

範例4_jsonDecoder範例說明



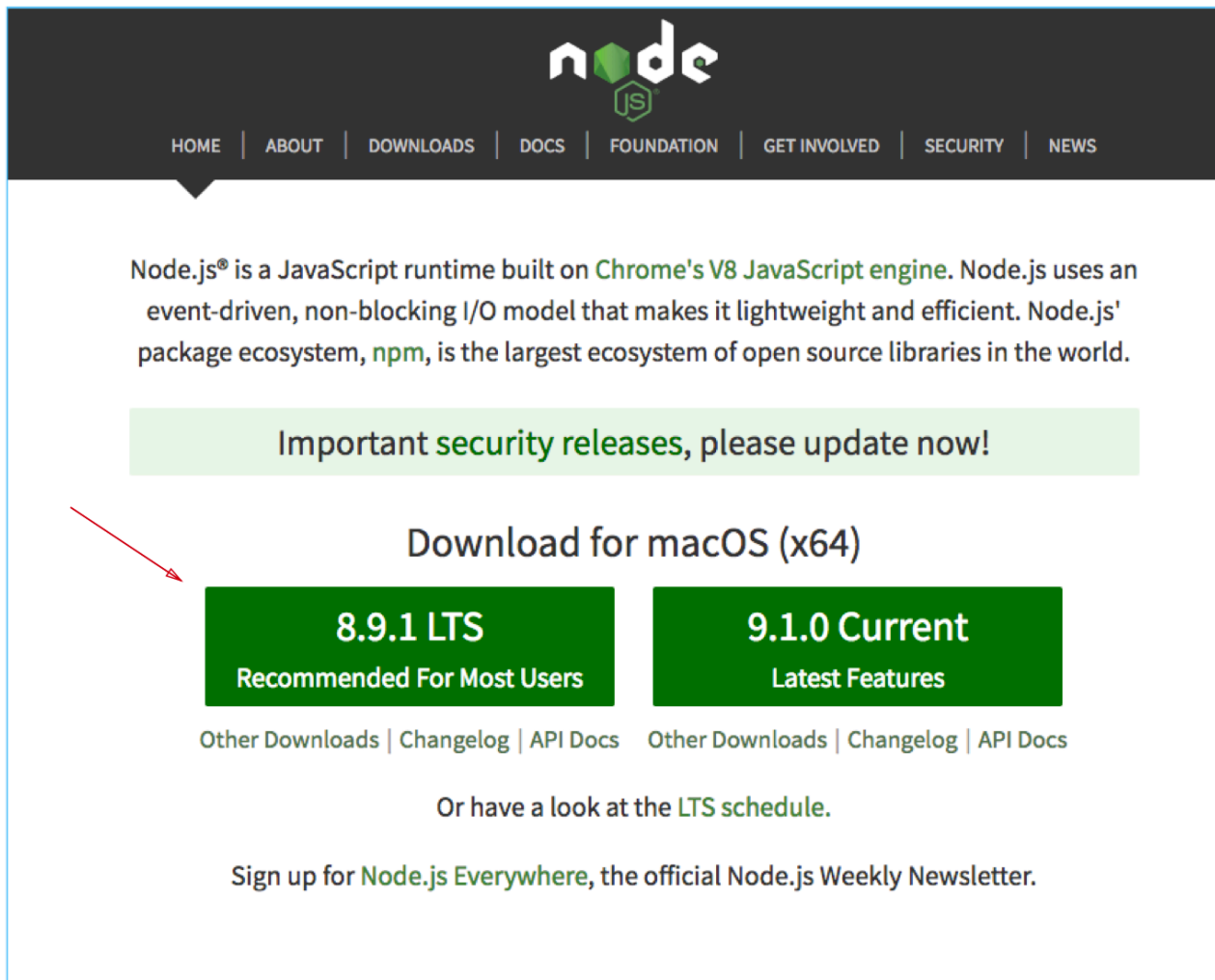
學習目地

- 建立Firebase Host
- 建立json service 網頁
- 使用URLSession下載雲端json內容
- 使用swift 4 新功能JSONDecoder和Keypath
- 使用UICollectionView建立Scroll功能
- 使用MapKit

建立Firebase Host

Firebase提供每個專案，一個靜態網站的功能。讓我們開始建立一個自己專案的Host。

在電腦上，安裝Node.js。至 <https://nodejs.org/en/>，建議下載建議版。



打開終端機，安裝Firebase-tools

```
npm install -g firebase-tools
```

登入Firebase帳號

```
firebase login
```

建立專案目錄，先進入到文件目錄內，建立iosWeb目錄，進入iosWeb目錄，執行Firebase init。

```
mkdir iosWeb  
cd iosWeb  
Firebase init
```

選擇這個目錄目前要的功能，請使用方向鍵，移至Hosting，再按空白鍵選擇。然後按下Enter。

```
? Which Firebase CLI features do you want to setup for this folder  
res, then Enter to confirm your choices.  
○ Database: Deploy Firebase Realtime Database Rules  
○ Firestore: Deploy rules and create indexes for Firestore  
○ Functions: Configure and deploy Cloud Functions  
> ● Hosting: Configure and deploy Firebase Hosting sites  
○ Storage: Deploy Cloud Storage security rules
```

再選擇使用哪一個Firebase專案。

建立web的目錄，預設是public，使用預設值，直接按下Enter。

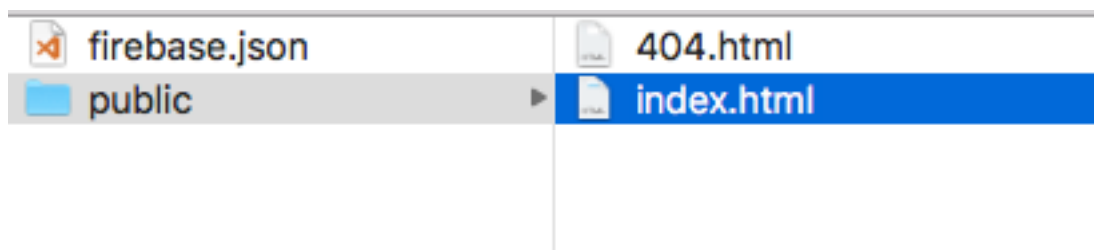
```
[? What do you want to use as your public directory? public
```

