delegates Exemple

- Mettez ces ligne de code dans le fichier .m et .h



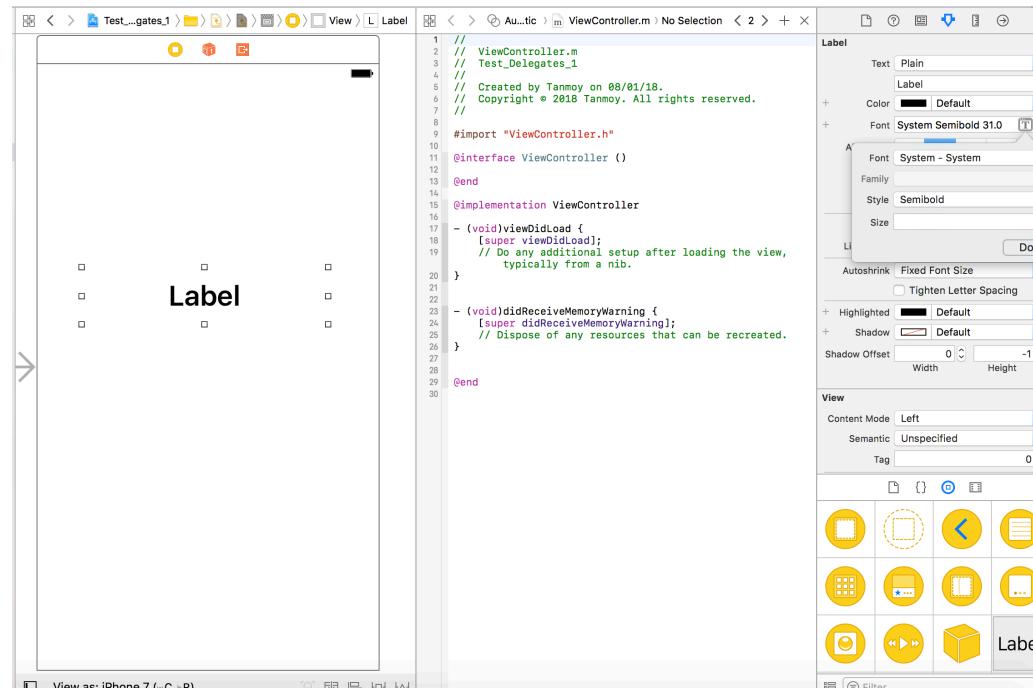
```
1 //  
2 // SampleProtocol_LaRochelle.h  
3 // Test_Delegates_1  
4 //  
5 // Created by Tanmoy on 08/01/18.  
6 // Copyright © 2018 Tanmoy. All rights reserved.  
7 //  
8 #import <Foundation/Foundation.h>  
9  
10 #ifndef SampleProtocol_LaRochelle_h  
11 #define SampleProtocol_LaRochelle_h  
12  
13 @protocol SampleProtocol_LaRochelleDelegate <NSObject> // definir le proctocol  
14 @required  
15 -(void) processCompleted; // declarer un fonction  
16 @end // Definition de Protocol finir ici  
17  
18 @interface SampleProtocol_LaRochelle : NSObject{  
19     // Delegate pour répondre  
20     id <SampleProtocol_LaRochelleDelegate> _delegate;  
21 }  
22  
23 @property (nonatomic, strong) id delegate;  
24  
25 -(void)startSampleProcess; // Méthode d'instance  
26 @end  
27  
28  
29 #endif /* SampleProtocol_LaRochelle_h */
```



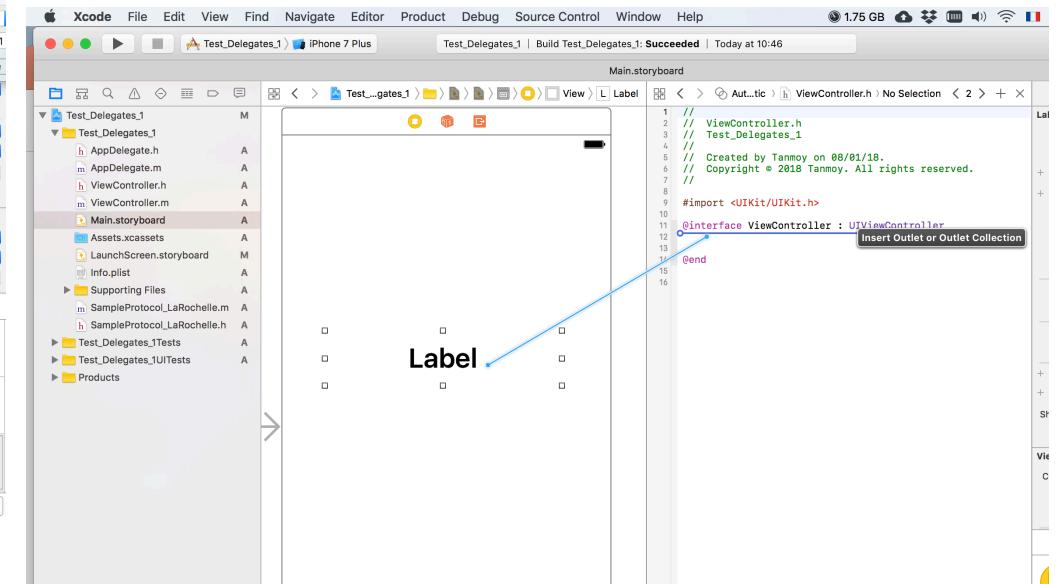
```
1 //  
2 // SampleProtocol_LaRochelle.m  
3 // Test_Delegates_1  
4 //  
5 // Created by Tanmoy on 08/01/18.  
6 // Copyright © 2018 Tanmoy. All rights reserved.  
7 //  
8  
9 #import "SampleProtocol_LaRochelle.h"  
10  
11 @implementation SampleProtocol_LaRochelle  
12 -(void) startSampleProcess{  
13     [NSTimer scheduledTimerWithTimeInterval:3.0 target:self.delegate  
                                         selector:@selector(processCompleted) userInfo:nil repeats:NO];  
14 }  
15  
16 @end
```

Delegates Exemple

Ajouter un label dans « main.storyboard »



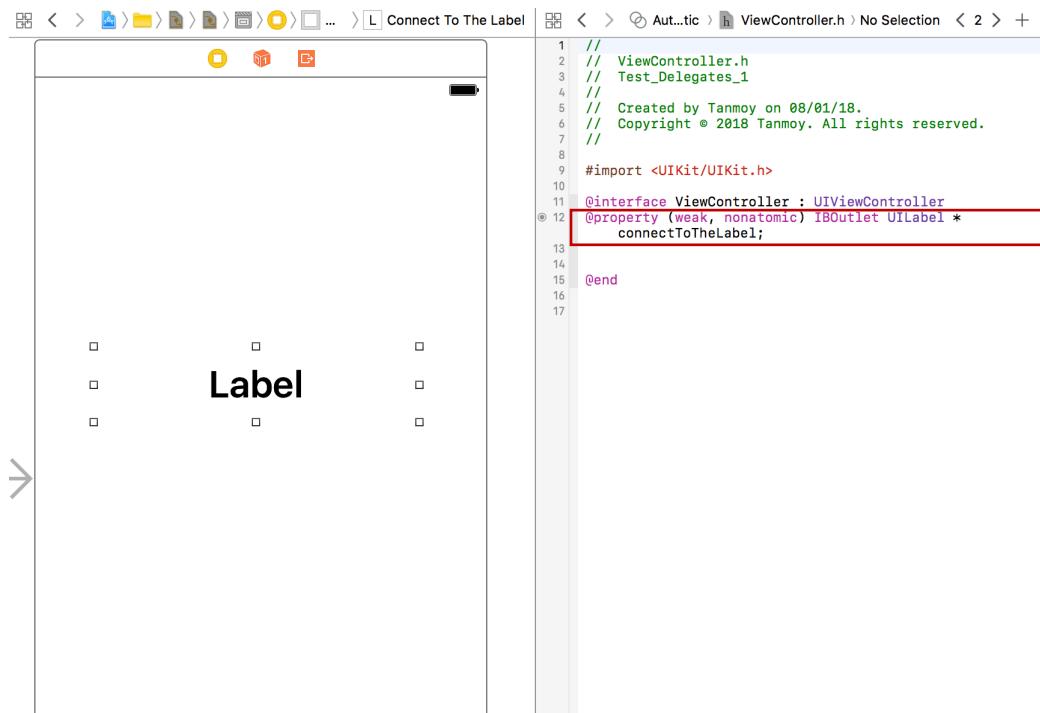
Ctrl+click pour connecter dans le fichier
« ViewController.h »



Les diapositives sont faites à partir des matériaux à:
www.tutorialpoint.com

Delegates Exemple

Ajouter un label dans « main.storyboard »



Delegates Exemple

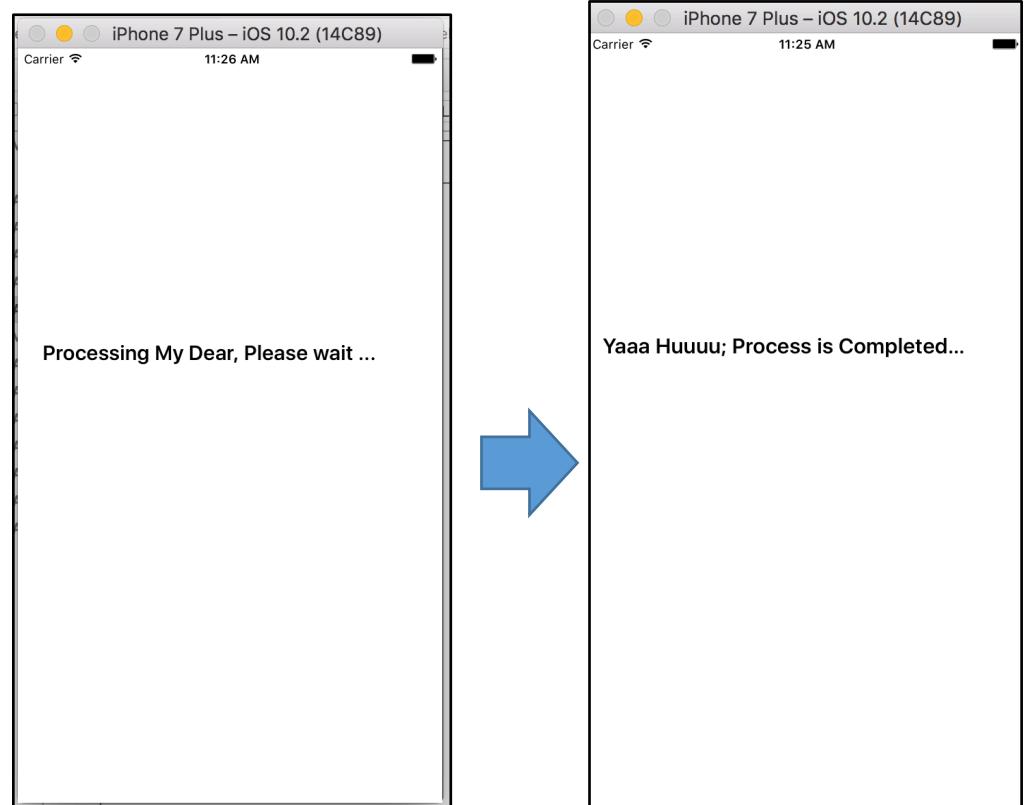
Modifier le code comme suit pour adopter **SampleProtocol_LaRochelleDelegate** dans ViewController. h.

```
1 //  
2 // ViewController.h  
3 // Test_Delegates_1  
4 //  
5 // Created by Tanmoy on 08/01/18.  
6 // Copyright © 2018 Tanmoy. All rights reserved.  
7 //  
8  
9 #import <UIKit/UIKit.h>  
10 #import "SampleProtocol_LaRochelle.h"  
11 @interface ViewController : UIViewController<SampleProtocol_LaRochelleDelegate>  
12 @property (weak, nonatomic) IBOutlet UILabel *connectToTheLabel;  
13  
14  
15 @end  
16  
17
```

Delegates Exemple

Modifier le code dans ViewController.m

```
1 // ViewController.m
2 // Test_Delegates_1
3 // Test_Delegates_1
4 //
5 // Created by Tanmoy on 08/01/18.
6 // Copyright © 2018 Tanmoy. All rights reserved.
7 //
8
9 #import "ViewController.h"
10
11 @interface ViewController : UIViewController
12
13 @end
14
15 @implementation ViewController
16
17 - (void)viewDidLoad {
18     [super viewDidLoad];
19     SampleProtocol_LaRochelle *sampleProtocol_Paris =
20         [[SampleProtocol_LaRochelle alloc] init];
21     sampleProtocol_Paris.delegate = self;
22     [_connectToTheLabel setText:@"Processing My Dear, Please wait ..."];
23     [sampleProtocol_Paris startSampleProcess];
24     // Do any additional setup after loading the view, typically from a nib.
25 }
26
27 - (void)didReceiveMemoryWarning {
28     [super didReceiveMemoryWarning];
29     // Dispose of any resources that can be recreated.
30 }
31 #pragma mark - Sample Protocol LaRochelle Delegate
32 -(void)processCompleted{
33     [_connectToTheLabel setText:@"Yaaa Huuuu; Process is Completed Now"];
34 }
35
36 @end
37
```

