

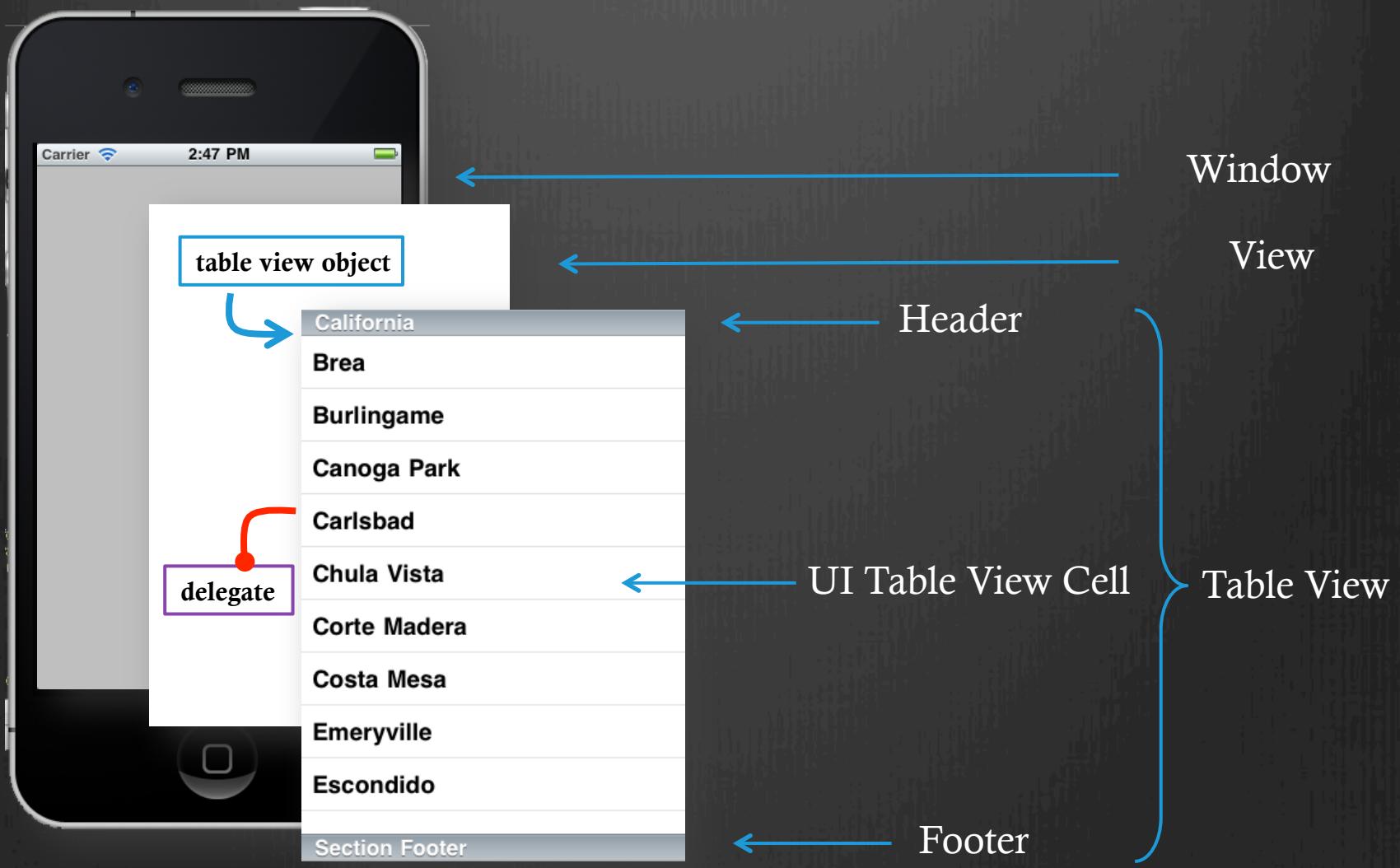
# Chapter 10

## iOS Table View

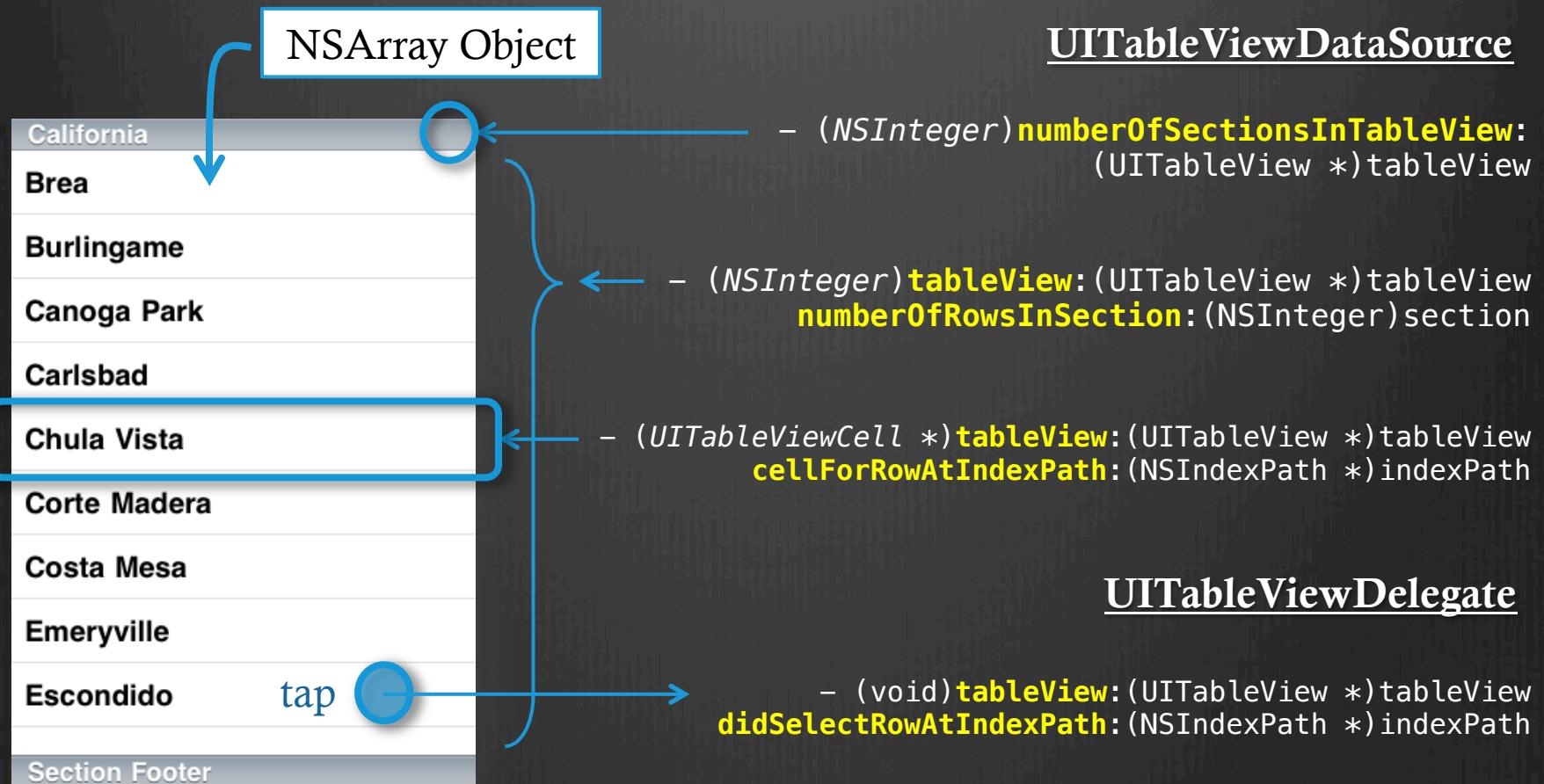
# Table View

- ⦿ Table View เป็น control ที่ใช้เพื่อแสดงข้อมูล ในลักษณะของตาราง ประกอบด้วย class 2 class คือ **UITableView** ทำหน้าที่เป็น view class และ **UITableViewController** เป็น controller class
- ⦿ ใน Table View 1 table จะมี row ที่ใช้แสดงข้อมูล เรียกว่า Table View Cell (class **UITableViewCell**)
- ⦿ เราสามารถกำหนด Type หรือ Style ให้กับ Cell ได้ โดยกำหนด Cell Style แต่ถ้า style ที่มีให้ไม่ตรงกับที่ต้องการ เราสามารถสร้าง template ของ cell ได้เอง
- ⦿ เราสามารถแสดง Header และ Footer ของ Table ได้
- ⦿ เราสามารถแบ่งกลุ่มของข้อมูล ใน Table View ได้ เช่น ใน App Contact บน iOS จะแบ่งกลุ่มของข้อมูลตามคำนำหน้าชื่อ
- ⦿ ข้อมูลที่แสดงใน Table View จะมาจากการ **NSArray** ถ้ามีการแบ่งกลุ่มข้อมูล ข้อมูลของแต่ละกลุ่มก็จะอยู่ในแต่ละ Array

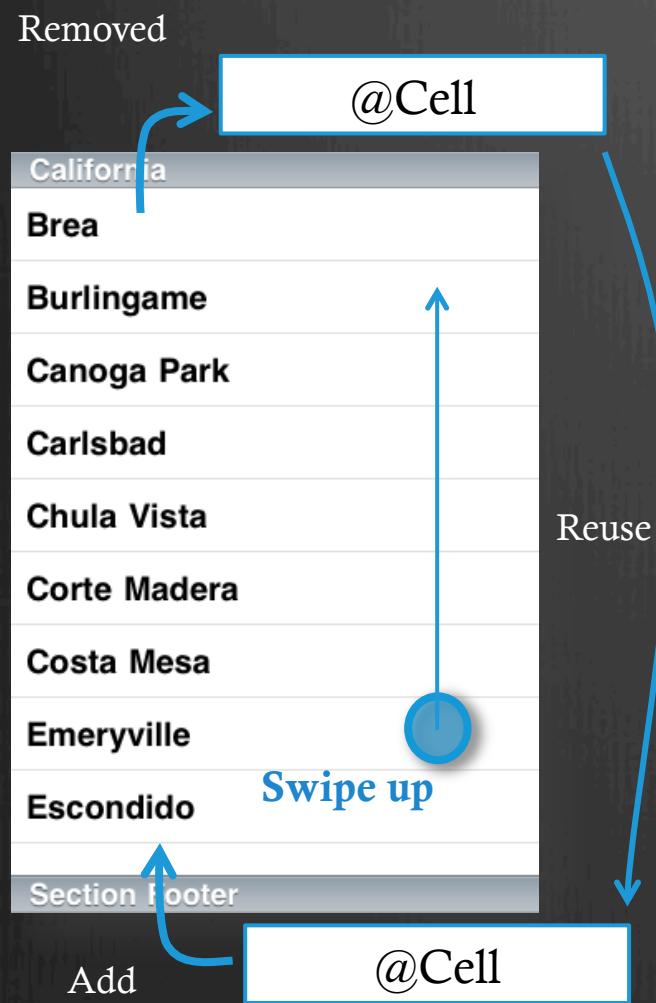