選擇單一頁面的App

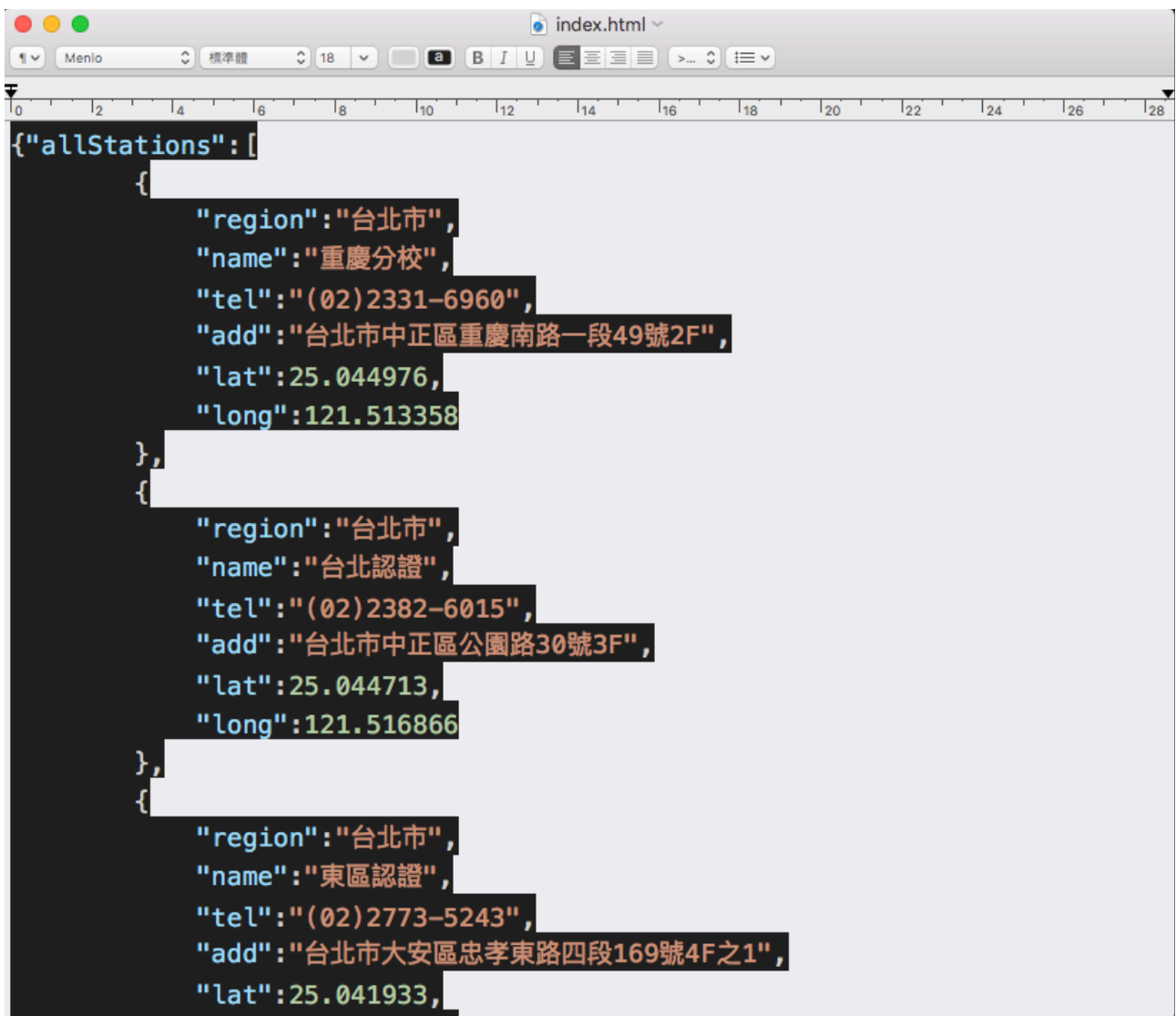
```
? Configure as a single-page app (rewrite all urls to /index.html)? (y/N)
```

Firebase Hosting的專用目錄建立完成。

使用Finder進入Documents/iosWeb/public/內，並用文字編輯器打開index.html。



將index.html所有內容刪除，打開我們提供的gjun.json,複製所有內容，並且在index.html內貼上。

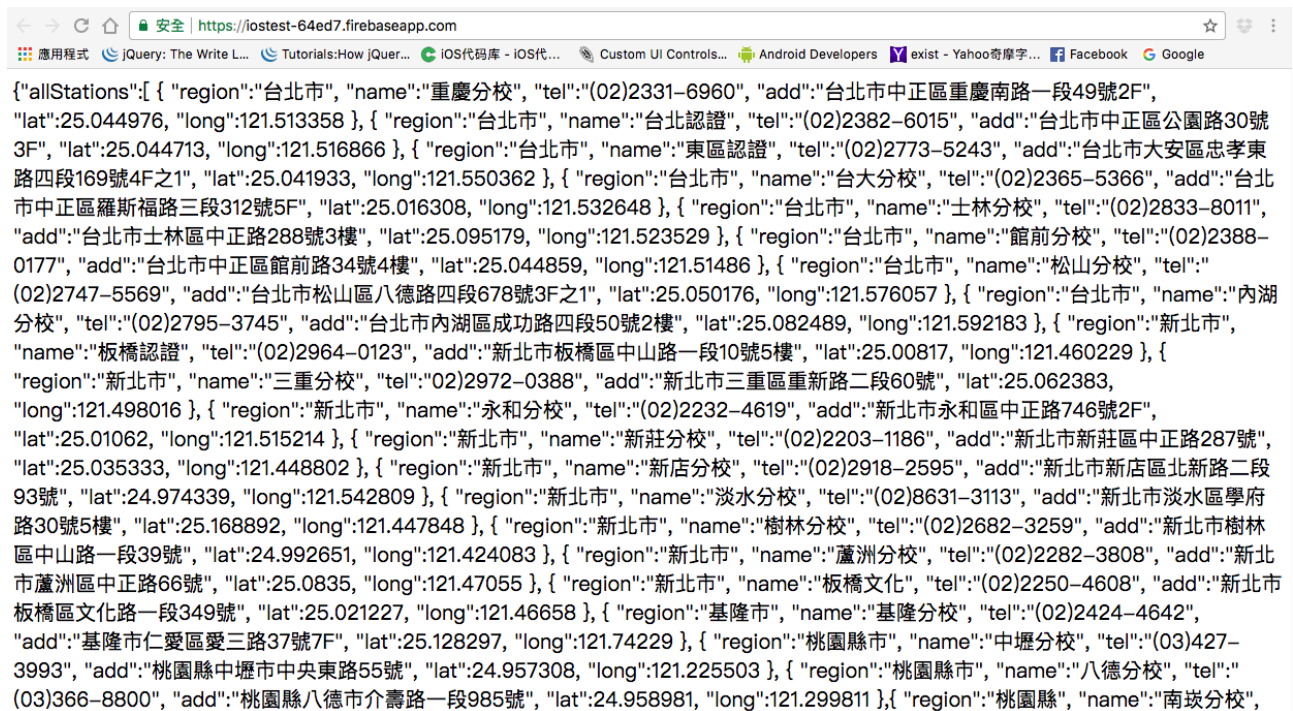
A screenshot of a text editor window titled 'index.html'. The editor shows a JSON array of three station objects. The first object is for '重慶分校' (Chongqing Branch) in Taipei, with address '台北市中正區重慶南路一段49號2F'. The second object is for '台北認證' (Taipei Certification) in Taipei, with address '台北市中正區公園路30號3F'. The third object is for '東區認證' (East District Certification) in Taipei, with address '台北市大安區忠孝東路四段169號4F之1'. Each object contains fields for region, name, tel, add, lat, and long.

```
{  
  "allStations": [  
    {  
      "region": "台北市",  
      "name": "重慶分校",  
      "tel": "(02)2331-6960",  
      "add": "台北市中正區重慶南路一段49號2F",  
      "lat": 25.044976,  
      "long": 121.513358  
    },  
    {  
      "region": "台北市",  
      "name": "台北認證",  
      "tel": "(02)2382-6015",  
      "add": "台北市中正區公園路30號3F",  
      "lat": 25.044713,  
      "long": 121.516866  
    },  
    {  
      "region": "台北市",  
      "name": "東區認證",  
      "tel": "(02)2773-5243",  
      "add": "台北市大安區忠孝東路四段169號4F之1",  
      "lat": 25.041933,  
      "long": 121.516866  
    }  
  ]  
}
```