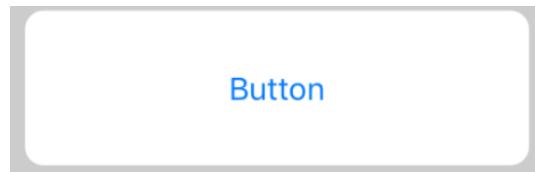


iOS Boutons

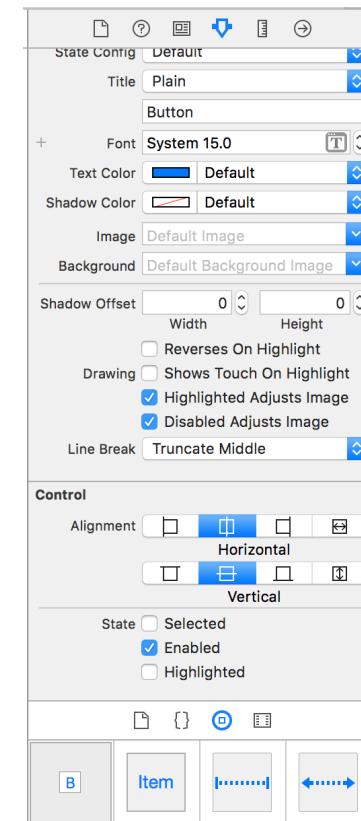
Utilisation des boutons

- Les boutons sont utilisés pour gérer les actions de l'utilisateur
- Il intercepte les événements tactiles et envoie un message à l'objet cible

Un bouton rond et rectangulaire



Propriétés du bouton dans xib



Les diapositives sont faites à partir des matériaux à:
www.tutorialpoint.com

iOS Boutons

Créer un « Single View Application »

Ajouter une méthode personnalisée

```
// A rounded Rect button created by using class method
UIButton *roundRectButton = [UIButton buttonWithType:
                             UIButtonTypeRoundedRect];
[roundRectButton setFrame:CGRectMake(60, 50, 200, 40)];
[roundRectButton setBackgroundColor: [UIColor blueColor]];
roundRectButton.layer.cornerRadius = 10;
// sets title for the button
[roundRectButton setTitle:@"Rounded Rect Button" forState:
                     UIControlStateNormal];
[self.view addSubview:roundRectButton];
```

```
UIButton *customButton = [UIButton buttonWithType: UIButtonTypeCustom];
[customButton setBackgroundColor: [UIColor lightGrayColor]];
[customButton setTitleColor:[UIColor blackColor] forState:
                     UIControlStateHighlighted];
//sets background image for normal state
[customButton setBackgroundImage:[UIImage imageNamed:
                               @"button_images.jpeg"]
                           forState:UIControlStateNormal];
//sets background image for highlighted state
[customButton setBackgroundImage:[UIImage imageNamed:
                               @"images.jpeg"]
                           forState:UIControlStateHighlighted];
[customButton setFrame:CGRectMake(60, 100, 200, 40)];
[customButton setTitle:@"Custom Button" forState:UIControlStateNormal];
[self.view addSubview:customButton];
```

```
UIButton *detailDisclosureButton = [UIButton buttonWithType:
                                       UIButtonTypeDetailDisclosure];
[detailDisclosureButton setFrame:CGRectMake(60, 150, 200, 40)];
[detailDisclosureButton setTitle:@"Detail disclosure" forState:
                     UIControlStateNormal];
[self.view addSubview:detailDisclosureButton];
```

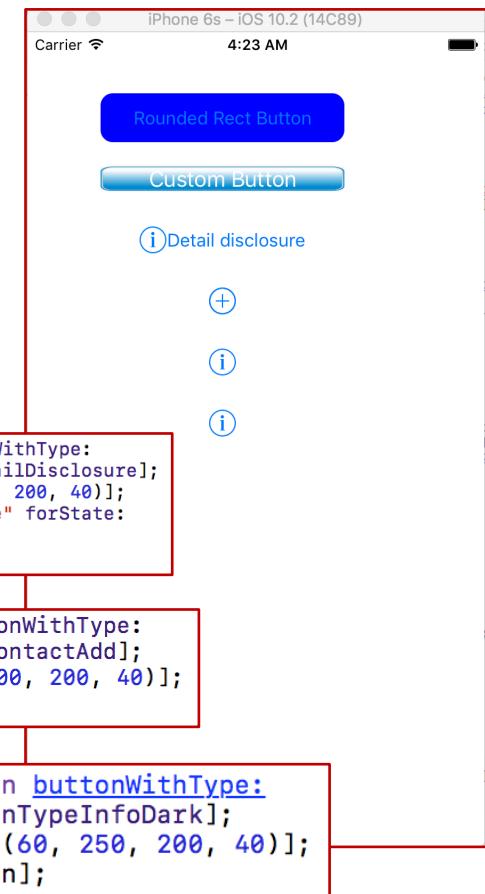
```
UIButton *contactButton = [UIButton buttonWithType:
                            UIButtonTypeContactAdd];
[contactButton setFrame:CGRectMake(60, 200, 200, 40)];
[self.view addSubview:contactButton];
```

```
UIButton *infoDarkButton = [UIButton buttonWithType:
                            UIButtonTypeInfoDark];
[infoDarkButton setFrame:CGRectMake(60, 250, 200, 40)];
[self.view addSubview:infoDarkButton];
```

```
@implementation ViewController
- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    //The custom method to create our different types of button is called
    [self addDifferentTypesOfButton];
}
```

```
UIButton *infoLightButton = [UIButton buttonWithType:
                            UIButtonTypeInfoLight];
[infoLightButton setFrame:CGRectMake(60, 300, 200, 40)];
[self.view addSubview:infoLightButton];
```

des matériaux à:



iOS «Labels»

Utilisation des «Labels»

Les étiquettes sont utilisées pour afficher le contenu statique, qui consiste en une ou plusieurs lignes

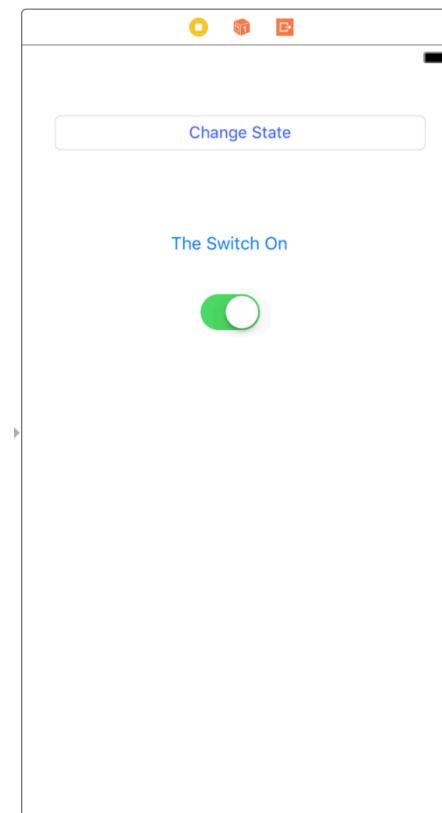
```
- (void)addLabel {
    UILabel *aLabel = [[UILabel alloc] initWithFrame:
                        CGRectMake(20, 200, 280, 80)];
    aLabel.numberOfLines = 0;
    aLabel.textColor = [UIColor blueColor];
    aLabel.backgroundColor = [UIColor clearColor];
    aLabel.textAlignment = NSTextAlignmentCenter;
    aLabel.text = @"This is a sample text\n of multiple lines";
    [self.view addSubview:aLabel];
}

- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    //The custom method to create our different types of button is called
    [self addLabel];
}
```

iOS «Switches»

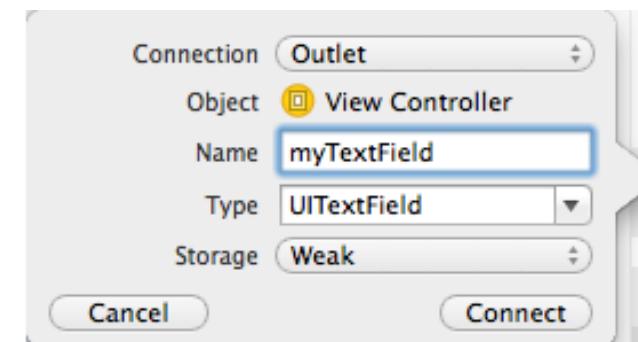
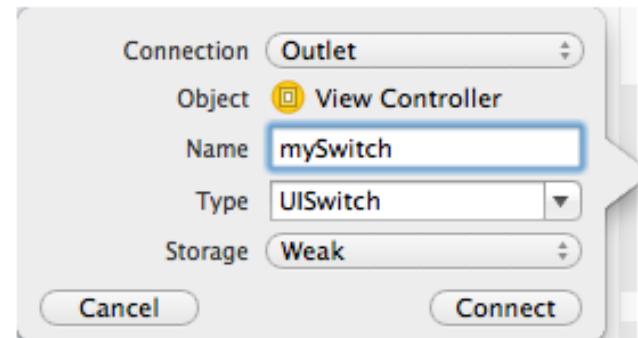
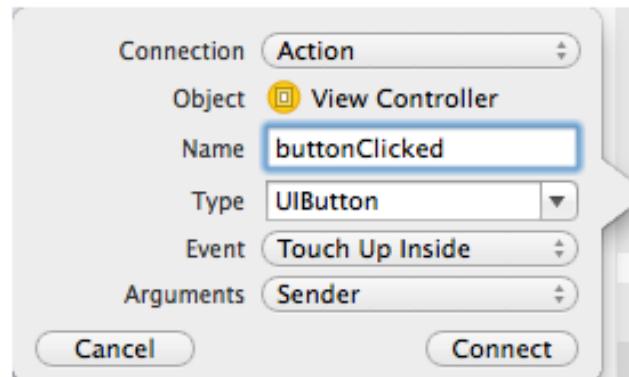
Utilisation des «Switches»

Les « Switches » sont utilisés pour basculer entre les états Marche et Arrêt



iOS «Switches»

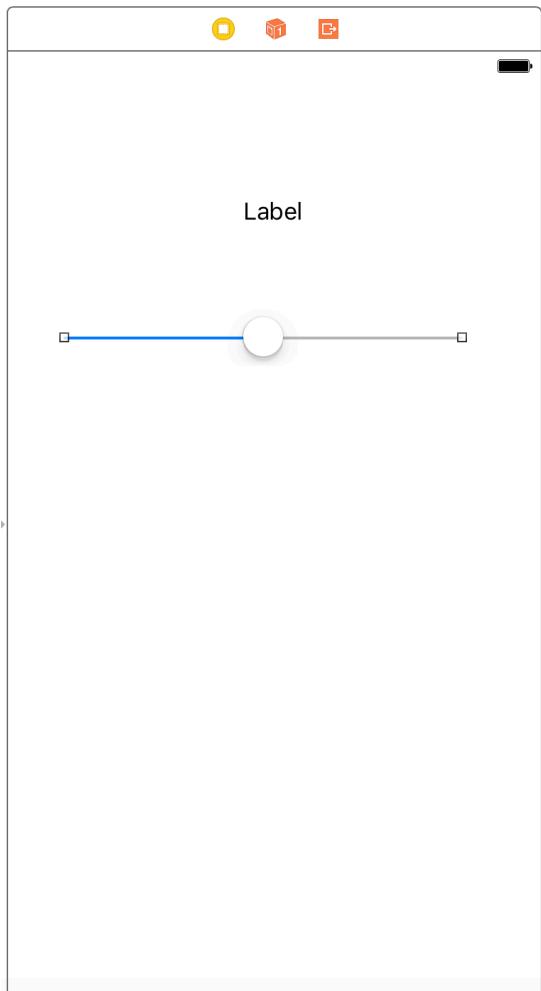
- Sélectionnez « **Assistant Editor** » et ouvrez **ViewController.m**. Ctrl et faites glisser du «Text Field» vers la section **@interface** et créez la «**Outlet**»
- Ensuite, **Ctrl et glisser** à partir du **Switch** et créer une «**Outlet**»
- **Ctrl et glissez** à partir du bouton et créez l'**action**



iOS «Switches»

```
8 #import "ViewController.h"
9
10 @interface ViewController ()
11 @property (weak, nonatomic) IBOutlet UITextField *myTextField;
12 @property (weak, nonatomic) IBOutlet UISwitch *mySwitch;
13 - (IBAction)buttonClicked:(id)sender;
14
15 @end
16
17 @implementation ViewController
18
19 - (void)viewDidLoad {
20     [super viewDidLoad];
21     // Do any additional setup after loading the view, typically from a nib.
22     [self.mySwitch addTarget:self
23         action:@selector(stateChanged:) forControlEvents:UIControlEventValueChanged];
24 }
25
26
27
28 - (void)didReceiveMemoryWarning {
29     [super didReceiveMemoryWarning];
30     // Dispose of any resources that can be recreated.
31 }
32
33
34 - (IBAction)buttonClicked:(UIButton *)sender {
35     if ([self.mySwitch isOn]) {
36         self.myTextField.text = @"The Switch is Off";
37         NSLog(@"Switch is on");
38         [self.mySwitch setOn:NO animated:YES];
39     } else {
40         self.myTextField.text = @"The Switch is On";
41         [self.mySwitch setOn:YES animated:YES];
42     }
43 }
44
45 - (void)stateChanged:(UISwitch *)switchState
46 {
47     if ([switchState isOn]) {
48         self.myTextField.text = @"The Switch is On";
49     } else {
50         self.myTextField.text = @"The Switch is Off";
51     }
52 }
```

iOS «Sliders»