# iOS TableView Layers



# Table Protocol – Data Source & Delegate



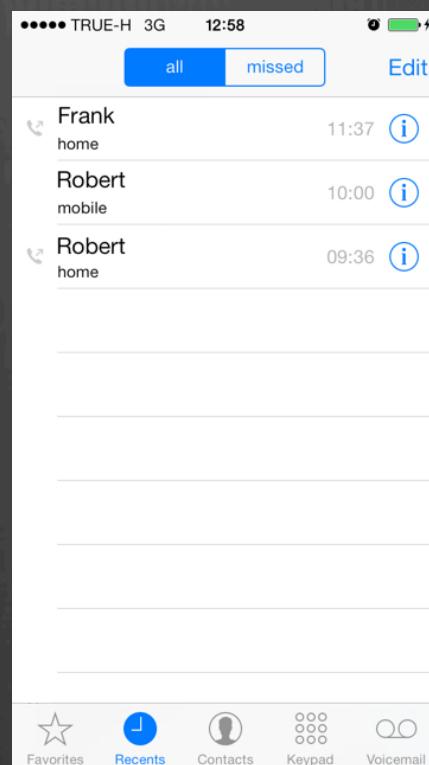
# Reuse Identifier



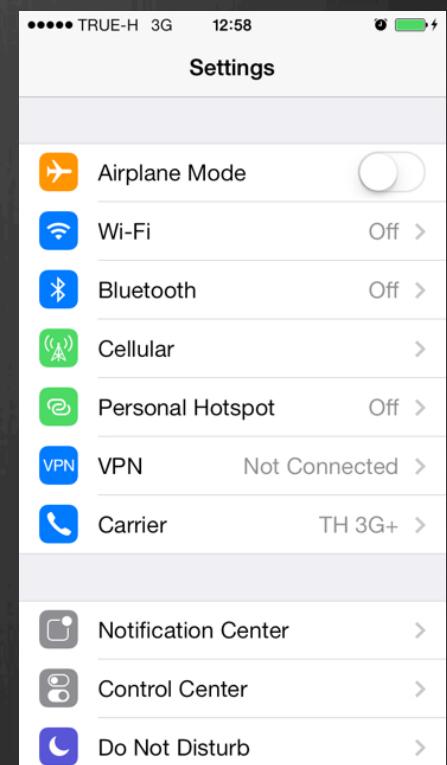
- ในการนี้ที่มีข้อมูลลู๊กจำนวนมากแสดงบน Table มากร้า เรา load ข้อมูลทั้งหมดเข้าไปใน table view ก็จะต้องใช้ memory จำนวนมากในการสร้าง cell ทำให้เกิดปัญหาเรื่อง memory และ performance
  - แนวทางที่วิศวกรรมของ Apple แก้ปัญหาคือ เมื่อ load ข้อมูล Table View จะสร้าง Cell เฉพาะที่ปรากฏบน view เท่านั้นไม่ว่าจำนวน element ของ array จะมีกี่ element
  - เมื่อ user ทำการ scroll table iOS จะทำการ remove cell ออกจาก Table View และเพิ่ม cell ใหม่เข้ามาอีกด้านหนึ่ง
  - แต่ถ้าต้อง delete / create cell ตลอดเวลา ก็จะเกิดปัญหานี้การใช้ CPU และ Memory มากเกินไป ผู้ออกแบบจึงใช้วิธีนำ Cell ที่ถูก remove ออกไป re-use กลับมาใช้ใหม่ จึงไม่ต้อง create cell ใหม่ แค่เอา data fill ใส่ลงไปใน cell เท่านั้น