回到終端機內，執行firebase deploy --only hosting。將index.html佈置於雲端。

```
Firease deploy --only hosting
```

回到Firebase console，進入Hosting內，點選Domain內的網址，將進入您專案專屬的網站。



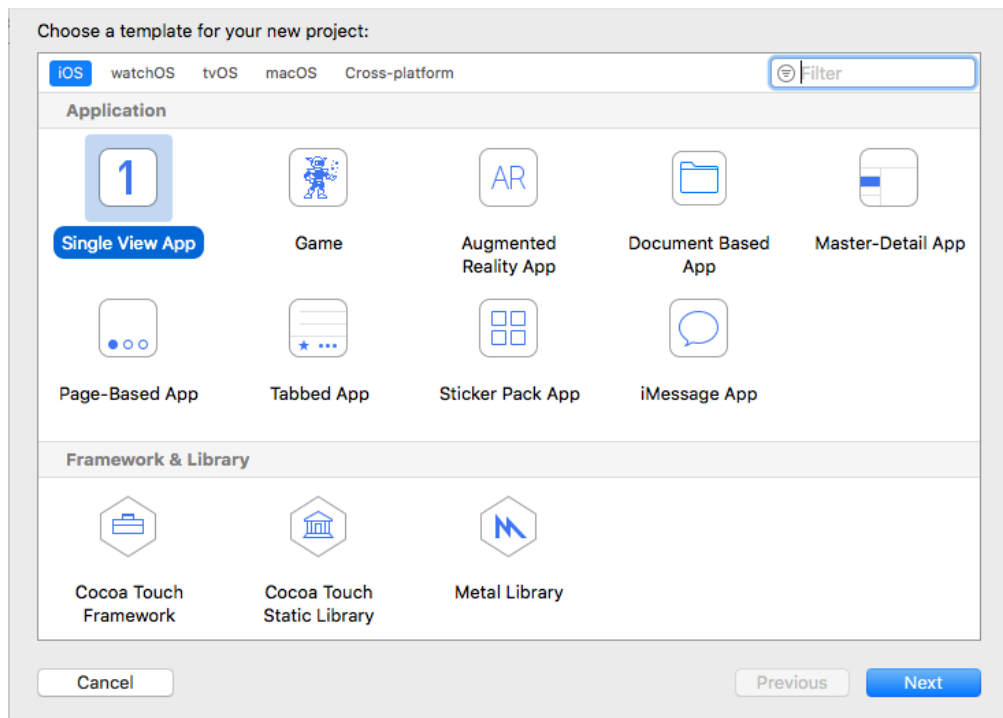
The screenshot shows a web browser window with the address bar displaying `https://loctest-64ed7.firebaseio.com`. The browser's address bar and tabs are visible at the top. The main content area displays a large JSON array of objects, each representing a station. Each object contains fields for region, name, tel, and add. The data is as follows:

```
{
  "allStations": [
    {
      "region": "台北市",
      "name": "重慶分校",
      "tel": "(02)2331-6960",
      "add": "台北市中正區重慶南路一段49號2F",
      "lat": 25.044976,
      "long": 121.513358
    },
    {
      "region": "台北市",
      "name": "台北認證",
      "tel": "(02)2382-6015",
      "add": "台北市中正區公園路30號3F",
      "lat": 25.044713,
      "long": 121.516866
    },
    {
      "region": "台北市",
      "name": "東區認證",
      "tel": "(02)2773-5243",
      "add": "台北市大安區忠孝東路四段169號4F之1",
      "lat": 25.041933,
      "long": 121.550362
    },
    {
      "region": "台北市",
      "name": "台大分校",
      "tel": "(02)2365-5366",
      "add": "台北市中正區羅斯福路三段312號5F",
      "lat": 25.016308,
      "long": 121.532648
    },
    {
      "region": "台北市",
      "name": "士林分校",
      "tel": "(02)2833-8011",
      "add": "台北市士林區中正路288號3樓",
      "lat": 25.095179,
      "long": 121.523529
    },
    {
      "region": "台北市",
      "name": "館前分校",
      "tel": "(02)2388-0177",
      "add": "台北市中正區館前路34號4樓",
      "lat": 25.044859,
      "long": 121.51486
    },
    {
      "region": "台北市",
      "name": "松山分校",
      "tel": "(02)2747-5569",
      "add": "台北市松山區八德路四段678號3F之1",
      "lat": 25.050176,
      "long": 121.576057
    },
    {
      "region": "台北市",
      "name": "內湖分校",
      "tel": "(02)2795-3745",
      "add": "台北市內湖區成功路四段50號2樓",
      "lat": 25.082489,
      "long": 121.592183
    },
    {
      "region": "新北市",
      "name": "板橋認證",
      "tel": "(02)2964-0123",
      "add": "新北市板橋區中山路一段10號5樓",
      "lat": 25.00817,
      "long": 121.460229
    },
    {
      "region": "新北市",
      "name": "三重分校",
      "tel": "(02)2972-0388",
      "add": "新北市三重區重新路二段60號",
      "lat": 25.062383,
      "long": 121.498016
    },
    {
      "region": "新北市",
      "name": "永和分校",
      "tel": "(02)2232-4619",
      "add": "新北市永和區中正路746號2F",
      "lat": 25.01062,
      "long": 121.515214
    },
    {
      "region": "新北市",
      "name": "新莊分校",
      "tel": "(02)2203-1186",
      "add": "新北市新莊區中正路287號",
      "lat": 25.035333,
      "long": 121.448802
    },
    {
      "region": "新北市",
      "name": "新店分校",
      "tel": "(02)2918-2595",
      "add": "新北市新店區北新路二段93號",
      "lat": 24.974339,
      "long": 121.542809
    },
    {
      "region": "新北市",
      "name": "淡水分校",
      "tel": "(02)8631-3113",
      "add": "新北市淡水區學府路30號5樓",
      "lat": 25.168892,
      "long": 121.447848
    },
    {
      "region": "新北市",
      "name": "樹林分校",
      "tel": "(02)2682-3259",
      "add": "新北市樹林區中山路一段39號",
      "lat": 24.992651,
      "long": 121.424083
    },
    {
      "region": "新北市",
      "name": "蘆洲分校",
      "tel": "(02)2282-3808",
      "add": "新北市蘆洲區中正路66號",
      "lat": 25.0835,
      "long": 121.47055
    },
    {
      "region": "新北市",
      "name": "板橋文化",
      "tel": "(02)2250-4608",
      "add": "新北市板橋區文化路一段349號",
      "lat": 25.021227,
      "long": 121.46658
    },
    {
      "region": "基隆市",
      "name": "基隆分校",
      "tel": "(02)2424-4642",
      "add": "基隆市仁愛區愛三路37號7F",
      "lat": 25.128297,
      "long": 121.74229
    },
    {
      "region": "桃園縣",
      "name": "中壢分校",
      "tel": "(03)427-3993",
      "add": "桃園縣中壢市中央東路55號",
      "lat": 24.957308,
      "long": 121.225503
    },
    {
      "region": "桃園縣",
      "name": "八德分校",
      "tel": "(03)366-8800",
      "add": "桃園縣八德市介壽路一段985號",
      "lat": 24.958981,
      "long": 121.299811
    },
    {
      "region": "桃園縣",
      "name": "南崁分校",
      "tel": "(03)366-8800",
      "add": "桃園縣八德市介壽路一段985號",
      "lat": 24.958981,
      "long": 121.299811
    }
  ]
}
```