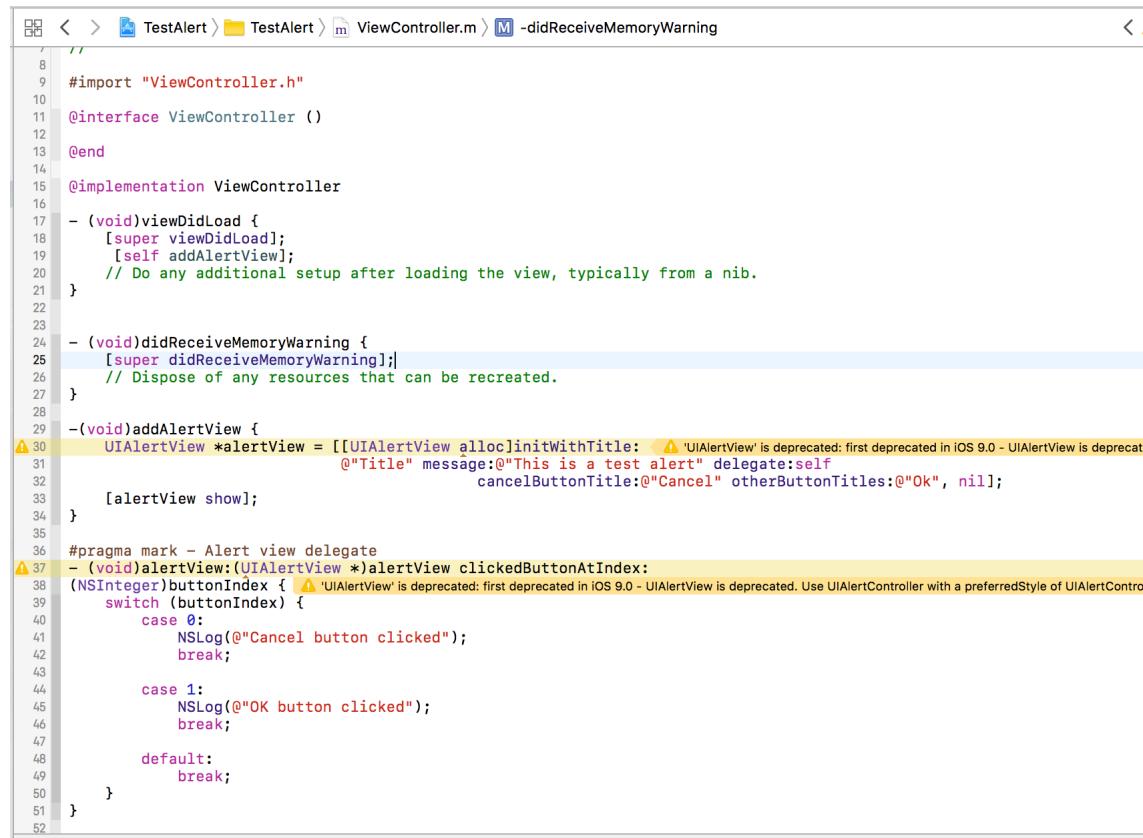


```
1 // ViewController.m
2 // SliderExampleTest
3 // Created by Tanmoy on 08/01/18.
4 // Copyright © 2018 Tanmoy. All rights reserved.
5
6
7 // Import ViewController.h
8 #import "ViewController.h"
9
10 @interface ViewController ()
11 @property (weak, nonatomic) IBOutlet UILabel *showInTheLabel;
12
13 @property (weak, nonatomic) IBOutlet UISlider *makeMySlider;
14 - (IBAction)sliderValueChanged:(id)sender;
15
16 @end
17
18 @implementation ViewController
19
20 - (void)viewDidLoad {
21     [super viewDidLoad];
22     // Do any additional setup after loading the view, typically from a nib.
23 }
24
25
26
27 - (void)didReceiveMemoryWarning {
28     [super didReceiveMemoryWarning];
29     // Dispose of any resources that can be recreated.
30 }
31
32
33
34 - (IBAction)sliderValueChanged:(id)sender {
35     // Set the label text to the value of the slider as it changes
36     self.showInTheLabel.text = [NSString stringWithFormat:@"%f", self.
37         makeMySlider.value];
38 }
39 @end
```

iOS «Alerts»

Utilisation des «Alerts»

Les alertes sont utilisées pour donner des informations importantes à l'utilisateur.



The screenshot shows a portion of the ViewController.m file from a Xcode project named TestAlert. The code demonstrates how to present an alert view. It includes imports, interface definitions, and implementations for viewDidLoad and didReceiveMemoryWarning methods. The addAlertView method creates an UIAlertView with a title, message, and two buttons (Cancel and OK). A warning message is displayed above the code indicating that UIAlertView is deprecated. The code also contains a switch statement to handle button clicks.

```
7 // 
8 // import "ViewController.h"
9 //
10 @interface ViewController : UIViewController
11 @end
12
13 @implementation ViewController
14
15 - (void)viewDidLoad {
16     [super viewDidLoad];
17     [self addAlertView];
18     // Do any additional setup after loading the view, typically from a nib.
19 }
20
21
22
23
24 - (void)didReceiveMemoryWarning {
25     [super didReceiveMemoryWarning];
26     // Dispose of any resources that can be recreated.
27 }
28
29 -(void)addAlertView {
30     UIAlertView *alertView = [[[UIAlertView alloc] initWithTitle:@"Title" message:@"This is a test alert" delegate:self cancelButtonTitle:@"Cancel" otherButtonTitles:@"OK", nil];
31     [alertView show];
32 }
33
34
35 #pragma mark - Alert view delegate
36 -(void)alertView:(UIAlertView *)alertView clickedButtonAtIndex:
37 (NSInteger)buttonIndex {
38     NSLog(@"UIAlertView is deprecated: first deprecated in iOS 9.0 - UIAlertView is deprecated");
39     NSLog(@"Title: %@", alertView.title);
40     NSLog(@"Message: %@", alertView.message);
41     NSLog(@"Buttons: %@", alertView.buttons);
42     NSLog(@"Delegate: %@", alertView.delegate);
43
44     switch (buttonIndex) {
45         case 0:
46             NSLog(@"Cancel button clicked");
47             break;
48
49         case 1:
50             NSLog(@"OK button clicked");
51             break;
52
53         default:
54             break;
55     }
56 }
```

Les diapositives sont faites à partir des matériaux à:
www.tutorialpoint.com

iOS «Image View»

Utilisation des «Image View»

La vue Image est utilisée pour afficher une seule image ou une séquence animée d'images.

```
#import "ViewController.h"
@interface ViewController : UIViewController
@end
@implementation ViewController
- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    // Load images
    NSArray *imageNames = @[@"win_1.png", @"win_2.png", @"win_3.png", @"win_4.png",
                           @"win_5.png", @"win_6.png", @"win_7.png", @"win_8.png",
                           @"win_9.png", @"win_10.png", @"win_11.png", @"win_12.png",
                           @"win_13.png", @"win_14.png", @"win_15.png", @"win_16.png"];
    NSMutableArray *images = [[NSMutableArray alloc] init];
    for (int i = 0; i < imageNames.count; i++) {
        [images addObject:[UIImage imageNamed:[imageNames objectAtIndex:i]]];
    }
    // Normal Animation
    UIImageView *animationImageView = [[UIImageView alloc] initWithFrame:CGRectMake(60, 95, 86, 193)];
    animationImageView.animationImages = images;
    animationImageView.animationDuration = 0.5;
    [self.view addSubview:animationImageView];
    [animationImageView startAnimating];
    // Slow motion animation
    UIImageView *slowAnimationImageView = [[UIImageView alloc] initWithFrame:CGRectMake(160, 95, 86, 193)];
    slowAnimationImageView.animationImages = images;
    slowAnimationImageView.animationDuration = 5;
    [self.view addSubview:slowAnimationImageView];
    [slowAnimationImageView startAnimating];
}
- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}
@end
```

Les diapositives sont faites à partir des matériaux à:

www.tutorialpoint.com

iOS Text View

```
- (void)addTextView {
    myTextView = [[UITextView alloc] initWithFrame:
    CGRectMake(10, 50, 300, 200)];
    [myTextView setText:@"Lorem ipsum dolor sit er elit lamet, consectetur
    cillum adipisicing pecu, sed do eiusmod tempor incididunt ut labore et
    dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation
    ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure
    dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat
    nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in
    culpa qui officia deserunt mollit anim id est laborum. Nam liber te
    conscient to factor tum poen legum odioque civiuda.
    Lorem ipsum dolor sit er elit lamet, consectetur cillum adipisicing pecu, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
    Excepteur sint occaecat cupidatat non proident, sunt in culpa
    qui officia deserunt mollit anim id est laborum. Nam liber te conscient
    to factor tum poen legum odioque civiuda."];
    myTextView.delegate = self;
    [self.view addSubview:myTextView];
}
```

ViewController.m

```
#pragma mark - Text View delegates

-(BOOL)textView:(UITextView *)textView shouldChangeTextInRange:
    (NSRange)range replacementText:(NSString *)text {
    if ([text isEqualToString:@"\n"]) {
        [textView resignFirstResponder];
    }
    return YES;
}

-(void)textViewDidBeginEditing:(UITextView *)textView {
    NSLog(@"Did begin editing");
}

-(void)textViewDidChange:(UITextView *)textView {
    NSLog(@"Did Change");
}

-(void)textViewDidEndEditing:(UITextView *)textView {
    NSLog(@"Did End editing");
}

-(BOOL)textViewShouldEndEditing:(UITextView *)textView {
    [textView resignFirstResponder];
    return YES;
}
```

Update `viewDidLoad` in `ViewController.m` as follows

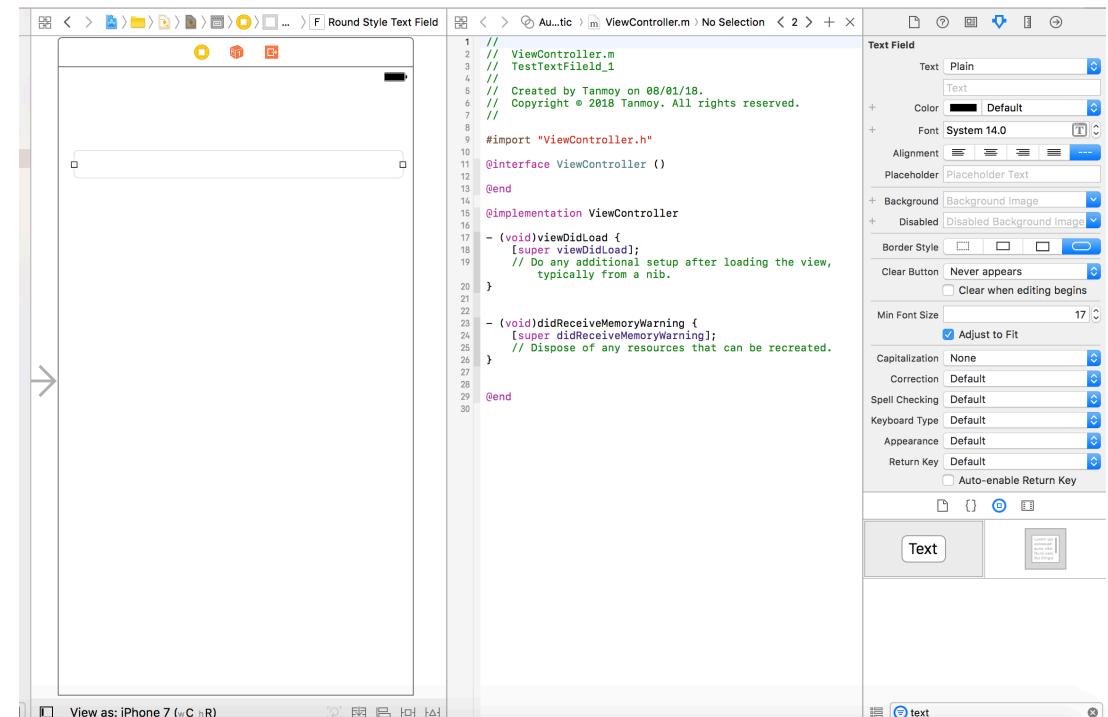
```
(void)viewDidLoad {
    [super viewDidLoad];
    [self addTextView];
}
```


iOS - Text Field

- Il permet à l'application d'obtenir **la saisie de l'utilisateur**

Propriétés importantes du champ Texte

- Texte par défaut qui s'affiche lorsqu'il n'y a pas de saisie utilisateur
- Texte normal
- Type de correction automatique
- Type de clavier
- Type de clé de retour
- Mode touche Effacer
- Alignement
- Delegates

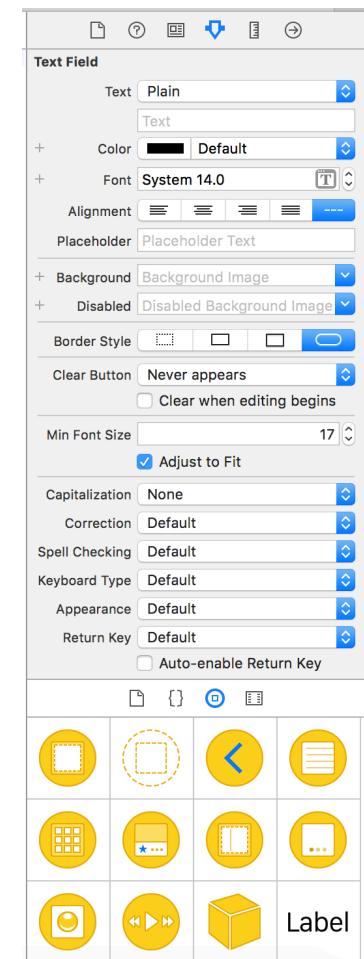
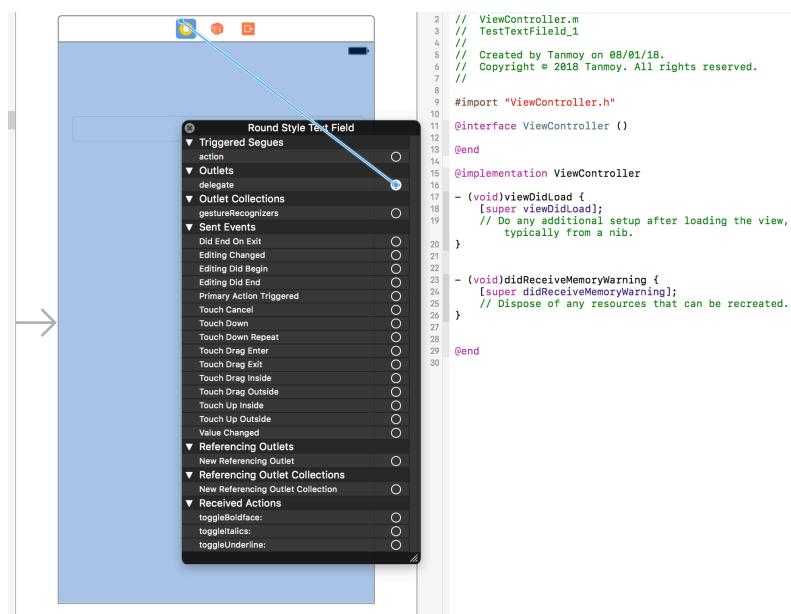


Les diapositives sont faites à partir des matériaux à:

www.tutorialpoint.com

iOS - Text Field

- Vous pouvez modifier les propriétés du champ de texte (côté droit de la fenêtre).
- Définir le «**Delegate**» dans le «**Interface Builder**» en cliquant avec le bouton droit de la souris sur **UIElement** et le connecter.