# Dynamic vs. Static Table View

- บน iOS นั้น Table View จะมี 2 แบบ คือ Dynamic View และ Static View
- Dynamic View จะเป็น Table View ที่อ่านข้อมูลมาจาก Data Source ซึ่งอาจเปลี่ยนแปลงได้ เช่น ข้อมูล log ใน App Phone เป็นต้น
- Static View เป็น Table View ที่ข้อมูลในแต่ละ row ของ table ไม่มีการเปลี่ยนแปลง หรือเปลี่ยนแปลงน้อยมาก เช่น ใน Settings ของ iOS จะเป็นแบบ Static View



Dynamic



Static

# Lab 1-5 : Simple Table View (1/8)

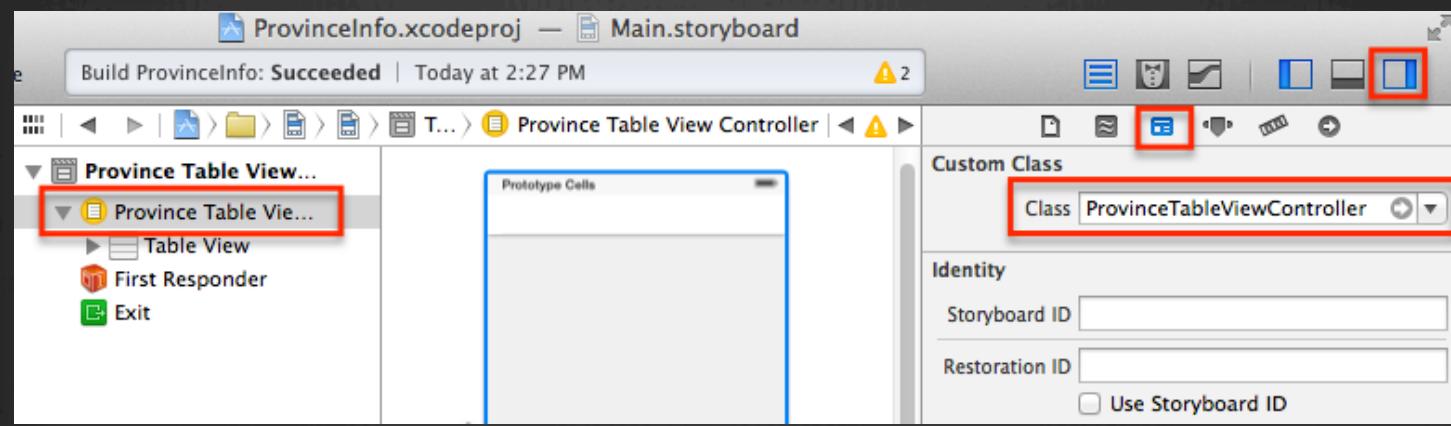
- ◎ วัตถุประสงค์
  - ◎ เพื่อให้สามารถสร้าง Table View อย่างง่ายได้
- ◎ ขั้นตอน
  - ◎ สร้าง view ใน storyboard โดยให้ view เริ่มต้นเป็น Table View
  - ◎ เพิ่ม code ในการ implement Data Source และ Delegate ของ Table View
  - ◎ เพิ่ม code เพื่อรับการเลือกรายการใน Table View

# Task : Create Project (2/8)

1. จาก Xcode สร้าง project ใหม่โดยเลือก iOS > Application > Single View Application
2. ตั้งชื่อ project ว่า “ProvinceInfo” และเลือก Devices เป็น iPhone
3. Click “Next” เลือก folder ที่จะ save project แล้ว click ปุ่ม “Create”
4. เปิด Main.storyboard แล้วลบ View Controller ออกจาก storyboard
5. ลบ ViewController.h และ ViewController.m ออกจาก project โดย click ขวา เลือก Delete และเลือก Move To Trash

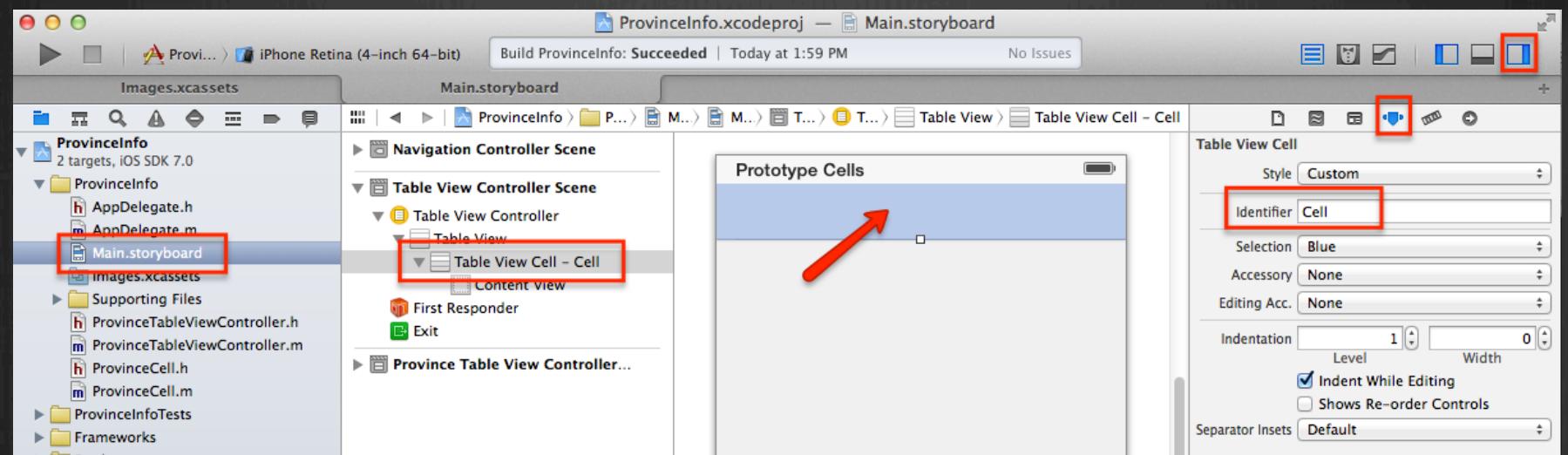
# Task : Create UITableView Controller Class (3/8)

6. สร้าง class ใหม่ โดย click ขวาที่ project ใน Navigation Pane เลือก New File... > iOS > Cocoa Touch > Objective-C class และ click ปุ่ม “Next”
7. ตั้งชื่อ class ว่า “ProvinceTableViewController” และเลือก Subclass of เป็น “UITableViewController” (ไม่ต้องเลือก check box ทั้ง 2 ตัว) จากนั้น click ปุ่ม Next และ Create ตามลำดับ
8. เปิดไฟล์ Main.storyboard และลาก Table View Controller จาก Library Pane มาวางลงบน storyboard
9. Click ที่ table view และเปิด Identity Inspector บน Inspector Pane และเปลี่ยน class เป็น ProvinceTableViewController



# Task : Create UITableView Controller Class (4/8)

10. กำหนด Identifier ของ Table View Cell เพื่อให้สามารถ reuse cell ใน method “tableView:cellForRowAtIndexPath;” ได้ โดย click เลือก Cell บน Table View ใน Storyboard เปิด Attribute inspector และกำหนด property “Identifier” เป็น “Cell”



# Task : Initial Array (5/8)

11. เปิดไฟล์ ProvinceTableViewController.h เพิ่ม property ให้กับ class ตั้งชื่อว่า “provinces” เป็นประเภท NSArray