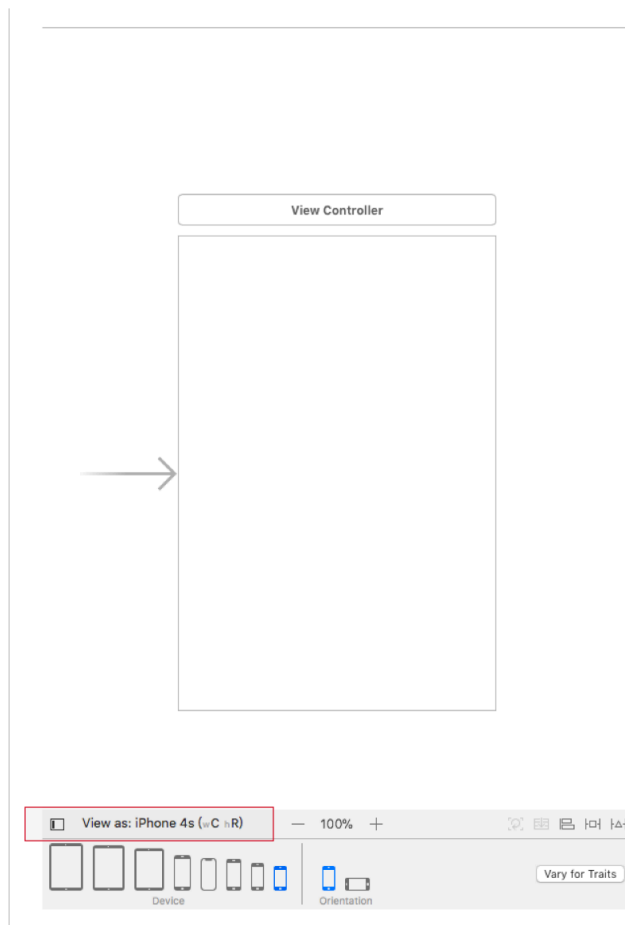
雲端json service建立完成。

建立JSONEncoder xcode專案

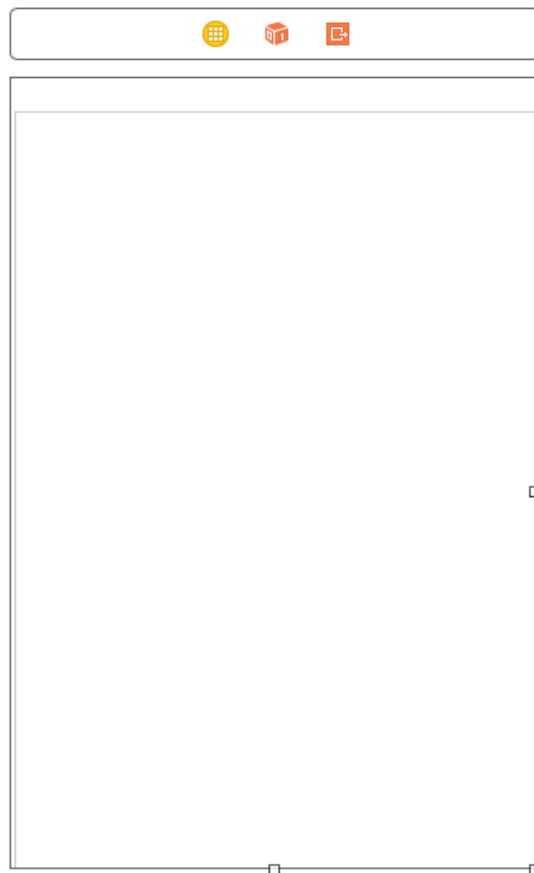
使用樣版Single View App，建立專案JSONEncoder的專案。



建立介面，切換畫面至 Main.storyboard，使用iphone 4s模式。



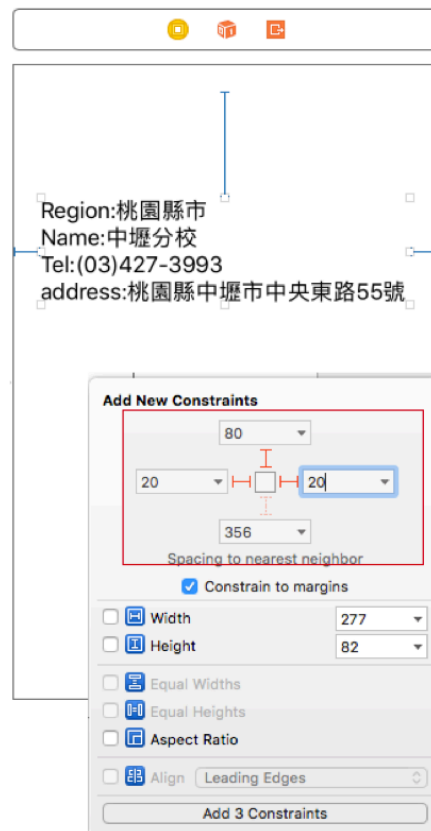
將storyboard的ViewController刪除，在介面圖庫拉出UICollectionViewController,並且將UICollectionViewCell拉到和View一樣大(大約);