Eléments UI

Notre approche

Nous allons créer une application iOS simple et l'utiliser pour expliquer certains des éléments de l'interface utilisateur.

- **Etape 1** - Créer une application basée sur la vue comme nous l'avons fait dans notre première application iOS.
- **Étape 2** - Nous mettrons seulement à jour les fichiers `ViewController.h` et `ViewController.m`
- **Etape 3** - Ensuite, nous ajoutons une méthode à notre fichier `ViewController.m` spécifique pour créer l'élément UI.
- **Étape 4** - Nous appellerons cette méthode dans notre méthode `viewDidLoad`.
- **Étape 5** - Les lignes de code importantes ont été expliquées dans le code avec un commentaire d'une seule ligne au-dessus de ces

iOS - Accéléromètre

Étape-1: Créer une simple « View Based Application »

Étape-2: Ajoutez 3 «Labels» dans **ViewController.xib** et créez **ibOutlets** en les nommant **xlabel**, **ylabel** et **zlabel**

Étape-3: Mettre à jour **ViewController.h** comme suit :

```
#import <UIKit/UIKit.h>

@interface ViewController : UIViewController<UIAccelerometerDelegate> {
    IBOutlet UILabel *xlabel;
    IBOutlet UILabel *ylabel;
    IBOutlet UILabel *zlabel;
}
@end
```

Étape-4: Mettre à jour **ViewController.m** comme suit : →

```
#import "ViewController.h"

@interface ViewController ()
```

Implementation

```
@implementation ViewController

- (void)viewDidLoad {
    [super viewDidLoad];
    [[UIAccelerometer sharedAccelerometer] setDelegate:self];
    //Do any additional setup after loading the view,typically from a nib
}

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

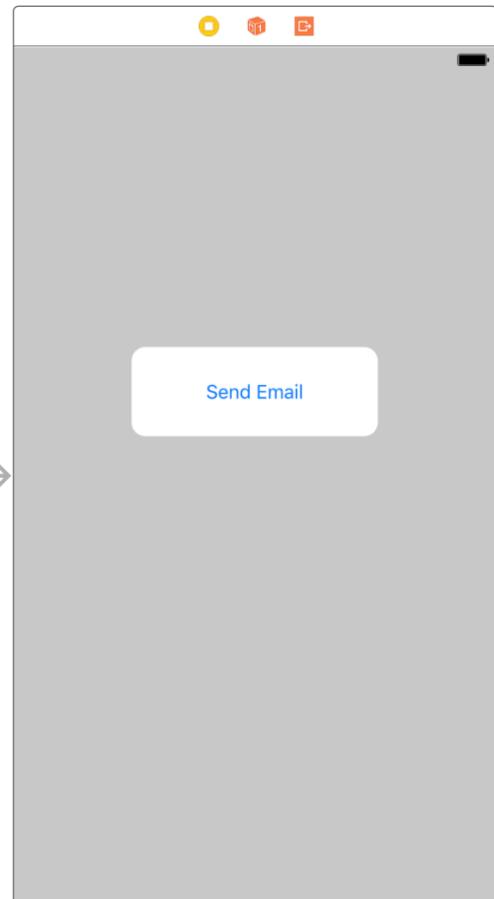
- (void)accelerometer:(UIAccelerometer *)accelerometer didAccelerate:(UIAcceleration *)acceleration {
    [ xlabel setText:[NSString stringWithFormat:@"%f", acceleration.x]];
    [ ylabel setText:[NSString stringWithFormat:@"%f", acceleration.y]];
    [ zlabel setText:[NSString stringWithFormat:@"%f", acceleration.z]];
}
```

Output



Les diapositives sont faites à partir des matériaux à: www.tutorialpoint.com

Envoyer E-Mail



- Ajouter un bouton
- Crée une action dans «**ViewController.h**»
- Importez le »**MessageUI/MessageUI.h**« et implémentez le delegate "**MFMailComposeViewControllerDelegate**" dans «**ViewController.h**»

```
@interface ViewController : UIViewController<MFMailComposeViewControllerDelegate> {
    MFMailComposeViewController *mailComposer;
    -(IBAction)sendEmail:(id)sender;
}
@end
```

Envoyer E-Mail

- Dans la fonction de button, mettez ces liens de code
- Xcode n'a aucune idée de ce qu'est le **"MFMailComposeViewController"**
- Bien que **"MFMailComposeViewController"** soit un contrôleur intégré dans iOS SDK
- il est de la responsabilité des développeurs d'intégrer le **framework** nécessaire dans votre projet.

```
- (void)sendMail:(id)sender {
    // Email Subject
    NSString *emailTitle = @"Test Email";
    // Email Content
    NSString *messageBody = @"iOS programming is so fun!";
    // To address
    NSArray *toRecipients = [NSArray arrayWithObject:@"support@appcoda.com"];

    MFMailComposeViewController *mc = [[MFMailComposeViewController alloc] init];
    mc.mailComposeDelegate = self;
    [mc setSubject:emailTitle];
    [mc setMessageBody:messageBody isHTML:NO];
    [mc setToRecipients:toRecipients];

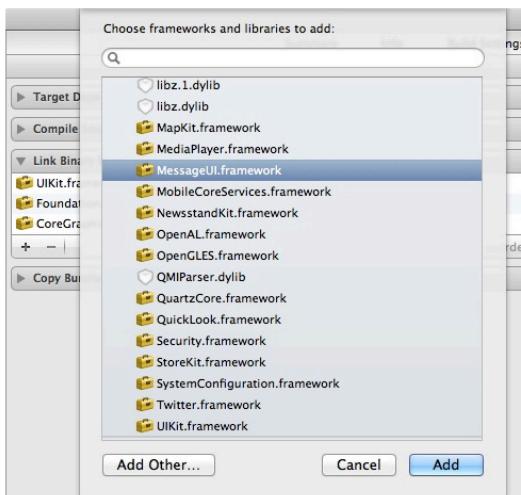
    // Present mail view controller on screen
    [self presentViewController:mc animated:YES completion:NULL];
}
```

- Ci-dessus définit l'sujet, le contenu et les destinataires du courriel
- La classe **MFMailComposeViewController** fournit une interface standard qui gère l'édition et l'envoi d'un e-mail
- Utiliser **ViewController** pour afficher une vue email standard à l'intérieur de votre application
- Nous remplissons les champs de cette vue avec les valeurs initiales
- Qui compris le courriel du destinataire, l'sujet et le corps du texte
- Code appelle le **presentViewController** pour afficher l'interface mail à l'écran.

Envoyer E-Mail

```
- (void) mailComposeController:(MFMailComposeViewController *)controller didFinishWithResult:(MFMailComposeResult)result
error:(NSError *)error
{
    switch (result)
    {
        case MFMailComposeResultCancelled:
            NSLog(@"Mail cancelled");
            break;
        case MFMailComposeResultSaved:
            NSLog(@"Mail saved");
            break;
        case MFMailComposeResultSent:
            NSLog(@"Mail sent");
            break;
        case MFMailComposeResultFailed:
            NSLog(@"Mail sent failure: %@", [error localizedDescription]);
            break;
        default:
            break;
    }

    // Close the Mail Interface
    [self dismissViewControllerAnimated:YES completion:NULL];
}
```



- Le "**didFinishWithResult:**" est une méthode du protocole **MFMailComposeViewControllerDelegate**.
- Cette méthode est automatiquement appelée lorsque l'interface mail est fermée (par ex. l'utilisateur annule l'opération).
- Le paramètre de résultat vous indique le code résultat lorsque l'interface de composition du courrier est rejetée.
- Nous enregistrons simplement l'état dans cet exemple.
- Dans le cas d'une application réelle, vous devez fournir un traitement spécial pour le code de résultat (par exemple, afficher une alerte lorsque l'application n'envoie pas le courriel)
- la dernière ligne est pour supprimer l'interface de composition du courrier



Envoyer E-Mail

- Vous pouvez facilement activer le support de l'email HTML en changeant le paramètre "isHTML" dans la méthode "setMessageBody" de "N" à "Y".
- Essayez de changer le code dans la méthode "showEmail" pour ce qui suit:

```
-(void)sendMail:(id)sender {
    // Email Subject
    NSString *emailTitle = @"Test Email";
    // Email Content
    NSString *messageBody = @"<h1>Learning iOS Programming!</h1>"; // Change the message body to HTML
    // To address
    NSArray *toRecipients = [NSArray arrayWithObject:@"support@appcoda.com"];

    MFMailComposeViewController *mc = [[MFMailComposeViewController alloc] init];
    mc.mailComposeDelegate = self;
    [mc setSubject:emailTitle];
    [mc setMessageBody:messageBody isHTML:YES];
    [mc setToRecipients:toRecipients];

    // Present mail view controller on screen
    [self presentViewController:mc animated:YES completion:NULL];
}
```

- Compiler et relancer votre application. Avec un simple changement, l'application offre maintenant le support HTML!

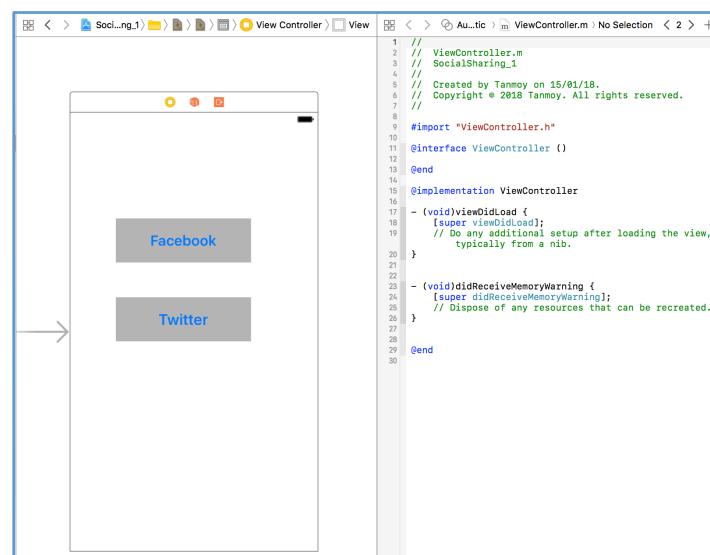
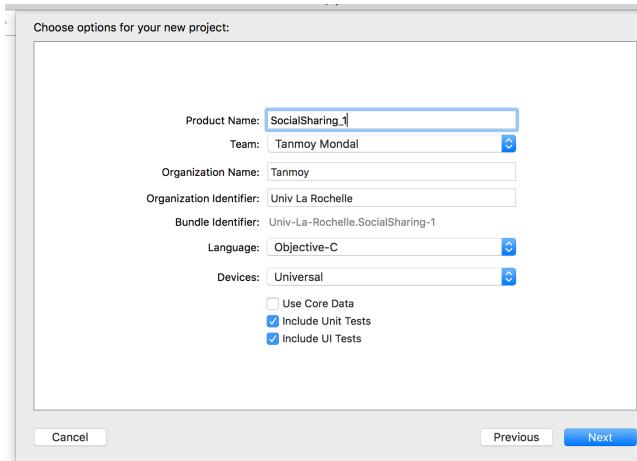
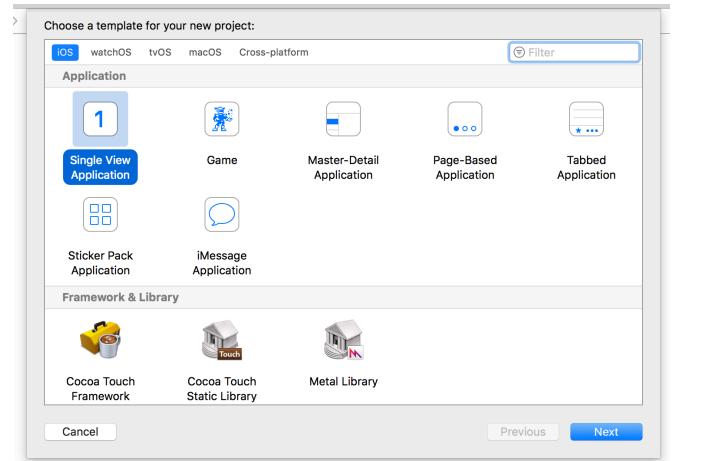


Réseaux Social (Facebook et Twitter)