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

@interface ProvinceTableViewController : UITableViewController
@property (strong, nonatomic) NSMutableArray * provinces;
@end
```

12. เปิดไฟล์ ProvinceTableViewController.m เพิ่ม code ในการ initial array ใน method “viewDidLoad” และใน method “didReceiveMemoryWarning”

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    self.provinces = [NSMutableArray arrayWithObjects:@"กรุงเทพฯ", @"พัทยา",
                      @"ภูเก็ต", @"เชียงใหม่", @"ขอนแก่น", nil];
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    self.provinces = nil;
}
```

# Task : Implement Table View Data Source (6/8)

13. แก้ code ใน method “numberOfSectionsInTableView:” และ “tableView:numberOfRowsInSection:” โดยลบ warning ออกและเพิ่ม code เข้าไปดังนี้

```
- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    #warning Potentially incomplete method implementation. ลบออก
    return 1;
}

- (NSInteger)tableView:(UITableView *)tableView
    numberOfRowsInSection:(NSInteger)section
{
    #warning Incomplete method implementation. ลบออก
    return [self.provinces count];
}
```

# Task : Implement Table View Delegate (7/8)

14. แก้ code ใน method “tableView:cellForRowAtIndexPath:” เพื่อ create object ของ UITableViewCell และ assign ค่าจาก array ไปยัง text ของ cell

```
- (UITableViewCell *)tableView:(UITableView *)tableView
    cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"Cell";
    UITableViewCell *cell =
        [tableView dequeueReusableCellWithIdentifier:CellIdentifier
                                 forIndexPath:indexPath];

    if (!cell) {
        cell = [[UITableViewCell alloc]
                initWithStyle:UITableViewCellStyleDefault
                reuseIdentifier:CellIdentifier];
    }
    cell.textLabel.text = [self.provinces objectAtIndex:indexPath.row];
    return cell;
}
```

# Task : Handle Touch (8/8)

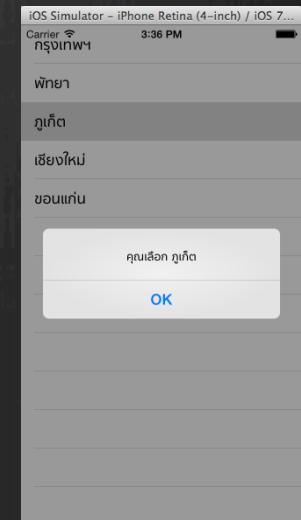
15. เพิ่ม method “tableView:didSelectRowAtIndexPath:” เพื่อแสดง alert เมื่อ user แตะเลือกรายการใน table view

```
#pragma mark - Table view delegate

- (void)tableView:(UITableView *)tableView
    didSelectRowAtIndexPath:(NSIndexPath *)indexPath
{
    NSString * selectedProvince = [NSString stringWithFormat:@"คุณเลือก %@", [self.provinces objectAtIndex:indexPath.row]];

    [[[UIAlertView alloc] initWithTitle:nil
                                message:selectedProvince
                                delegate:nil
                           cancelButtonTitle:@"OK"
                           otherButtonTitles:nil, nil] show];
}
```

16. Run เพื่อดูผลลัพธ์



# Lab 2-5 : Section in Table View (1/8)

- ◎ วัตถุประสงค์
  - ◎ เพื่อให้สามารถสร้างแสดงข้อมูลใน Table View เป็น group หรือ section ได้
  - ◎ เข้าใจรูปแบบของ style ใน table view
- ◎ ขั้นตอน
  - ◎ เพิ่ม NSArray สำหรับแต่ละภาคและใส่ข้อมูลแต่ละจังหวัดตามภาคลงไปใน array
  - ◎ เพิ่ม code เพื่อแสดง title ของ section
  - ◎ แก้ code เดิมใน method ของ Data Source และ Delegate เพื่อรับการแสดงผลแบบแบ่ง section
  - ◎ แก้ code เพื่อรับการ touch ใหม่

# Task : Add more Arrays (2/8)

- เพิ่ม NSArray สำหรับเก็บข้อมูลของจังหวัด ในแต่ละภาค

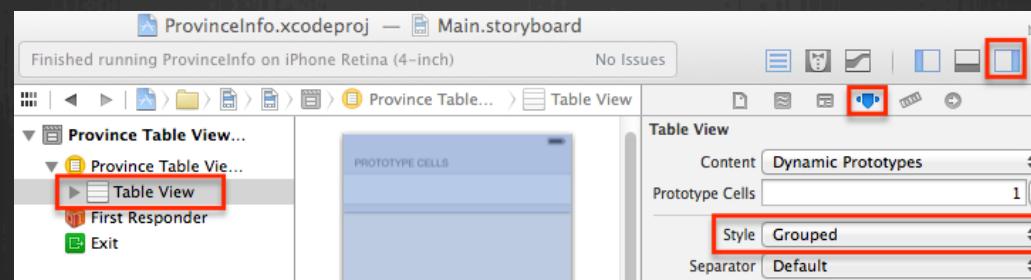
```
#import <UIKit/UIKit.h>

@interface ProvinceTableViewController : UITableViewController

@property (strong, nonatomic) NSMutableArray *provinces;
@property (strong, nonatomic) NSMutableArray *northern;
@property (strong, nonatomic) NSMutableArray *southern;
@property (strong, nonatomic) NSMutableArray *northEastern;
@property (strong, nonatomic) NSMutableArray *centralRegion;

@end
```

- เปลี่ยน style ของ Table View ให้แสดงผลแบบ group แทน โดยเปิดไฟล์ Main.storyboard และ click ที่ Table View ใน Document Outline จากนั้นเปิด Attributes Inspector ใน Inspector Pane และเปลี่ยน property “Style” เป็น Grouped



# Task : Initial Arrays (3/8)

3. เปิดไฟล์ ProvinceViewController.m แก้ code ใน method “viewDidLoad:” เพื่อเพิ่ม code การ initial จังหวัด ในแต่ละภาค และ ใน method “didReceiveMemoryWarning:” เพื่อ clear memory

```

- (void)viewDidLoad
{
    [super viewDidLoad];
    self.provinces = [[NSMutableArray alloc] initWithObjects:@"กรุงเทพฯ", @"พัทยา",
                      @"ภูเก็ต", @"เชียงใหม่", @"ขอนแก่น", nil];

    self.centralRegion = [[NSMutableArray alloc] initWithObjects:@"กรุงเทพฯ", @"สระบุรี",
                          @"สมุทรสาคร", @"สมุทรสงคราม", nil];
    self.northern = [[NSMutableArray alloc] initWithObjects:
                     @"เชียงใหม่", @"เชียงราย", @"แม่ฮ่องสอน", @"ลำพูน", @"ลำปาง", nil];
    self.northEastern = [[NSMutableArray alloc] initWithObjects:@"อุดรธานี",
                          @"อุบลราชธานี", @"ขอนแก่น", @"ร้อยเอ็ด", @"นครราชสีมา", nil];
    self.southern = [[NSMutableArray alloc] initWithObjects:@"พัทงา", @"ภูเก็ต",
                     @"ชุมพร", nil];
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning]; self.provinces = nil;
    self.centralRegion = nil; self.northern = nil; self.northEastern = nil;
    self.southern = nil;
}

```

# Task : Update Data Source (4/8)

4. แก้ code ใน method “numberOfSectionsInTableView:” และ “tableView:numberOfRowsInSection:” เพื่อให้รองรับข้อมูล 4 ภาค

```
- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    return 4;
}

- (NSInteger)tableView:(UITableView *)tableView
    numberOfRowsInSection:(NSInteger)section
{
    NSInteger total;
    switch (section) {
        case 0: total = [self.centralRegion count]; break;
        case 1: total = [self.northern count]; break;
        case 2: total = [self.northEastern count]; break;
        case 3: total = [self.southern count]; break;
        default: total = 0;
    }
    return total;
}
```