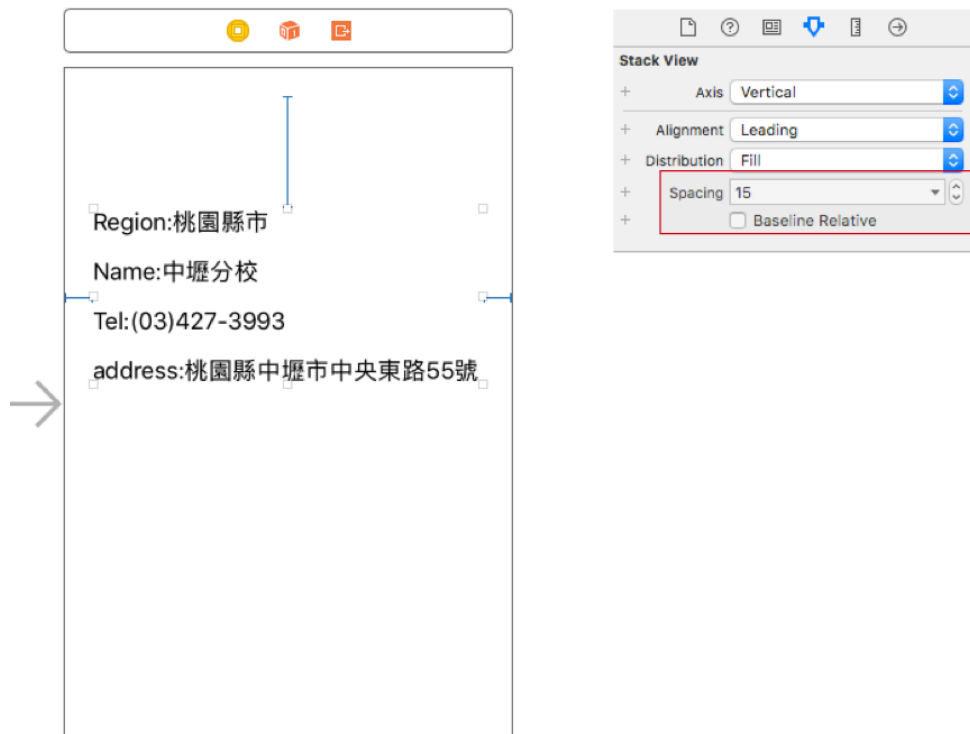
建立4個UILabel，並且加入假設文字，由上而下，依序排列。



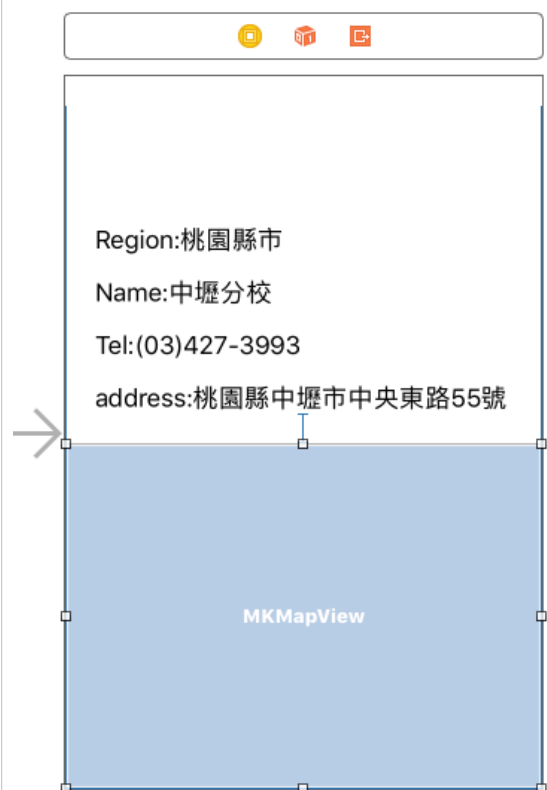
將4個Label加上StackView，並且在stackView加上上:80,左:20,右:20的constraints。