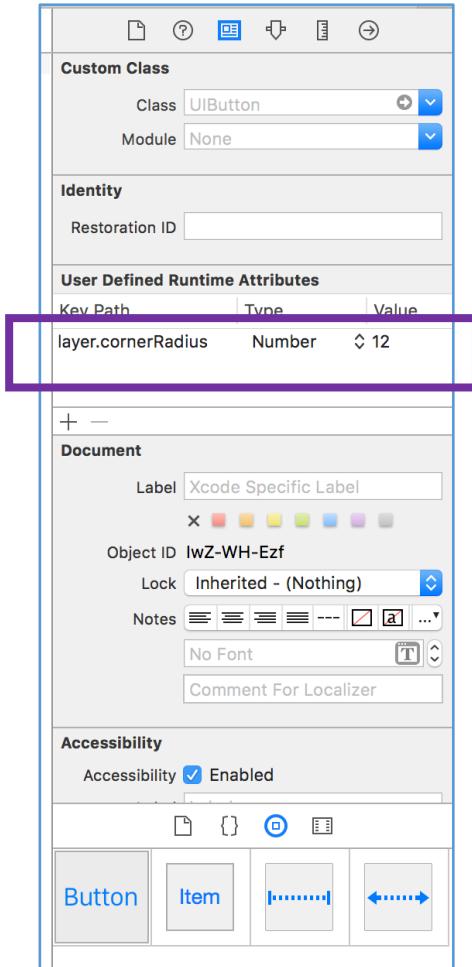


- Twitter est ajouté dans iOS 5, Facebook dans iOS 6
- iOS 6 introduit un nouveau **FrameWork** appelé "**Social Framework**"
- Permet d'intégrer votre application à tous les services de réseautage
- il supporte **Facebook**, **Twitter** et **Sina Weibo**
- Le framework est livré avec une classe très pratique appelée **SLComposeViewController**
- La classe **SLComposeViewController** présente un **ViewController** standard pour les utilisateurs de **tweet** ou de poste **Facebook**
- Permet également de prérégler le texte initial, de joindre des images et d'ajouter des URL au message

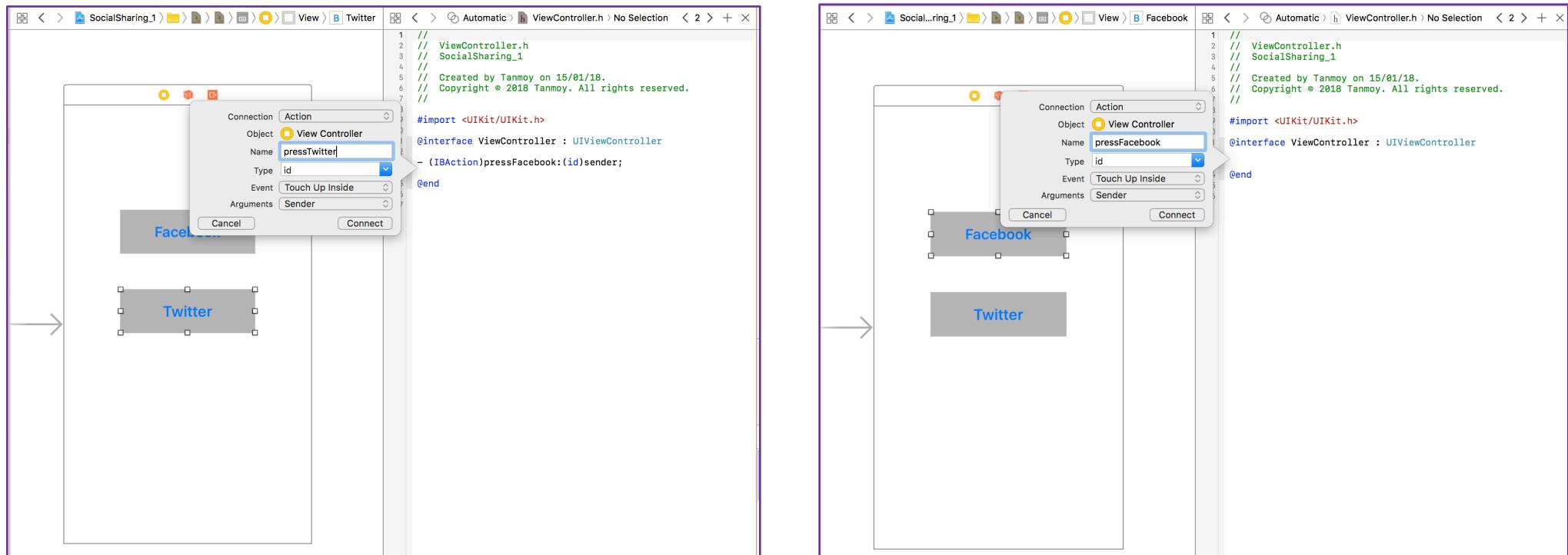
Réseaux Social (Facebook et Twitter)



Les diapositives sont faites à partir des matériaux à:
www.tutorialpoint.com



Réseaux Social (Facebook et Twitter)



```
9 #import <UIKit/UIKit.h>
10
11 @interface ViewController : UIViewController
12
13 - (IBAction)pressFacebook:(id)sender;
14 - (IBAction)pressTwitter:(id)sender;
15
16 @end
```

Les diapositives sont faites à partir des matériaux à:

www.tutorialpoint.com

Réseaux Social (Facebook et Twitter)

Ajouter dans «**ViewController.h**», ce fichier

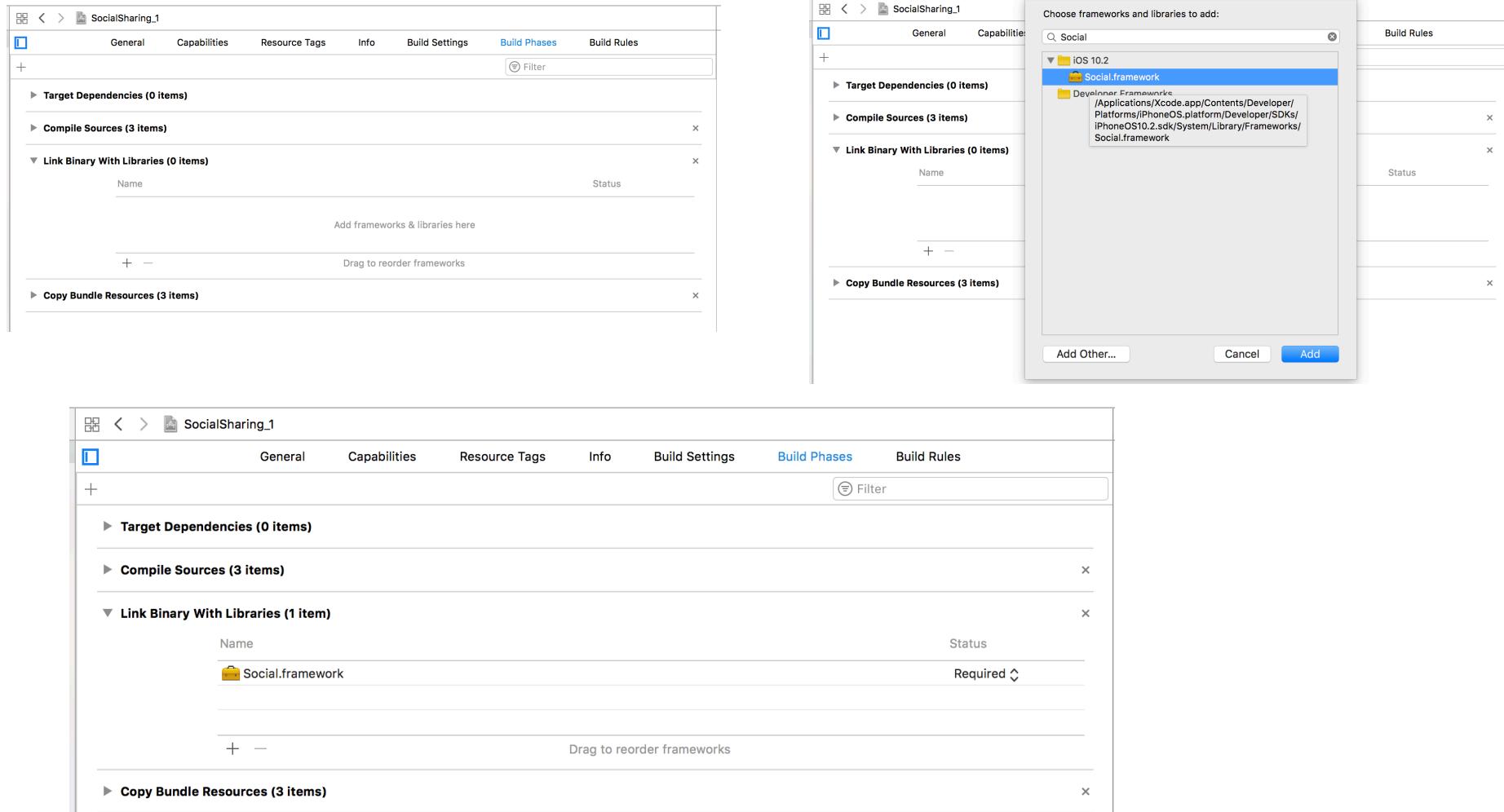
```
#import <UIKit/UIKit.h>
#import <Social/Social.h>
```

Changer le fonction en «**ViewController.m**», comme ça

```
- (IBAction)pressTwitter:(id)sender {
    if ([SLComposeViewController isAvailableForServiceType:SLServiceTypeTwitter])
    {
        SLComposeViewController *tweetSheet = [SLComposeViewController
                                              composeViewControllerForServiceType:SLServiceTypeTwitter];
        [tweetSheet setInitialText:@"Great fun to learn iOS programming at appcoda.com!"];
        [self presentViewController:tweetSheet animated:YES completion:nil];
    }
}
```

- La méthode "**isAvailableForServiceType**" pour vérifier si le service Twitter est accessible
- Les utilisateurs ne peuvent pas accéder au service Twitter est qu'ils ne se sont pas connectés au service
- Nous créons alors une instance du **SLComposeViewController** pour le service Twitter
- Nous invoquons **presentViewController: tweetSheet** pour faire apparaître le compositeur Twitter

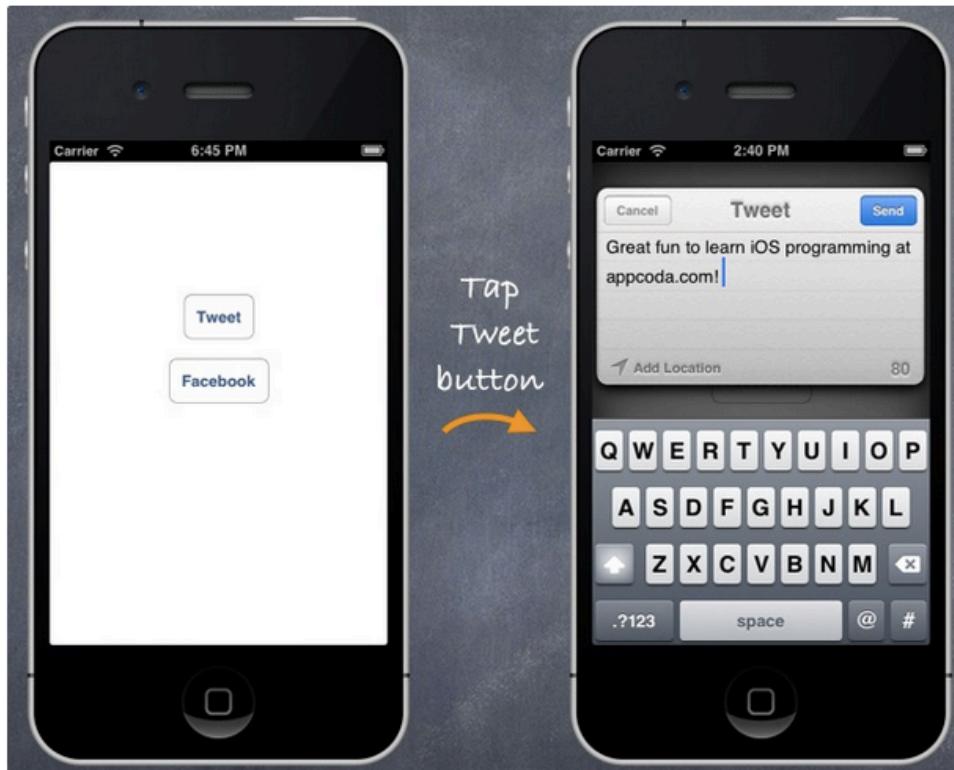
Réseaux Social (Facebook et Twitter)



LES DIAPPOSITIVES SONT TAKES A PARTIR DES MATERIAUX DE

www.tutorialpoint.com

Réseaux Social (Facebook et Twitter)



Réseaux Social (Facebook et Twitter)

Ajouter supporte de Facebook

```
- (IBAction)pressFacebook:(id)sender {
    if([SLComposeViewController isAvailableForServiceType:SLServiceTypeFacebook]) {
        SLComposeViewController *controller = [SLComposeViewController
            composeViewControllerForServiceType:SLServiceTypeFacebook];

        [controller setInitialText:@"First post from my iPhone app"];
        [self presentViewController:controller animated:YES completion:nil];
    }
}
```

-- La seule modification est le type de service.

-- Au lieu d'utiliser **SLServiceTypeTwitter**, nous conseillons à **SLComposeViewController** d'utiliser **SLServiceTypeFacebook**.