# Task : Add Section's Title (5/8)

5. เพิ่ม method “tableView:titleForHeaderInSection:” เพื่อแสดงชื่อของภาคในแต่ละ section

```
- (NSString *)tableView:(UITableView *)tableView
    titleForHeaderInSection:(NSInteger)section
{
    switch (section) {
        case 0:
            return @"ภาคกลาง"; break;
        case 1:
            return @"ภาคเหนือ"; break;
        case 2:
            return @"ภาคอีสาน"; break;
        case 3:
            return @"ภาคใต้"; break;
        default:
            return @"";
    }
}
```

# Task : Update Delegate (6/8)

6. แก้ code ใน method “tableView:cellForRowAtIndexPath:” เพื่อแสดงข้อมูลใน cell ตาม section ที่ถูกแบ่งตามภาค

```
- (UITableViewCell *)tableView:(UITableView *)tableView
    cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"Cell";
    UITableViewCell *cell =
        [tableView dequeueReusableCellWithIdentifier:CellIdentifier forIndexPath:indexPath];

    if (!cell) {
        cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault
                                      reuseIdentifier:@"Cell"];
    }
    switch (indexPath.section) {
        case 0:
            cell.textLabel.text = [self.centralRegion objectAtIndex:indexPath.row]; break;
        case 1:
            cell.textLabel.text = [self.northern objectAtIndex:indexPath.row]; break;
        case 2:
            cell.textLabel.text = [self.northEastern objectAtIndex:indexPath.row]; break;
        case 3:
            cell.textLabel.text = [self.southern objectAtIndex:indexPath.row]; break;
        default: break;
    }
    return cell;
}
```

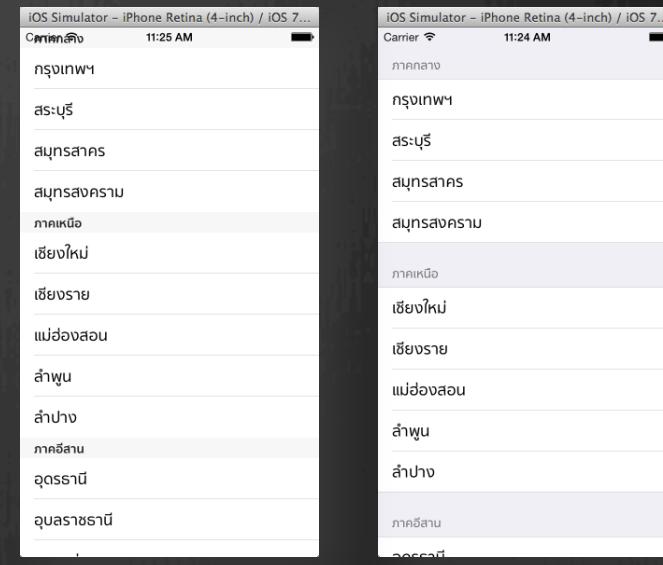
# Task : Update Touch Handler (7/8)

7. แก้ method “tableView:didSelectRowAtIndexPath:” เพื่อรับการ alert ชื่อ  
จังหวัดที่ user เลือกตาม section

# Task : Run & Test (8/8)

## 8. Run เพื่อดูผลลัพธ์

- ทดลองเปลี่ยน style ของ Table View ( ในข้อ 2.) กลับไปเป็นแบบ plain และดูความแตกต่าง
- ทดลองเพิ่ม Footer ของ Section โดยเพิ่ม method
  - `(NSString *)tableView:(UITableView *)tableView titleForFooterInSection:(NSInteger)section;`เพื่อแสดงจำนวนของจังหวัดในภาคนั้น



# UI Table View Cell Style

- ใน API ของ iOS ได้เตรียม style ของ table view cell ไว้หลายแบบเพื่อให้เหมาะสมกับการนำไปใช้
- เราสามารถเปลี่ยน style ได้โดยกำหนดค่าให้กับ property “accessoryType” ของ UITableViewCell



UITableViewCellAccessoryCheckmark



UITableViewCellAccessoryDetailDisclosureButton



UITableViewCellAccessoryDisclosureIndicator



UITableViewCellAccessoryNone

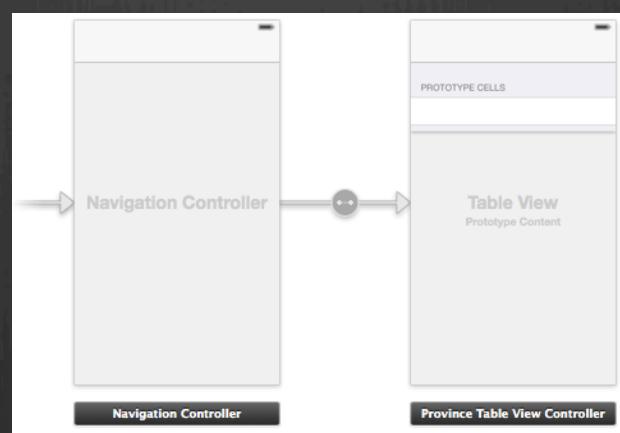
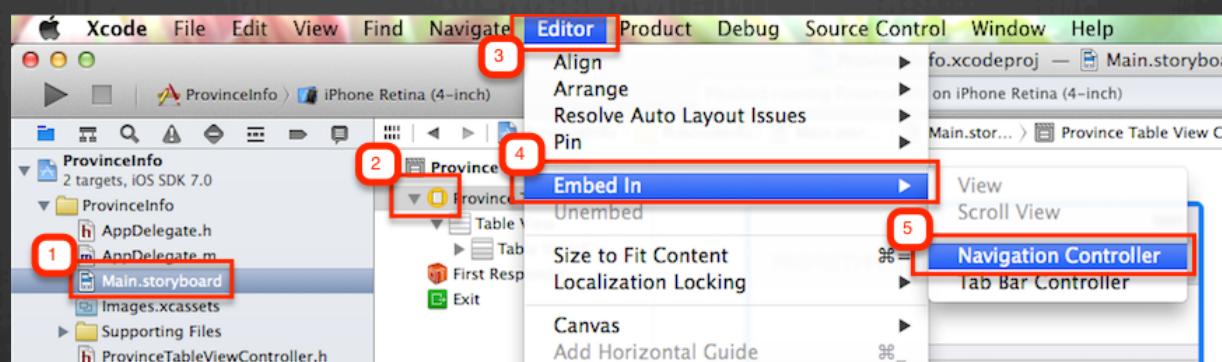


# Lab 3-5 : Table View Cell Style (1/6)

- ◎ วัตถุประสงค์
  - ◎ เพื่อให้เข้าใจ Style ของ Cell ในแบบต่างๆ
  - ◎ เขียน code เพื่อเปลี่ยน style ของ
- ◎ ขั้นตอน
  - ◎ เปลี่ยนให้ Table View เข้าไป Embed อยู่ใต้ Navigation View (optional)
  - ◎ เพิ่ม code เพื่อเปลี่ยน cell style เมื่อ user แตะที่ cell เพื่อแสดงว่า cell นั้นถูกเลือก