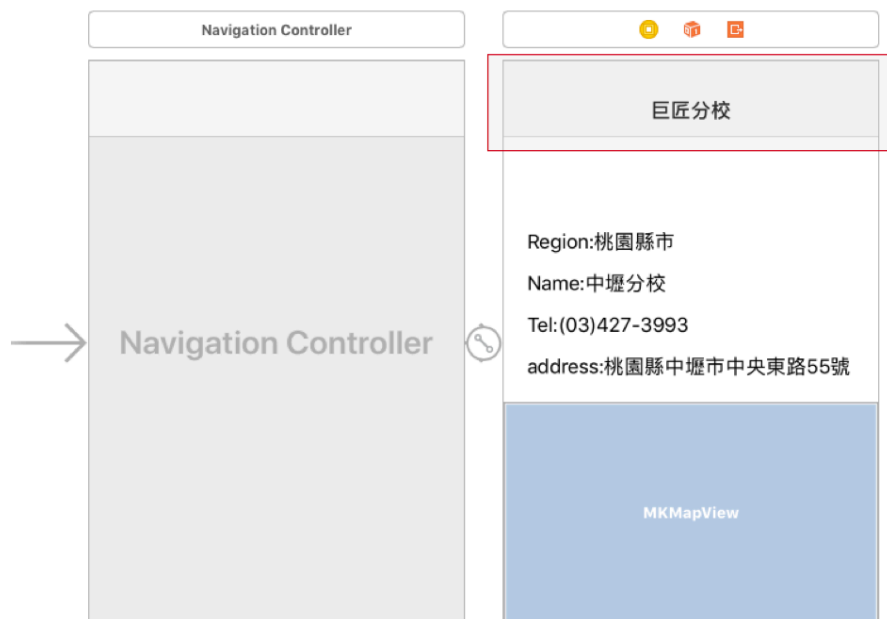
將stackView加上spacing 15。



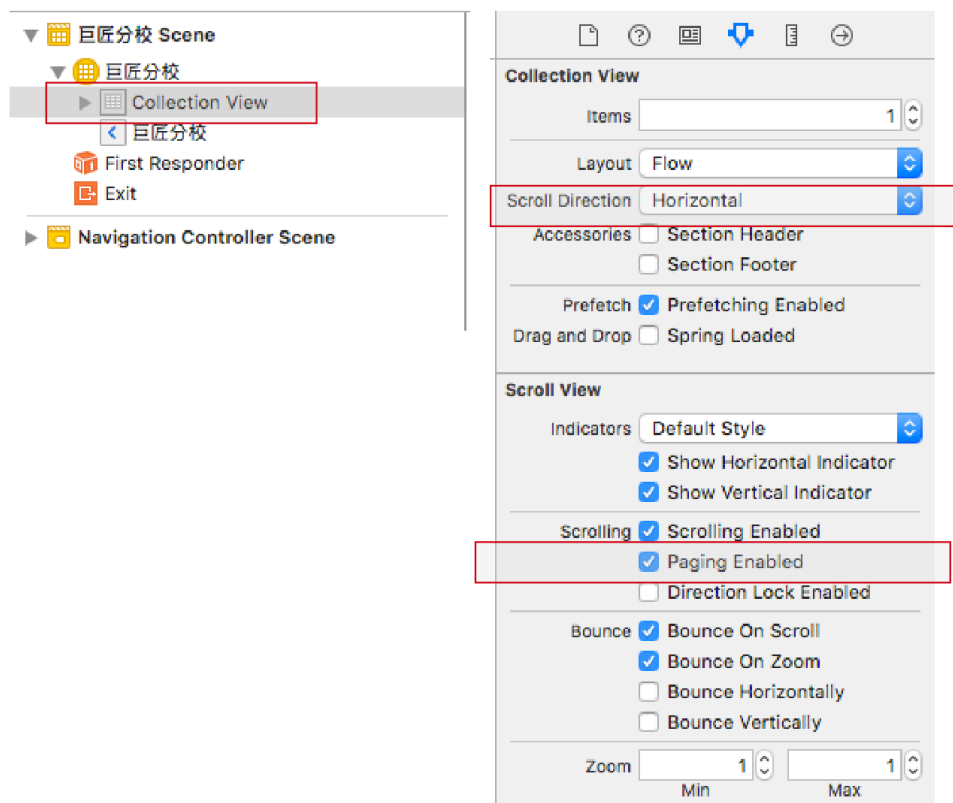
加入MKMapView，並且加上上:20,左:0,右:0,下:0。的constraints。



將ViewController加上UINavigationController和將ViewController的NavItem的title改為”巨匠分校”

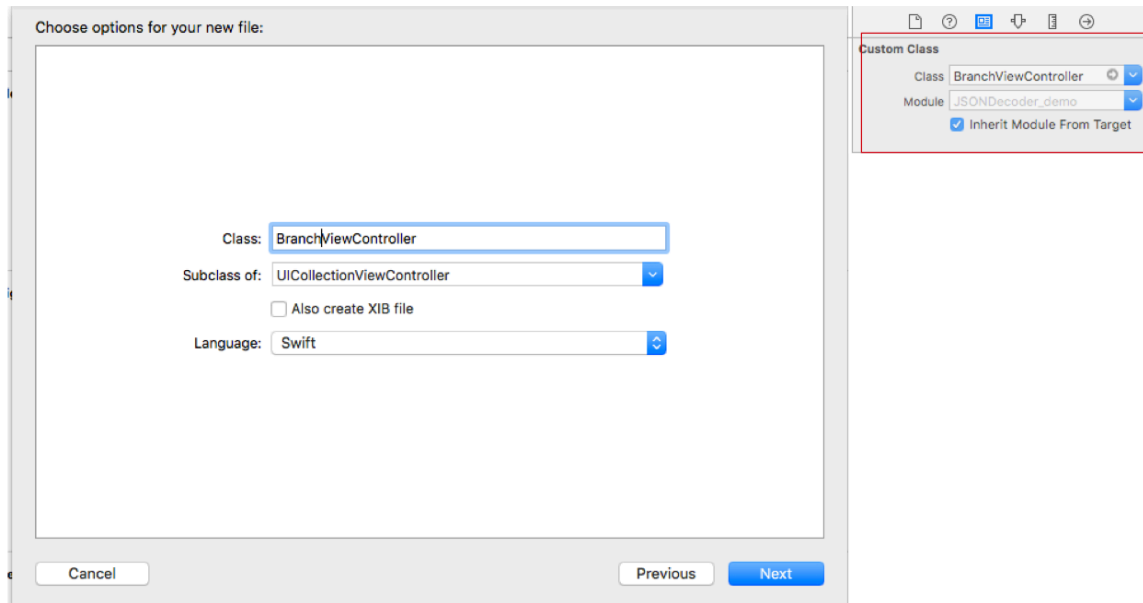


在storyboard內選取Collection View，並到屬性欄位，將Scroll Direction改為Horizontal，打開Paging Enabled。