Les diapositives sont faites à partir des matériaux à:
www.tutorialpoint.com

Réseaux Social (Facebook et Twitter)

Ajouter supporte de Facebook

- Avec **SLComposeViewController** vous pouvez télécharger facilement les images
- Téléchargez cette image et ajoutez-la au projet Xcode. (Vous pouvez utiliser n'importe quelle image mais assurez-vous de l'appeler "**socialsharing-facebook-image.jpg**")
- Vous pouvez simplement utiliser la méthode "**addImage**" pour attacher une image et la méthode "**addURL**" pour inclure un lien

```
[controller addURL:[NSURL URLWithString:@"http://www.appcoda.com"]];
[controller addImage:[UIImage imageNamed:@"socialsharing-facebook-image.jpg"]]
```

```
- (IBAction)pressFacebook:(id)sender {
    if([SLComposeViewController isAvailableForServiceType:SLServiceTypeFacebook]) {

        SLComposeViewController *controller = [SLComposeViewController
                                              composeViewControllerForServiceType:SLServiceTypeFacebook];

        [controller setInitialText:@"First post from my iPhone app"];
        [controller addURL:[NSURL URLWithString:@"http://www.appcoda.com"]];
        [controller addImage:[UIImage imageNamed:@"socialsharing-facebook-image.jpg"]];

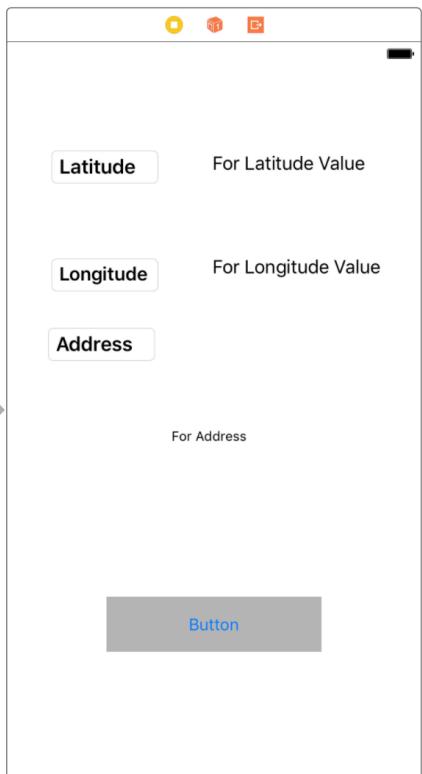
        [self presentViewController:controller animated:YES completion:Nil];
    }
}
```



Les diapositives sont faites à partir des matériaux à:
www.tutorialpoint.com

iOS - Gestion des emplacements

- Concevoir l'écran de cette manière



- Ajouter le bibliothèque ici

Link Binary With Libraries (1 item)

Name	Status
CoreLocation.framework	Required

+ - Drag to reorder frameworks

- le code de **ViewController.h** devrait ressembler au suivant:

```
9 #import <UIKit/UIKit.h>
10 #import <CoreLocation/CoreLocation.h>
11
12 @interface ViewController : UIViewController <CLLocationManagerDelegate>
13
14 @property (weak, nonatomic) IBOutlet UILabel *showLatValue;
15 @property (weak, nonatomic) IBOutlet UILabel *showLongValue;
16 @property (weak, nonatomic) IBOutlet UILabel *typeAddressHere;
17
18 - (IBAction)clickOnButton:(id)sender;
19 @end
```

- CoreLocation** utilise le modèle de délégation
- Pour travailler avec le framework **CoreLocation**, notre ViewController doit être conforme au protocole **CLLocationManagerDelegate**
- Ce protocole définit les méthodes utilisées pour recevoir les mises à jour de localisation et d'en-tête d'un objet **CLLocationManager**

iOS - Gestion des emplacements

```
 9 #import <UIKit/UIKit.h>
10 #import <CoreLocation/CoreLocation.h>
11
12 @interface ViewController : UIViewController <CLLocationManagerDelegate>
13
○ 14 @property (weak, nonatomic) IBOutlet UILabel *showLatValue;
○ 15 @property (weak, nonatomic) IBOutlet UILabel *showLongValue;
○ 16 @property (weak, nonatomic) IBOutlet UILabel *typeAddressHere;
17
○ 18 - (IBAction)clickOnButton:(id)sender;
19 @end
20
```

- Pour faire connaître au ViewController le **CLLocationManagerDelegate**, vous importez d'abord le fichier d'en-tête correspondant.
- Dans le "**MyLocationViewController. h**", ajoutez l'instruction **#import** et implémentez le "**CLLocationManagerDelegate** »

iOS - Gestion des emplacements

- Dans « **ViewController.m** ». Déclarez une variable d'instance et nommez-la comme **locationManager**.

```
@implementation ViewController{  
    CLLocationManager *locationManager;  
}
```

- Le **CLLocationManager** est l'objet qui vous fournit les données de localisation

```
- (void)viewDidLoad {  
    [super viewDidLoad];  
  
    locationManager = [[CLLocationManager alloc] init];  
}
```

- Dans la méthode **viewDidLoad**, instanciez l'objet **CLLocationManager**

```
③ 35 - (IBAction)clickOnButton:(id)sender {  
  36     locationManager.delegate = self;  
  37     locationManager.distanceFilter = kCLLocationAccuracyNone;  
  38     locationManager.desiredAccuracy = kCLLocationAccuracyBest;  
  39     if([[UIDevice currentDevice] systemVersion] floatValue] >= 8.0)  
  40         [locationManager requestWhenInUseAuthorization];  
  41  
  42     [locationManager startUpdatingLocation];  
  43 }  
...  
44 }
```

iOS - Gestion des emplacements

- Les données de localisation sont rapportées à votre application via l'objet délégué associé du gestionnaire de localisation.
- Ici, nous attribuons MyLocationViewController comme objet délégué.
- Toutes les mises à jour de la localisation seront envoyées au délégué
- Pour capturer l'événement de localisation, nous devons implémenter les méthodes de délégation définies dans le protocole

```
43 #pragma mark - CLLocationManagerDelegate
44
45 - (void)locationManager:(CLLocationManager *)manager didFailWithError:(NSError *)error
46 {
47     NSLog(@"didFailWithError: %@", error);
48     UIAlertView *errorAlert = [[UIAlertView alloc]
49                               initWithTitle:@"Error" message:@"Failed to Get Your Location" delegate:
50                               nil cancelButtonTitle:@"OK" otherButtonTitles:nil];
51     [errorAlert show];
52 }
53
54 - (void)locationManager:(CLLocationManager *)manager didUpdateToLocation:(CLLocation *)newLocation
55 fromLocation:(CLLocation *)oldLocation
56 {
57     NSLog(@"didUpdateToLocation: %@", newLocation);
58     CLLocation *currentLocation = newLocation;
59
60     if (currentLocation != nil) {
61         _showLatValue.text = [NSString stringWithFormat:@"%.8f", currentLocation.coordinate.longitude];
62         _showLongValue.text = [NSString stringWithFormat:@"%.8f", currentLocation.coordinate.latitude];
63     }
64 }
```

iOS - Gestion des emplacements

- Après avoir lancé l'application, cliquez sur le bouton "Get My Location".
- Pour la toute première fois que vous lancez l'application, l'alerte suivante vous invite à demander l'accès au gestionnaire d'emplacement:
- N'oubliez pas de l'accepter, sinon l'application ne pourra pas accéder au service de localisation.
- Oops! L'application affiche une erreur même si vous touchez le bouton OK.



Failed to get the current location



sont faites à partir des matériaux à:
www.tutorialpoint.com

iOS - Gestion des emplacements

```
 9 #import "ViewController.h"
10
11 @interface ViewController : UIViewController
12
13 @end
14
15 @implementation ViewController
16 CLLocationManager *locationManager;
17 CLGeocoder *geocoder;
18 CLPlacemark *placemark;
19
20
21 - (void)viewDidLoad {
22     [super viewDidLoad];
23     // Do any additional setup after loading the view, typically from a nib.
24     locationManager = [[CLLocationManager alloc] init];
25     geocoder = [[CLGeocoder alloc] init];
26 }
27
28
29 - (void)didReceiveMemoryWarning {
30     [super didReceiveMemoryWarning];
31     // Dispose of any resources that can be recreated.
32 }
```

- La classe **CLGeocoder** fournit des services de conversion entre une coordonnée GPS et l'adresse lisible par l'utilisateur de cette coordonnée
- En spécifiant la latitude et la longitude d'un emplacement donné, vous pouvez utiliser **CLGeocoder** pour trouver une adresse lisible par l'utilisateur

iOS - Gestion des emplacements

```
46 #pragma mark - CLLocationManagerDelegate
47
48 - (void)locationManager:(CLLocationManager *)manager didFailWithError:(NSError *)error
49 {
50     NSLog(@"didFailWithError: %@", error);
51     UIAlertView *errorAlert = [[UIAlertView alloc] initWithTitle:@"Error" message:@"Failed to Get Your Location" delegate:nil cancelButtonTitle:@"OK" otherButtonTitles:nil];
52     [errorAlert show];
53 }
54
55
56 - (void)locationManager:(CLLocationManager *)manager didUpdateToLocation:(CLLocation *)newLocation fromLocation:
57 (CLLocation *)oldLocation
58 {
59     NSLog(@"didUpdateToLocation: %@", newLocation);
60     CLLocation *currentLocation = newLocation;
61
62     if (currentLocation != nil) {
63         _showLatValue.text = [NSString stringWithFormat:@"%.8f", currentLocation.coordinate.longitude];
64         _showLongValue.text = [NSString stringWithFormat:@"%.8f", currentLocation.coordinate.latitude];
65     }
66
67     [locationManager stopUpdatingLocation];
68
69 // Reverse Geocoding
70 NSLog(@"Resolving the Address");
71 [geocoder reverseGeocodeLocation:currentLocation completionHandler:^(NSArray *placemarks, NSError *error) {
72     NSLog(@"Found placemarks: %@", error);
73     if (error == nil && [placemarks count] > 0) {
74         placemark = [placemarks lastObject];
75         _typeAddressHere.text = [NSString stringWithFormat:@"%@ %@\n%@ %@\n%@",
76             placemark.subThoroughfare, placemark.thoroughfare,
77             placemark.postalCode, placemark.locality,
78             placemark.administrativeArea,
79             placemark.country];
80     } else {
81         NSLog(@"%@", error.debugDescription);
82     }
83 }];
84 }
```

- Le résultat (c'est-à-dire l'adresse) renvoyé par **CLGeocoder** est sauvegardé dans un objet **CLPlacemark**

- Nous utilisons la méthode **"reverseGeocodeLocation"** pour traduire les données de localisation en une adresse lisible
- Les opérations de géocodage n'ont pas lieu sur le device
- La méthode soumet plutôt les données au serveur de géocodage dans le nuage
- vous devez fournir le gestionnaire qui contient le code à exécuter après la résolution de l'adresse
- A la fin d'une requête de géocodage, la commande **completionHandler** sera appelée automatiquement.
- L'adresse résolue est sauvegardée dans le tableau **CLPlacemark**. Les données de repère comprennent des informations telles que le pays, l'état, la ville et l'adresse municipale.
- Il suffit donc de sélectionner l'objet **CLPlacemark** dans le tableau et d'afficher l'adresse dans l'étiquette d'adresse.

iOS - Gestion des emplacements

```
46 #pragma mark - CLLocationManagerDelegate
47
48 - (void)locationManager:(CLLocationManager *)manager didFailWithError:(NSError *)error
49 {
50     NSLog(@"didFailWithError: %@", error);
51     UIAlertView *errorAlert = [[UIAlertView alloc] initWithTitle:@"Error" message:@"Failed to Get Your Location" delegate:nil cancelButtonTitle:@"OK" otherButtonTitles:nil];
52     [errorAlert show];
53 }
54
55
56 - (void)locationManager:(CLLocationManager *)manager didUpdateToLocation:(CLLocation *)newLocation fromLocation:
57 (CLLocation *)oldLocation
58 {
59     NSLog(@"didUpdateToLocation: %@", newLocation);
60     CLLocation *currentLocation = newLocation;
61
62     if (currentLocation != nil) {
63         _showLatValue.text = [NSString stringWithFormat:@"%.8f", currentLocation.coordinate.longitude];
64         _showLongValue.text = [NSString stringWithFormat:@"%.8f", currentLocation.coordinate.latitude];
65     }
66
67     [locationManager stopUpdatingLocation];
68
69 // Reverse Geocoding
70 NSLog(@"Resolving the Address");
71 [geocoder reverseGeocodeLocation:currentLocation completionHandler:^(NSArray *placemarks, NSError *error) {
72     NSLog(@"Found placemarks: %@", error: %@, placemarks, error);
73     if (error == nil && [placemarks count] > 0) {
74         placemark = [placemarks lastObject];
75         _typeAddressHere.text = [NSString stringWithFormat:@"%@ %@\n%@ %@\n%@", placemark.subThoroughfare, placemark.thoroughfare,
76                             placemark.postalCode, placemark.locality,
77                             placemark.administrativeArea,
78                             placemark.country];
79     } else {
80         NSLog(@"%@", error.debugDescription);
81     }
82 }];
83 }
```

- Le problème, c'est que la méthode va rapporter les données de localisation en continu.
- Il continue juste à avancer et donne des mises à jour de localisation chaque seconde, même l'emplacement est inchangé.
- Jetez un coup d'oeil à la sortie de la zone de débogage. Vous devriez avoir quelque chose de similaire à ce qui suit:

iOS - Gestion des emplacements

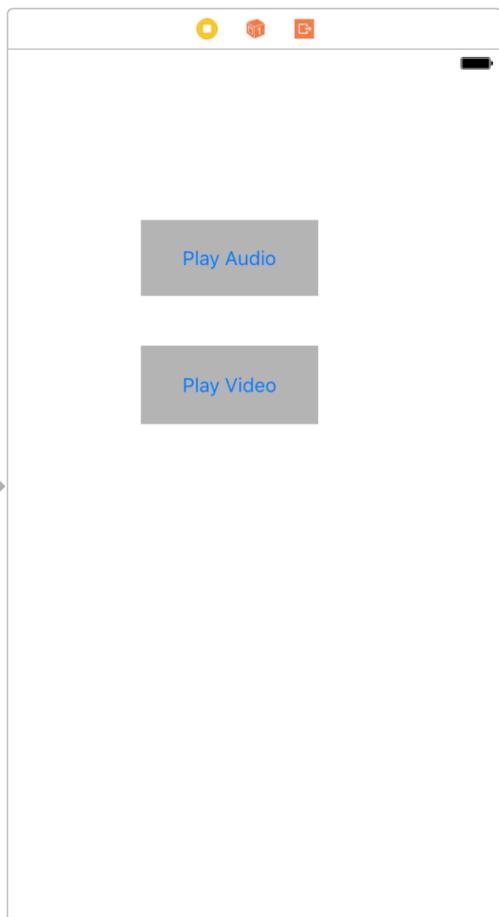
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>NSLocationWhenInUseUsageDescription</key>
    <string>I want to get your location Information in background</string>