# Task : Embed Table View into Navigation View (2/6)

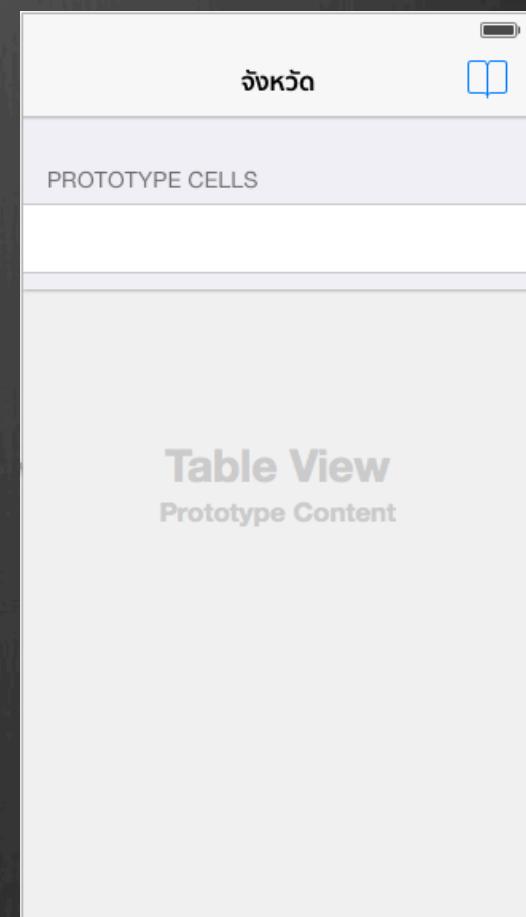
- เพิ่ม Navigation Bar (ให้ Table View embed อธิบายไว้ใน Section 1) โดยเปิด Main.storyboard และ click เลือก Table View บน Storyboard จากนั้นไปที่เมนู Editor > Embed In > Navigation Controller จะเห็นว่ามี view ใหม่บน storyboard



# Task : Embed Table View into Navigation View (3/6)

2. Double Click บนพื้นที่ว่างบน Navigation ที่  
ว่างอยู่ด้านบน Table View เพื่อเปลี่ยน title  
เป็น “จังหวัด”
3. เพิ่ม Control “Bar Button Item” ลงบน  
Navigation Bar ด้านขวาเมื่อ แล้วเปลี่ยน  
Identifier ของ Bar Button เป็น  
“Bookmarks”
4. เปลี่ยน editor mode เป็น Assistant Editor  
แล้วผูก IBAction ของ Bar Button ตั้งชื่อว่า  
“btnBookmarkTapped”

*Note: เราสามารถลองเปลี่ยน Accessory ของ Cell  
เพื่อดูตอน design time ได้โดย click ที่ cell บน Table  
View และเปลี่ยนที่ property “Accessory” ใน  
Attributes Indicator*



# Task : Add another Array (4/6)

5. เปิด ProvinceTableViewController.h เพิ่ม NSMutableArray สำหรับเก็บค่าจังหวัดที่ user เลือกจาก Table

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

@interface ProvinceTableViewController : UITableViewController

@property (strong, nonatomic)NSArray * northern;
@property (strong, nonatomic)NSArray * southern;
@property (strong, nonatomic)NSArray * northEastern;
@property (strong, nonatomic)NSArray * centralRegion;
@property (strong, nonatomic)NSMutableArray * selectedProvinces;

@end
```

6. เปิดไฟล์ ProvinceTableViewController.m เพิ่มการ initial array ใน method “viewDidLoad:”

```
- (void)viewDidLoad
{
    [super viewDidLoad];
self.selectedProvinces = [[NSMutableArray alloc] init];
    // code เดิม
}
```

# Task : Change Cell Accessory Style (5/6)

7. แก้ code “tableView:didSelectRowAtIndexPath:” เพื่อเก็บค่าลง Array

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath
{
    NSString * selectedProvince;
    switch (indexPath.section) {
        case 0:
            selectedProvince = [self.centralRegion objectAtIndex:indexPath.row]; break;
        case 1:
            selectedProvince = [self.northern objectAtIndex:indexPath.row];      break;
        case 2:
            selectedProvince = [self.northEastern objectAtIndex:indexPath.row]; break;
        case 3:
            selectedProvince = [self.southern objectAtIndex:indexPath.row];      break;
        default: break;
    }

    UITableViewCell *cell = [tableView cellForRowAtIndexPath:indexPath];

    if (cell.accessoryType == UITableViewCellAccessoryCheckmark) {
        cell.accessoryType = UITableViewCellAccessoryNone;
        [self.selectedProvinces removeObject:selectedProvince];
    } else {
        cell.accessoryType = UITableViewCellAccessoryCheckmark;
        [self.selectedProvinces addObject:selectedProvince];
    }
}
```

# Task : Display Array Data (6/6)

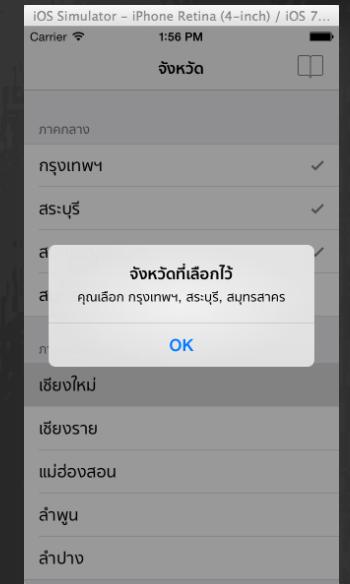
8. เพิ่ม code ใน method “btnBookmarkTapped:” เพื่อแสดงชื่อจังหวัดที่เลือกไว้

```
- (IBAction)btnBookmarkTapped:(id)sender
{
    if ([self.selectedProvinces count] == 0)
        return;

    NSString * msg = [NSString stringWithFormat:@"คุณเลือก %@",
                      [self.selectedProvinces componentsJoinedByString:@", "]];

    [[[UIAlertView alloc] initWithTitle:@"จังหวัดที่เลือกไว้"
                                 message:msg
                                 delegate:nil
                                 cancelButtonTitle:@"OK"
                                 otherButtonTitles:nil] show];
}
```

9. Run เพื่อดูผลลัพธ์



# Custom Table View Cell

- ในกรณีที่ style ของ cell ไม่ตอบโจทย์เราก็สามารถ customize cell บน table view ได้อย่างอิสระ
- ใน Xcode version เก่า การ customize cell ค่อนข้างยุ่งยาก เพราะจะต้องออกแบบ cell ใน xib file และแยกออกจาก table view แต่เมื่อ Xcode เพิ่ม storyboard เข้ามา การออกแบบ cell ทำได้ง่ายขึ้นมาก
- แต่เราก็ยังสามารถใช้เทคนิคแบบ xib file ได้ เช่นเดียวกับที่เราใช้ xib file สำหรับสร้าง view ที่ไม่ได้อยู่ใน storyboard