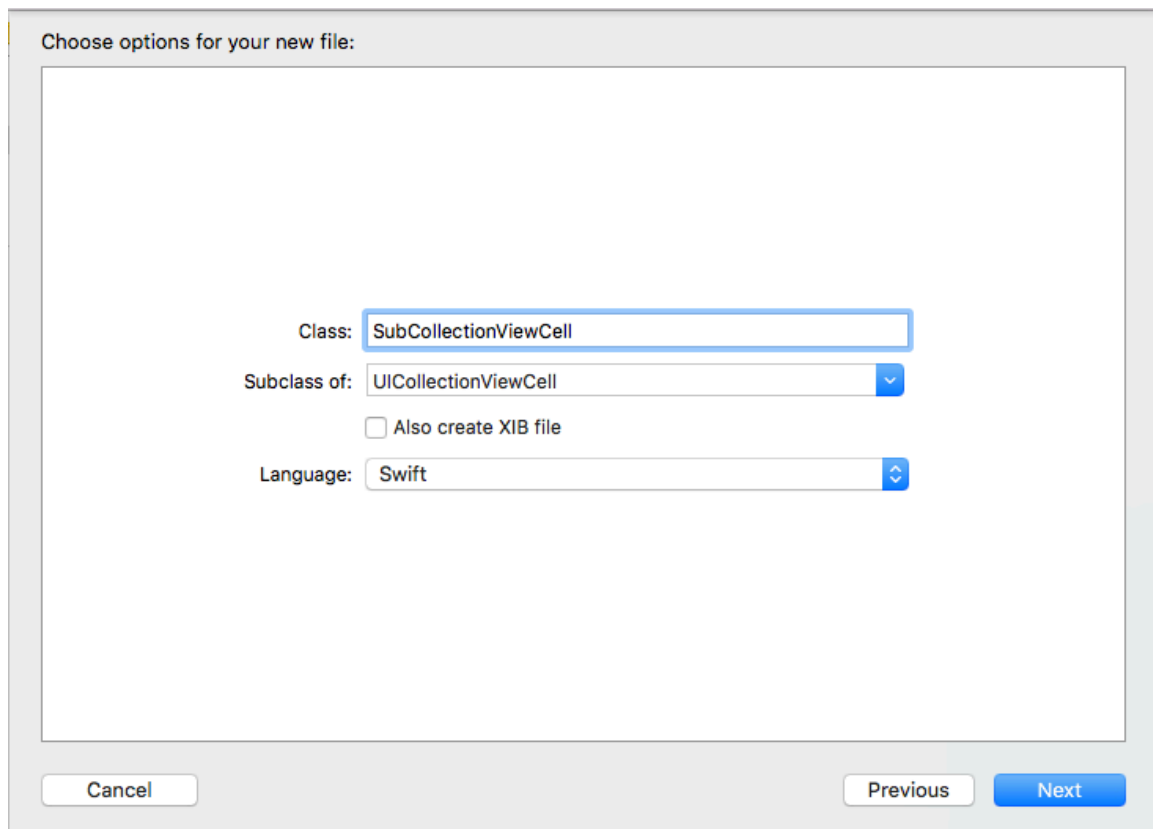


建立介面和程式碼的連結

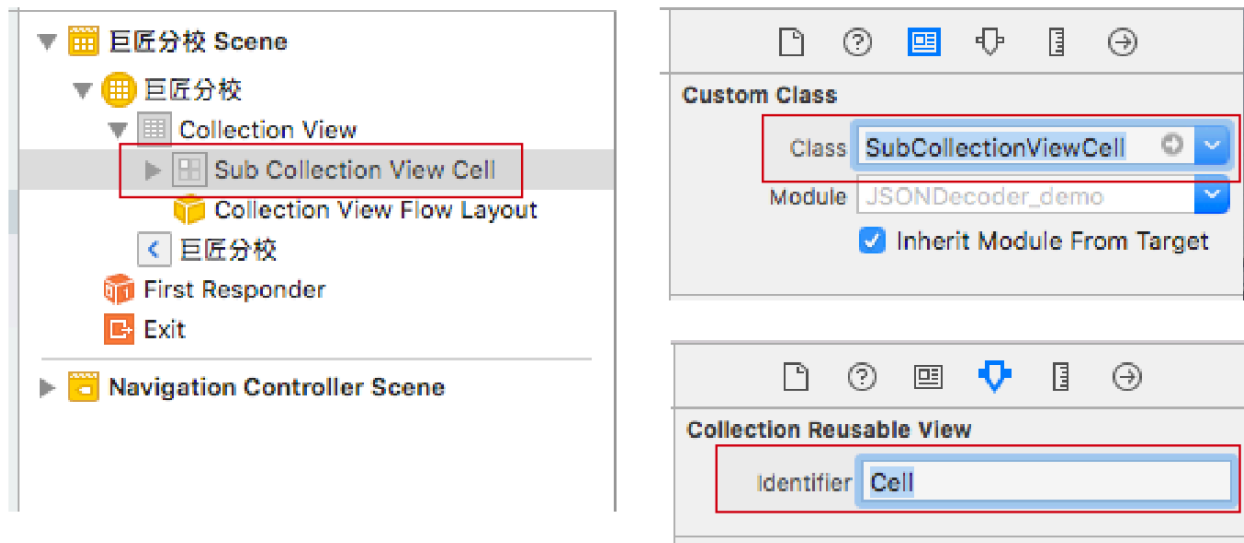
建立BranchViewController繼承UICollectionViewController。並將Storyboard內的UICollectionViewController置換為BranchViewController。



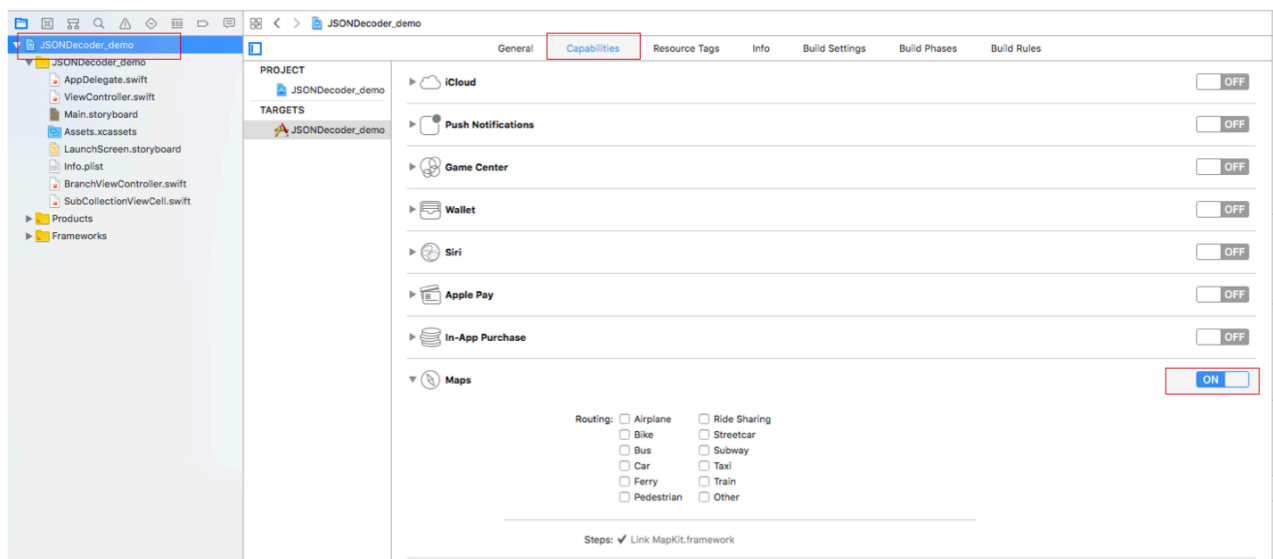
建立SubCollectionViewCell，繼承UICollectionViewCell。



將storyboard內的UICollectionViewCell的class改為SubCollectionViewCell。
reusable View改為Cell