    <key>UIBackgroundModes</key>
    <array>
        <string>location</string>
    </array>

    <key>CFBundleDevelopmentRegion</key>
    <string>en</string>
    <key>CFBundleExecutable</key>
    <string>$(EXECUTABLE_NAME)</string>
    <key>CFBundleIdentifier</key>
    <string>$(PRODUCT_BUNDLE_IDENTIFIER)</string>
    <key>CFBundleInfoDictionaryVersion</key>
    <string>6.0</string>
    <key>CFBundleName</key>
    <string>$(PRODUCT_NAME)</string>
    . . . . .
```

iOS - Audio et vidéo

- Concevoir l'écran de cette manière



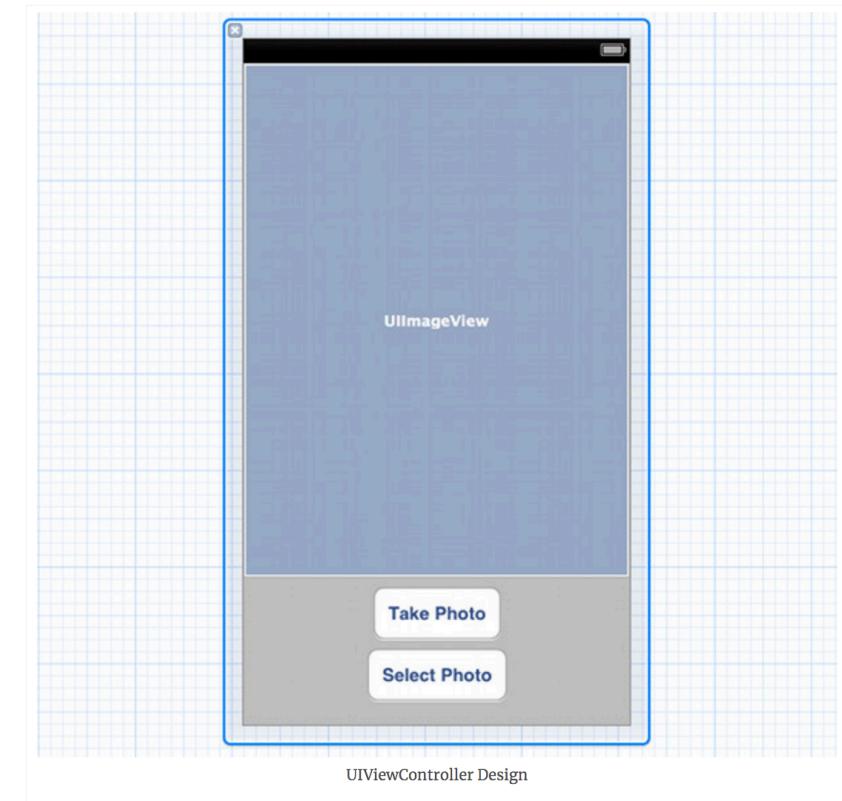
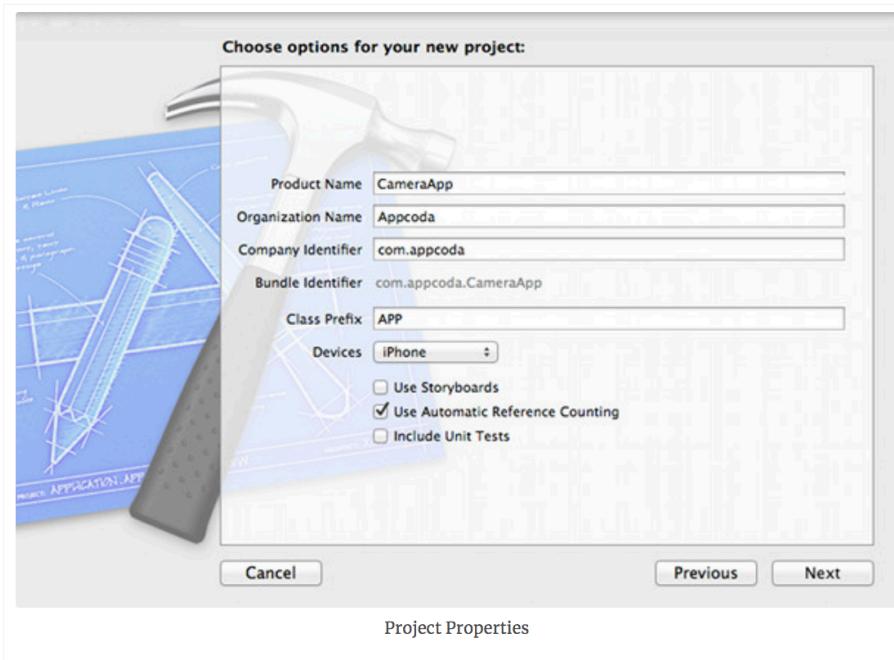
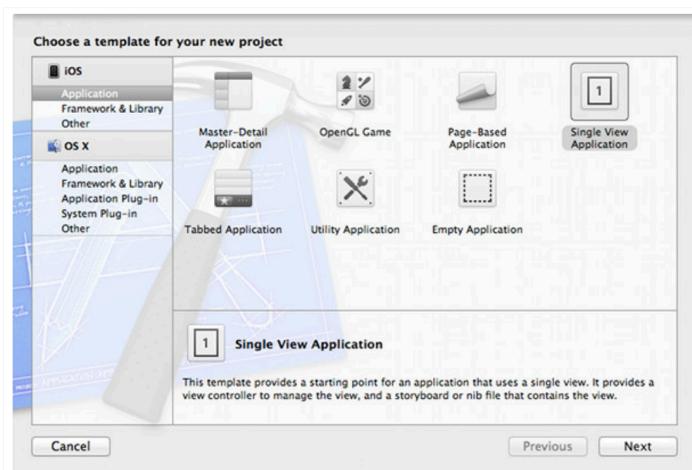
```
9  #import <UIKit/UIKit.h>
10 #import <AVFoundation/AVFoundation.h>
11 #import <MediaPlayer/MediaPlayer.h>
12 #import <AVKit/AVKit.h>
13
14 @interface ViewController : UIViewController{
15     AVAudioPlayer *audioPlayer;
16     AVPlayer *moviePlayer;
17 }
18 @property (strong, nonatomic) AVPlayerViewController *playerViewController;
19 - (IBAction)playAudioButton:(id)sender;
20 - (IBAction)playVedioButton:(id)sender;
21
22
23
24
```

iOS - Audio et vidéo

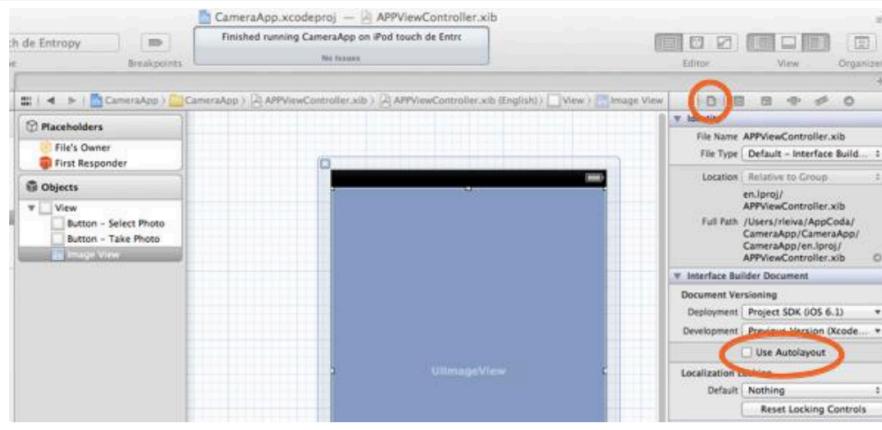
```
9 #import "ViewController.h"
10
11 @interface ViewController : UIViewController {
12     NSURL *videoURL;
13 }
14 @end
15
16 @implementation ViewController
17 @synthesize playerViewController;
18 - (void)viewDidLoad {
19     [super viewDidLoad];
20     // Do any additional setup after loading the view, typically from a nib.
21 }
22
23
24 - (void)didReceiveMemoryWarning {
25     [super didReceiveMemoryWarning];
26     // Dispose of any resources that can be recreated.
27 }
28
```

iOS - Audio et vidéo

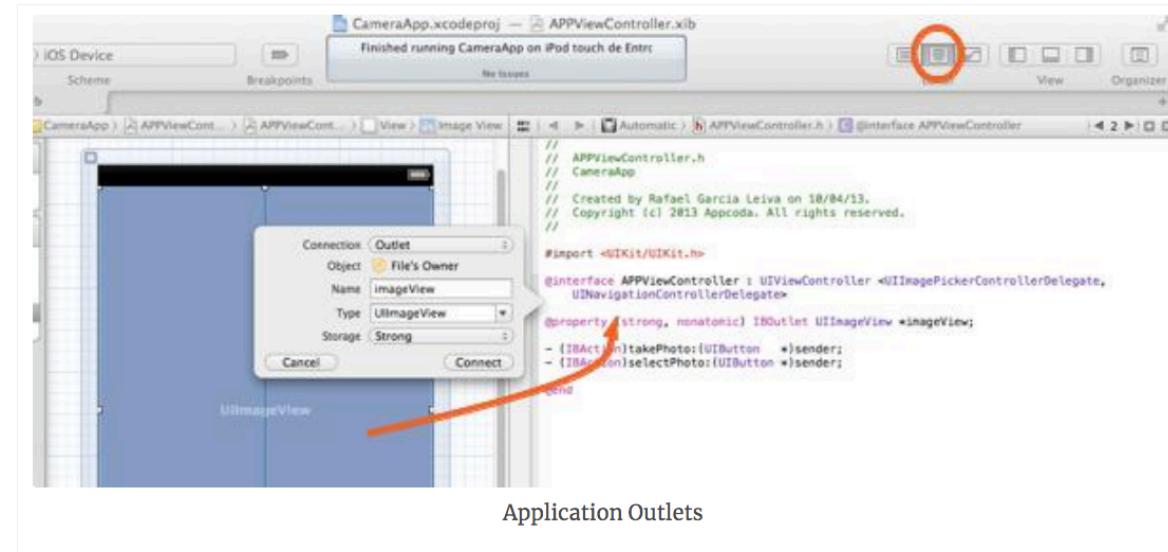
```
④ 30 - (IBAction)playAudioButton:(id)sender {
  31     NSString *path = [[NSBundle mainBundle]
  32                         pathForResource:@"audioTest" ofType:@"mp3"];
  33     audioPlayer = [[AVAudioPlayer alloc] initWithContentsOfURL:
  34                         [NSURL fileURLWithPath:path] error:NULL];
  35     [audioPlayer play];
  36 }
  37
④ 38 - (IBAction)playVedioButton:(id)sender {
  39     NSString *fullpath = [[self documentsDirectory]
  40                           stringByAppendingPathComponent:@"yourdate.3gp"];
  41     vedioURL =[NSURL fileURLWithPath:fullpath];
  42     AVPlayerItem* playerItem = [AVPlayerItem playerItemWithURL:vedioURL];
  43     AVPlayer* playVideo = [[AVPlayer alloc] initWithPlayerItem:playerItem];
  44     playerViewController = [[AVPlayerViewController alloc] init];
  45     playerViewController.player = playVideo;
  46     playerViewController.player.volume = 0;
  47     playerViewController.view.frame = self.view.bounds;
  48     [self.view addSubview:playerViewController.view];
  49     [playVideo play];
}
50
51 -(NSString *)documentsDirectory{
  52     NSArray *paths = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,
  53                                         NSUserDomainMask, YES);
  54     NSString *documentsDirectory = [paths objectAtIndex:0];
  55     return documentsDirectory;
}
```



Partir des matériaux à:
int.com



Disable Auto Layout



Application Outlets

Les diapositives sont faites à partir des matériaux à:

www.tutorialpoint.com

```
#import

@interface APPViewController : UIViewController

@property (strong, nonatomic) IBOutlet UIImageView *imageView;

- (IBAction)takePhoto: (UIButton *)sender;
- (IBAction)selectPhoto:(UIButton *)sender;

@end
```

```
@interface APPViewController : UIViewController <UIImagePickerControllerDelegate, UINavigationControllerDelegate>
```

```
@property (strong, nonatomic) IBOutlet UIImageView *imageView;

- (IBAction)takePhoto:(UIButton *)sender {

    UIImagePickerController *picker = [[UIImagePickerController alloc] init];
    picker.delegate = self;
    picker.allowsEditing = YES;
    picker.sourceType = UIImagePickerControllerSourceTypeCamera;

    [self presentViewController:picker animated:YES completion:NULL];
}
```

```

- (IBAction)selectPhoto:(UIButton *)sender {
    UIImagePickerController *picker = [[UIImagePickerController alloc] init];
    picker.delegate = self;
    picker.allowsEditing = YES;
    picker.sourceType = UIImagePickerControllerSourceTypePhotoLibrary;

    [self presentViewController:picker animated:YES completion:NULL];
}

- (void)imagePickerController:(UIImagePickerController *)picker didFinishPickingMedia
{
    UIImage *chosenImage = info[UIImagePickerControllerEditedImage];
    self.imageView.image = chosenImage;

    [picker dismissViewControllerAnimated:YES completion:NULL];
}

- (void)imagePickerControllerDidCancel:(UIImagePickerController *)picker {
    [picker dismissViewControllerAnimated:YES completion:NULL];
}

if (![UIImagePickerController isSourceTypeAvailable:UIImagePickerControllerSourceType
    UIAlertView *myalertView = [[UIAlertView alloc] initWithTitle:@"Error"
        message:@"Device has no camera"
        delegate:nil
        cancelButtonTitle:@"OK"
        otherButtonTitles: nil];