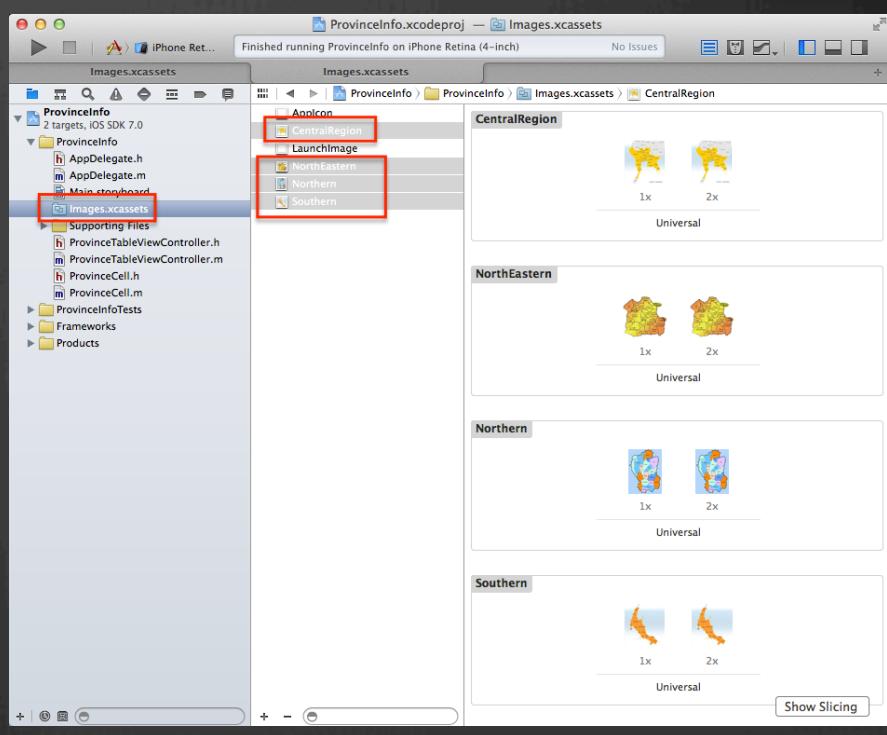


# Lab 4-5 : Custom TableViewCell (1/6)

- ◎ วัตถุประสงค์
  - ◎ เพื่อให้สามารถ customize Table View Cell ตามต้องการได้
- ◎ ขั้นตอน
  - ◎ สร้าง view controller class ใหม่สำหรับ custom table view cell
  - ◎ เปลี่ยน class ของ Cell ใน Table View บน storyboard เป็น class ใหม่ที่สร้างขึ้นมา
  - ◎ วาง control ที่เราต้องการ customize บน cell ใน table view
  - ◎ ผูก delegate แล้วเปลี่ยน class ใน method cellForRowAtIndexPath เป็น class ที่สร้างขึ้นพร้อมทั้ง assign ค่า ให้กับ control บน cell

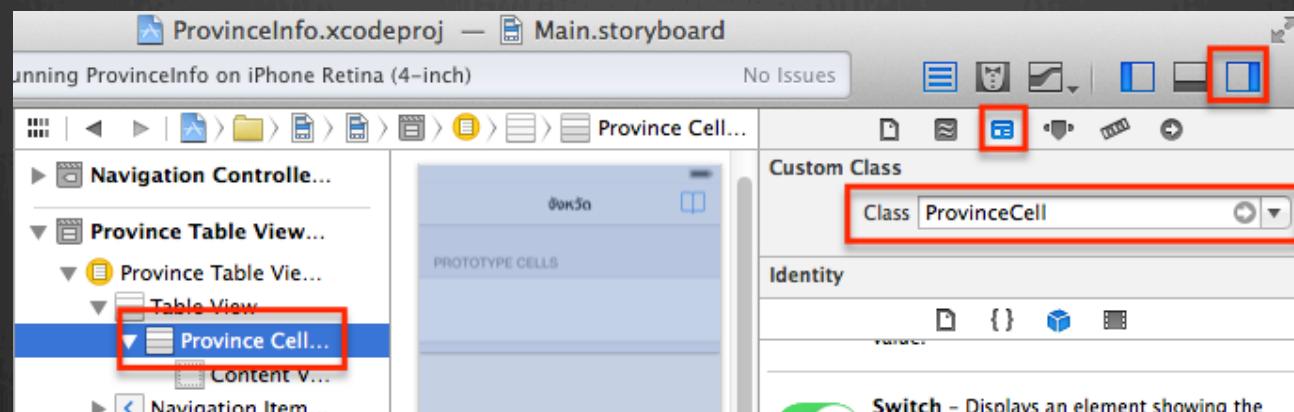
# Task : New Table Cell View Controller Class (2/6)

1. สร้าง class ใหม่สำหรับใช้เป็น view controller ให้กับ cell ของ table view โดย click ขวาที่ project ใน Navigation Pane แล้วเลือก New File ... > iOS > Cocoa Touch > Objective-C class และ click ปุ่ม Next
2. เลือก Subclass of เป็น “UITableViewCell” และตั้งชื่อว่า “ProvinceCell” click ปุ่ม Next แล้ว click ปุ่ม Create
3. เปิดไฟล์ Images.xcassets จาก Navigation Pane แล้ว click ขวาที่พื้นที่ว่างๆ บน image asset (ได้ AppIcon, Launch Image) และเลือก “Import...” จากนั้น browse ไปที่ folder “/Resources 7/Day3 – Lab10/” แล้วเลือกไฟล์ทั้งหมด และ click ปุ่ม Open

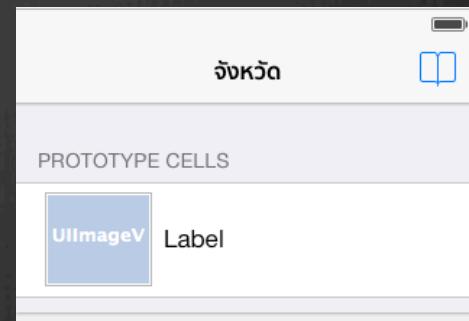


## Task : Customize Cell UI (3/6)

4. เปิด Main.storyboard แล้ว click ที่ Cell ของ Table View จากนั้นเปิด Identity inspector ใน Navigator Pane แล้วเปลี่ยน class ของ cell เป็น “ProvinceCell”

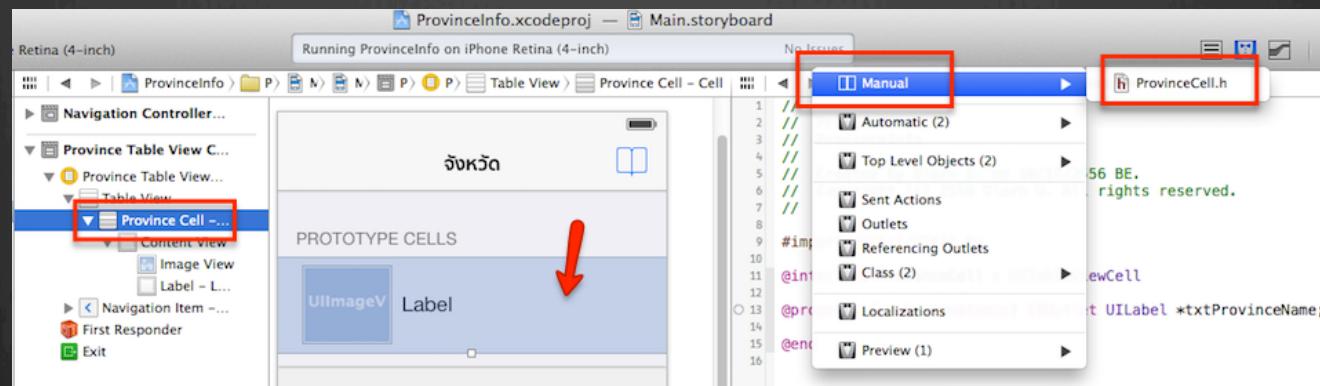


5. ปรับความสูงของ cell เป็น 80px และเพิ่ม control บน cell คือ
  - Image View
  - Label