點選專案，點選Capability，打開Maps的功能



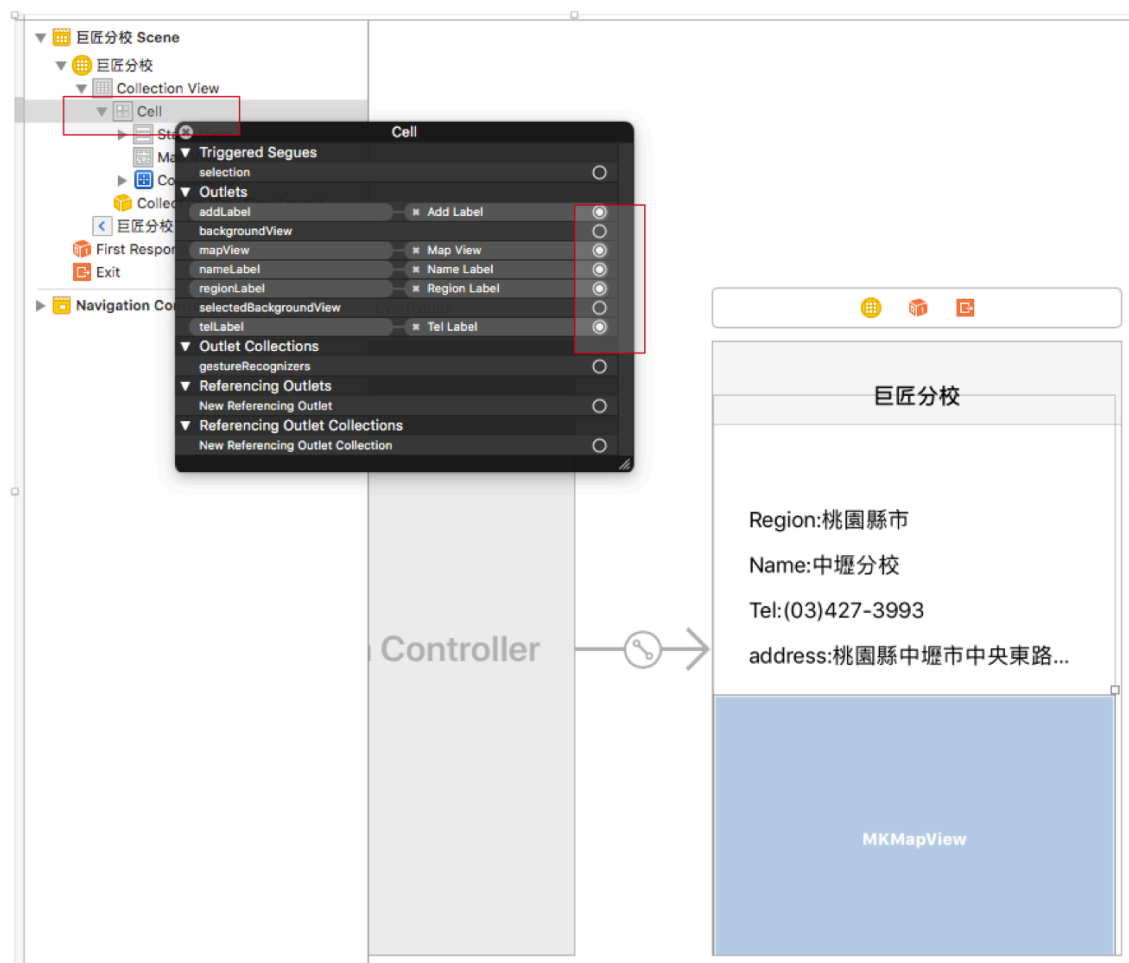
進入SubCollectionViewCell內，import MapKit

```
import MapKit
```

建立5個對應到Storyboard內的IBOutlet

```
class SubCollectionViewCell: UICollectionViewCell {  
    @IBOutlet weak var regionLabel: UILabel!  
    @IBOutlet weak var nameLabel: UILabel!  
    @IBOutlet weak var telLabel: UILabel!  
    @IBOutlet weak var addLabel: UILabel!  
    @IBOutlet weak var mapView: MKMapView!  
}
```

在storyboard內建立連結，點選cell按右鍵，將5個IBOutlet連結至畫面元件



打開BranchViewController.swift，取得UICollectionView內的UICollectionViewFlowLayout，並將itemSize的大小，改為和BranchViewController Viewd的大小一樣。

```
let collectionViewFlowLayout = self.collectionViewLayout as!
UICollectionViewFlowLayout;
collectionViewFlowLayout.itemSize = self.view.bounds.size;
```

Import MakKit

```
import MapKit
```

建立屬性 urlSession準備連線internet，下載json檔。

```
var urlSession:URLSession!;
```

在ViewDidLoad的時間點，建立URL實體，建立urlSession的實體，使用ulrSession建上下載任務，並且要求立即下載。

```
//建立連線
guard let url = URL(string: "https://iostest-64ed7.firebaseio.com/")
else {
    return;
}
urlSession = URLSession.shared;
let downloadTask = urlSession.downloadTask(with: url) { (url:URL?,
response:URLResponse?, error:Error?) in
    }
downloadTask.resume();
```

建立下載任務的Closure，檢查連線上載所有有可能的錯誤。取得後print出下載的字串是否正確。

```
guard let url = url, let response = response else {
    return;
}

guard error == nil else {
    return;
}

guard (response as! HTTPURLResponse).statusCode == 200 else {
    return;
}

guard let data = try? Data.init(contentsOf: url) else {
    return;
}

print(String.init(data: data, encoding: String.Encoding.utf8!));
```

建立下載json轉換的結構 Allstation，必需採納protocol Codable。

```
struct AllStation:Codable{
    struct Station:Codable{
        let region:String;
        let name:String;
        let tel:String;
        let add:String;
        let lat:Double;
        let long:Double;
    }
    let allStations:[Station];
}
```

在BranchViewController內建立屬性AllStation

```
var allStation:AllStation!;
```

建立JSONDecoder，轉換JSON資料成為allStation實體。

```
//轉換json為AllStation的實體  
let jsonDecoder = JSONDecoder();  
self.allStation = try? jsonDecoder.decode(AllStation.self, from:  
data);
```

進入主執行序，要求CollectionView重新載入資料

```
//進入主執行序，要求CollectionView重新載入資料  
DispatchQueue.main.sync {  
    self.collectionView?.reloadData();  
}
```

實作 UICollectionViewDataSource protocol

```
override func collectionView(_ collectionView: UICollectionView,  
numberOfItemsInSection section: Int) -> Int {  
    // #warning Incomplete implementation, return the number of items  
    return 0  
}
```

在BranchViewController內建立keyPath

```
let allStationsPath = \BranchViewController.allStation.allStations;
```


在collectionView(␣, numberOfItemsInSection)內告知要顯示Cell的數量。

```
if allStation == nil {  
    return 0;  
}else{  
    return self[keyPath:allStationsPath].count;  
}
```

實作 collectionView(␣, cellForItemAt)，並取得station實體，將station資料顯示在介面元件上。

```
let cell = collectionView.dequeueReusableCell(withReuseIdentifier: "Cell",  
for: indexPath) as! SubCollectionViewCell;  
  
//取得Station的實體  
let station = self[keyPath:allStationsPath][indexPath.row];  
  
cell.titleLabel?.text = "地址:\(station.add)";  
  
cell.nameLabel?.text = "分校名:\(station.name)";  
  
cell.regionLabel?.text = "地區:\(station.region)";  
  
cell.telLabel?.text = "電話:\(station.tel)";
```

取得MapView和顯示MapView的內容。

```
//建立MapView和顯示座標內容
let mapView:MKMapView = cell.mapView;

let annotation = MKPointAnnotation();

let coordinate = CLLocationCoordinate2D(latitude: station.lat,
longitude: station.long);

annotation.coordinate = coordinate;

mapView.addAnnotation(annotation);

let region = MKCoordinateRegionMakeWithDistance(coordinate, 250,
250);

mapView.setRegion(region, animated: false);
```