    [myalertView show];
}

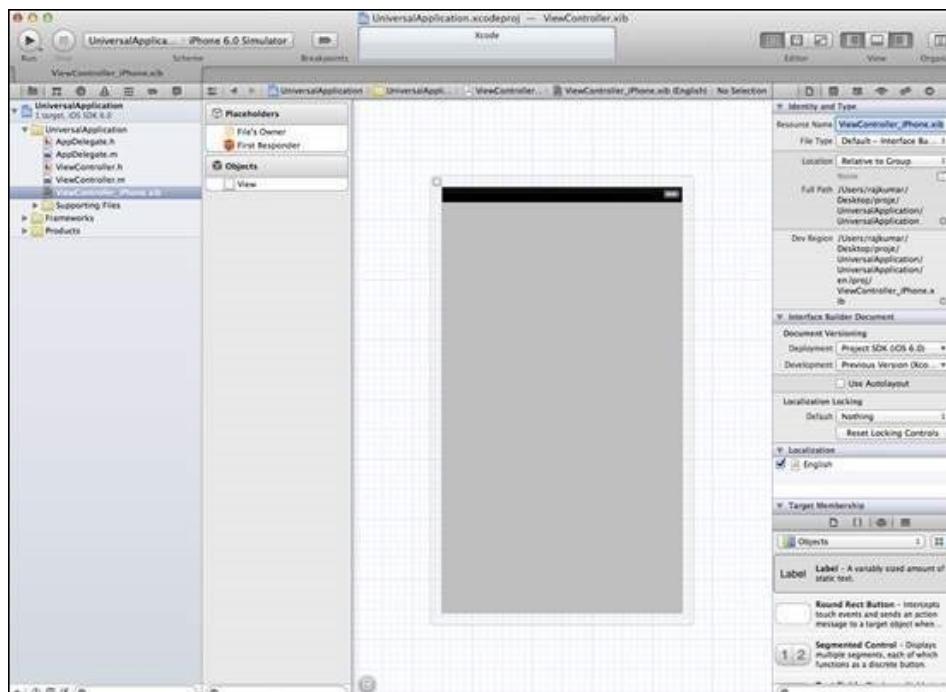
```

iOS - Applications universelles

-- Une application designer pour iPhone et iPad en un seul binaire

Étape-1: Créer une simple « View Based Application »

Étape-2: Changer le nom du fichier **ViewController.xib** en **ViewController_iPhone.xib**

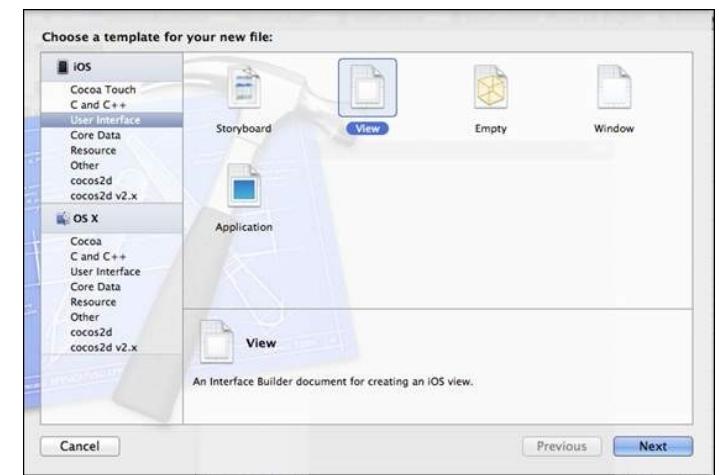


Les diapositives sont faites à partir des matériaux à:

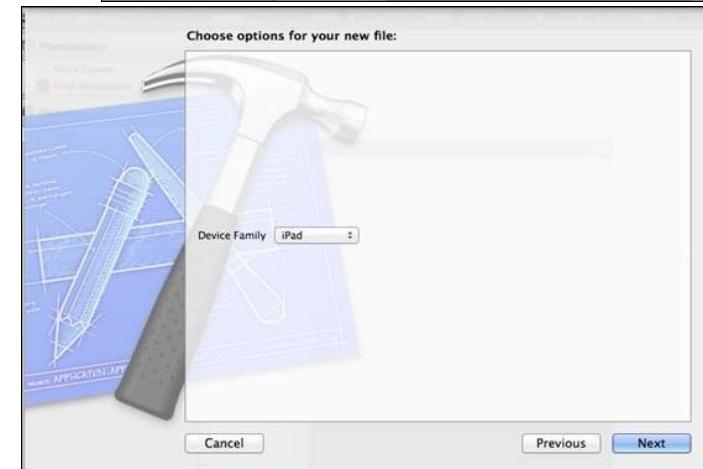
www.tutorialpoint.com

iOS - Applications universelles

Étape-3: Sélectionnez File -> New -> File... puis sélectionnez la sous-section "User Interface" et sélectionnez View.
Cliquez sur « Next»



Étape-4: Sélectionnez la « Device Family» comme iPad et cliquez sur «Next»

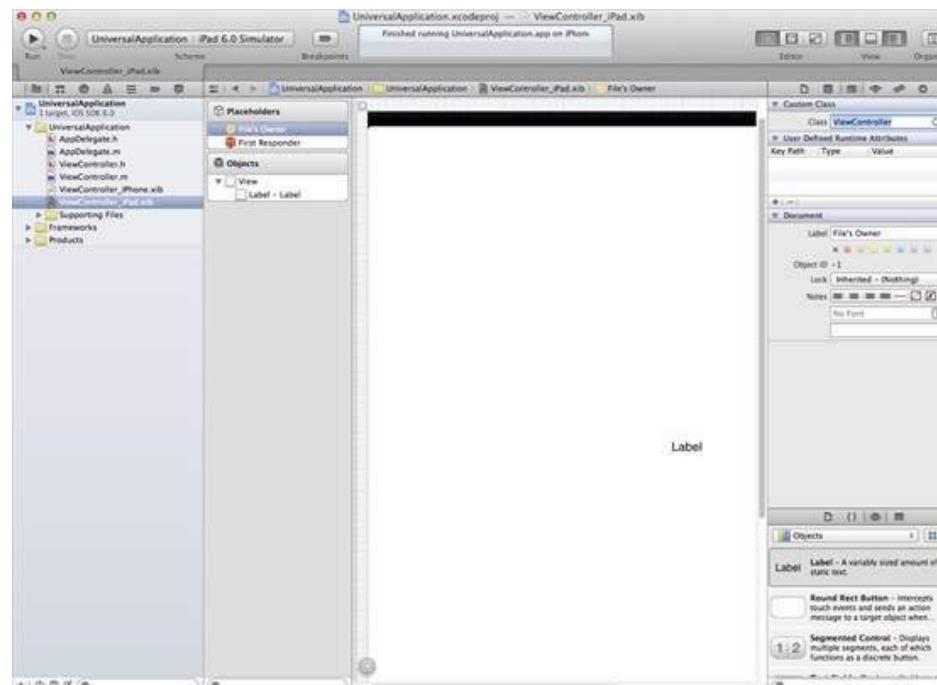


iOS - Applications universelles

Étape-5: Enregistrez le fichier en tant que « **ViewController_iPad. xib** » et sélectionnez « **create** »

Étape-6: Ajoutez une « **Label** » au centre de l'écran dans **ViewController_iPhone. xib** et **ViewController_iPad. xib**.

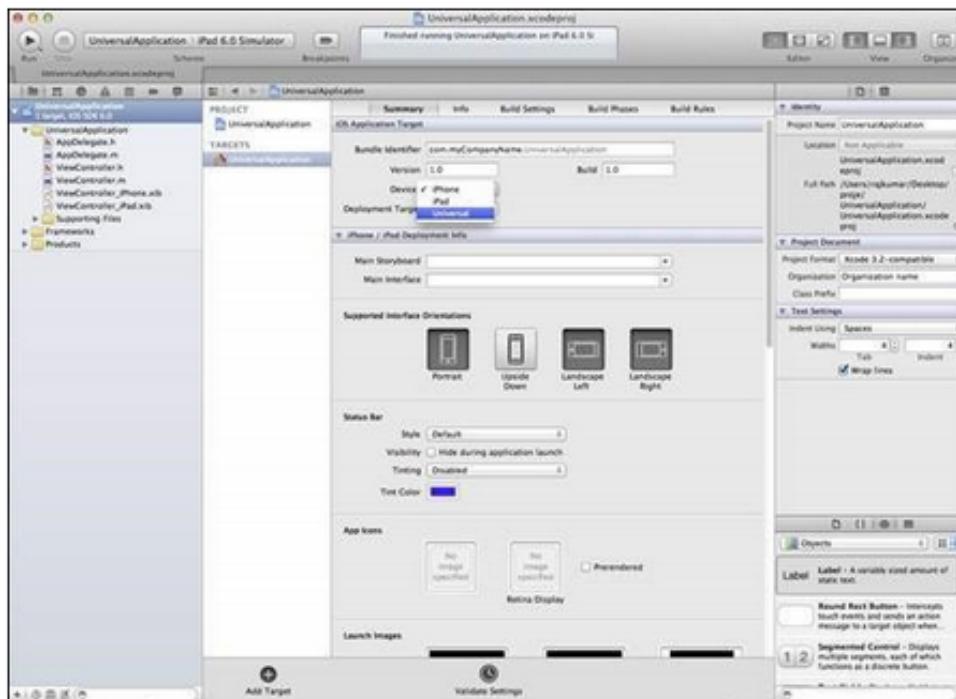
Étape-7: Dans **ViewController_iPad. xib**, sélectionnez « **Identity Inspector** » et définissez la classe personnalisée comme **ViewController**



iOS - Applications universelles

Étape-8: Mettre à jour l'application: **DidFinishLaunchingWithOptions** dans **AppDelegate.m** comme suit

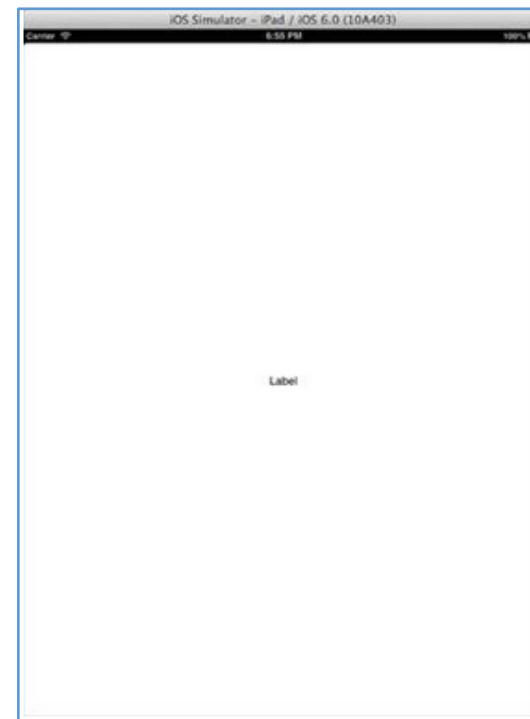
Étape-9: Mettre à jour « Device in Project Summary » à « Universal » comme montré ci-dessous -



```
- (BOOL)application:(UIApplication *)application  
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {  
    self.window = [[UIWindow alloc] initWithFrame:[UIScreen  
mainScreen] bounds]];  
  
    // Override point for customization after application launch.  
    if (UI_USER_INTERFACE_IDIOM() == UIUserInterfaceIdiomPhone) {  
        self.viewController = [[ViewController alloc]  
initWithNibName:@"ViewController_iPhone" bundle:nil];  
    } else {  
        self.viewController = [[ViewController alloc] initWithNibName:  
@"ViewController_iPad" bundle:nil];  
    }  
    self.window.rootViewController = self.viewController;  
    [self.window makeKeyAndVisible];  
    return YES;  
}
```

iOS - Applications universelles-- Output

Étape-10: Il s'agit de la **output** suivante sur le simulateur **iPhone** et **iPad**



iOS - Gestion de la caméra

Étape-1: Créez une simple « View Based Application »

Étape-2: Ajoutez un bouton dans **ViewController.xib** et créez **IBAction** pour le bouton

Étape-3: Ajoutez une « **Image View** » et créez **IBOutlet** en l'appelant **imageView**

Étape-4: Mise à jour de **ViewController.h** comme suit

```
#import <UIKit/UIKit.h>

@interface ViewController : UIViewController<UIImagePickerControllerDelegate>
    UIImagePickerController *imagePicker;
    IBOutlet UIImageView *imageView;
}

-(IBAction)showCamera:(id)sender;
@end
```

Étape-5: Mise à jour de **ViewController.m** comme suit

Output



```
#import "ViewController.h"

@interface ViewController ()
```

```
@end
```

```
@implementation ViewController
```

```
- (void)viewDidLoad {
    [super viewDidLoad];
}
```

```
- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}
```

```
- (IBAction)showCamera:(id)sender {
    imagePicker.allowsEditing = YES;
    if ([UIImagePickerController isSourceTypeAvailable:
        UIImagePickerControllerSourceTypeCamera]) {
        imagePicker.sourceType = UIImagePickerControllerSourceTypeCamera;
    } else {
        imagePicker.sourceType =
        UIImagePickerControllerSourceTypePhotoLibrary;
    }
    [self presentViewController:imagePicker animated:YES];
}
```

```
- (void)imagePickerController:(UIImagePickerController *)picker
didFinishPickingMediaWithInfo:(NSDictionary *)info {
    UIImage *image = [info objectForKey:UIImagePickerControllerEditedImage];
    if (image == nil) {
        image = [info objectForKey:UIImagePickerControllerOriginalImage];
    }
    imageView.image = image;
}
```

```
- (void)imagePickerControllerDidCancel:(UIImagePickerController *)picker {
    [self dismissViewControllerAnimated:YES];
}
```

```
@end
```