# Task : Customize Cell UI (4/6)

6. เปลี่ยน editor mode เป็น Assistant Mode แล้ว click ที่ “ProvinceCell” ที่อยู่ใต้รายการ Table View ใน Document Outline จากนั้นเปลี่ยน code ทางด้านข้ามก็ให้เป็นไฟล์ ProvinceCell.h โดยเลือกจาก Navigation Bar ด้านบนของ editor จากเมนู Manual > ProvinceCell.h



7. ผูก Outlet จาก control ที่อยู่ใน Cell ของ Table View โดยที่
- Image View เป็น outlet ตั้งชื่อว่า imgViewRegion
  - Label เป็น Outlet ตั้งชื่อว่า “txtProvinceName”

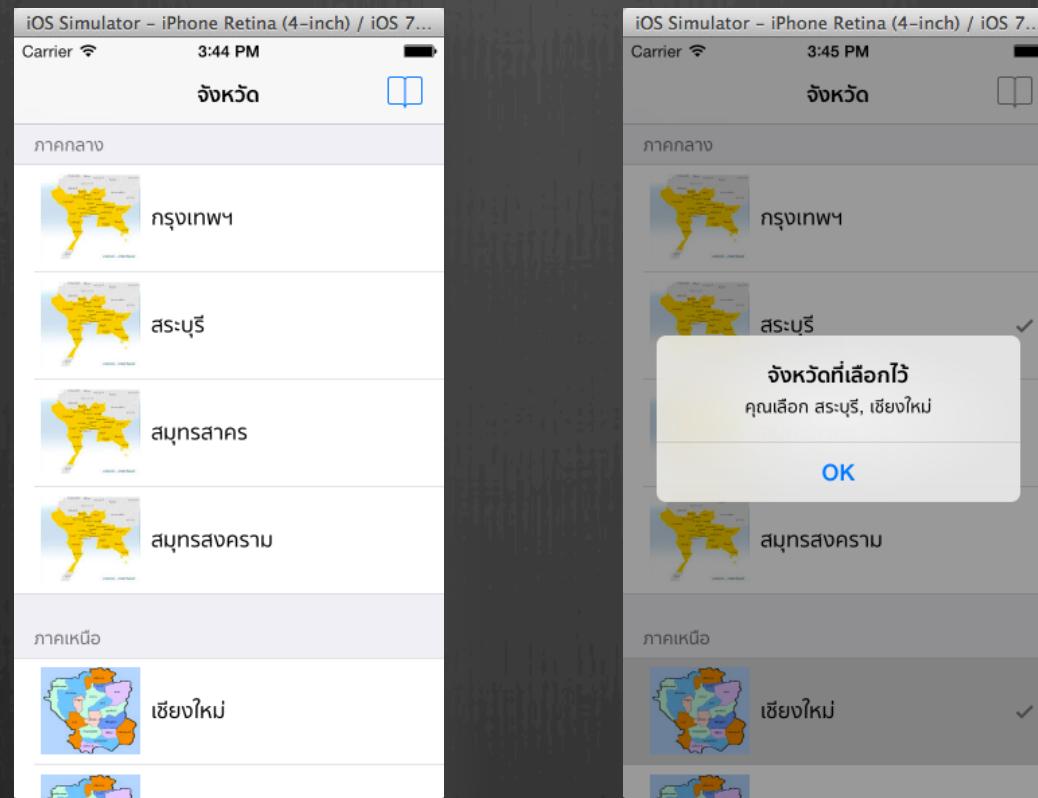
# Task : Customize Cell UI (5/6)

8. เปิดไฟล์ ProvinceTableViewController.m และแก้ code ของ method “tableView:cellForRowAtIndexPath:” โดยเปลี่ยน class ของ cell จาก “UITableViewCell” เป็น “ProvinceCell” แทน

```
- (UITableViewCell *)tableView:(UITableView *)tableView
    cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"Cell";
    ProvinceCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier
                                                       forIndexPath:indexPath];
    if (!cell)
        cell = [[ProvinceCell alloc] initWithStyle:UITableViewCellStyleDefault
                                       reuseIdentifier:@"Cell"];
    switch (indexPath.section) {
        case 0:
            cell.imgViewRegion.image = [UIImage imageNamed:@"CentralRegion"];
            cell.txtProvinceName.text = [self.centralRegion objectAtIndex:indexPath.row];
            break;
        case 1:
            cell.imgViewRegion.image = [UIImage imageNamed:@"Northern"];
            cell.txtProvinceName.text = [self.northern objectAtIndex:indexPath.row];
            break;
        case 2:
            cell.imgViewRegion.image = [UIImage imageNamed:@"NorthEastern"];
            cell.txtProvinceName.text = [self.northEastern objectAtIndex:indexPath.row];
            break;
        case 3:
            cell.imgViewRegion.image = [UIImage imageNamed:@"Southern"];
            cell.txtProvinceName.text = [self.southern objectAtIndex:indexPath.row];
            break;
        default: break;
    }
    return cell;
}
```

# Task : Run & Test (6/6)

## 9. Run เพื่อดูผลลัพธ์



# Lab 5-5 : Swipe to Delete Cell (1/4)

## ◎ วัตถุประสงค์

- ◎ เพื่อให้สามารถลบ Cell บน Table View ตามต้องการได้

## ◎ ขั้นตอน

- ◎ เพิ่ม delegate method ของ Table View เพื่อบอกว่าจะแก้ไข table cell แบบ delete cell
- ◎ เพิ่ม code เพื่อแสดง animation การลบ
- ◎ ลบข้อมูลออกจาก data source (array) ภาย ใต้ section ที่ส่งมาใน index path

# Task : New Table Cell View Controller Class (2/4)

1. เปิดไฟล์ “ProvinceTableViewController.m” เพิ่ม method “tableView:editingStyleForRowAtIndexPath:” เพื่อบอก table view ว่าเราต้องการ edit row และ delete

```
- (UITableViewCellEditingStyle) tableView:(UITableView*)tableView  
    editingStyleForRowAtIndexPath:(NSIndexPath*)indexPath  
{  
    return UITableViewCellEditingStyleDelete;  
}
```

2. ลบ comment ของ method “tableView:commitEditingStyle :forRowAtIndexPath:” ออกไป

# Task : New Table Cell View Controller Class (2/4)

3. เพิ่ม code ใน method “tableView:commitEditingStyle :forRowAtIndexPath:” ดังนี้

```
- (void)tableView:(UITableView *)tableView commitEditingStyle:(UITableViewCellEditingStyle)editingStyle forRowAtIndexPath:(NSIndexPath *)indexPath
{
    if (editingStyle == UITableViewCellEditingStyleDelete) {
        switch (indexPath.section) {
            case 0:
                [self.centralRegion removeObjectAtIndex:indexPath.row]; break;
            case 1:
                [self.northern removeObjectAtIndex:indexPath.row]; break;
            case 2:
                [self.northEastern removeObjectAtIndex:indexPath.row]; break;
            case 3:
                [self.southern removeObjectAtIndex:indexPath.row]; break;
            default: break;
        }
        [tableView deleteRowsAtIndexPaths:@[indexPath]
                           withRowAnimation:UITableViewRowAnimationFade];
    }
    else if (editingStyle == UITableViewCellEditingStyleInsert) { }
}
```

# Task : New Table Cell View Controller Class (2/4)

- Run โปรแกรมเพื่อดูผลลัพธ์

Note : ลบ cell โดย drag ใน cell จากขวาไปซ้าย จะมีปุ่ม Delete ปรากฏขึ้นมา click ที่ delete เพื่อ ลบ cell

