

# iUnitTestはじめの一歩

## インストール

Terminal.appで以下の手順でインストールします。

事前にgitをインストールしている場合。

```
> git clone git://github.com/katsuyoshi/iunittest.git  
> cd iunittest  
> sudo ./setup
```

gitをインストールしていない場合は、tar ballをダウンロードし、展開したディレクトリでsudo ./setupを実行して下さい

## チュートリアル

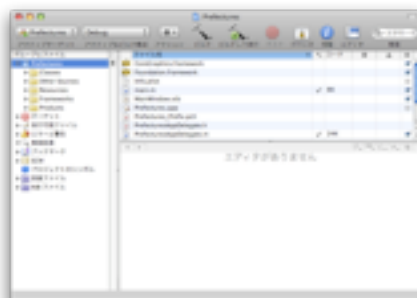
ここでは、都道府県名をUITableViewに表示するアプリケーションを想定します。ターゲットとなるプロジェクトと、テスト用のプロジェクトの2本立てで進めます。

### ターゲットプロジェクト作成

まずXcodeでiPhone OSのApplicationからNavigation-Based Applicationテンプレートを選択し、新規プロジェクトを作成します。



Prefecturesという名称にします。



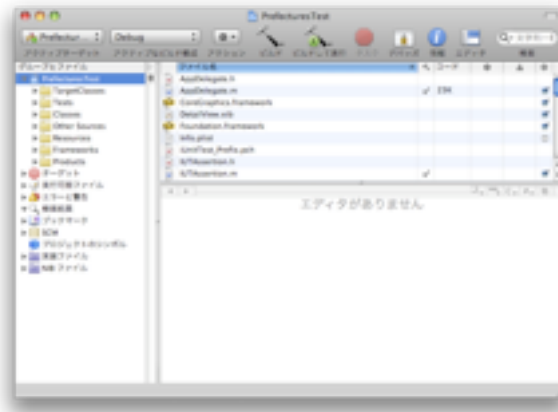
## テストプロジェクト作成

iUnitTestをインストールすると、UserTemplatesのApplicationにiUnitTest Applicationが追加されています。

iUnitTest Applicationを選択し、新規プロジェクトを作成します。



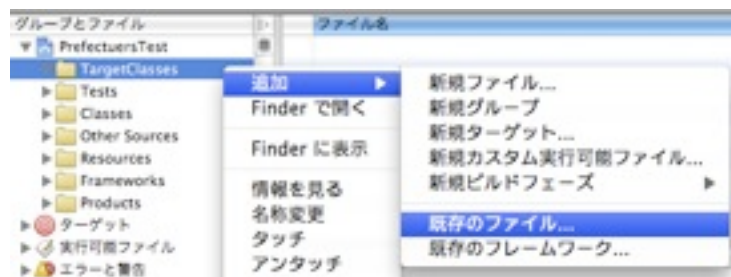
プロジェクト名はPrefecturesTestにします。

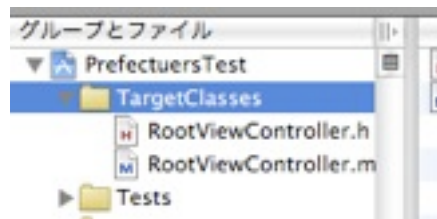
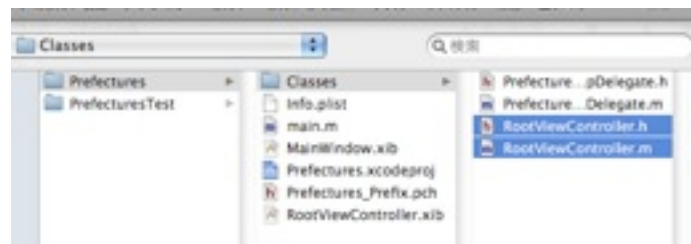


## テストプロジェクトへテスト対象のソースファイルを追加

PrefecturesTestのTargetClassesグループにPrefecturesの RootViewController.[m,h]を追加します。

TargetClassesグループでコンテキストメニューを表示させ、追加 -> 既存のファイル...を選択して、Prefecturesプロジェクト内のRootViewController.[m,h]を指定し、追加します。





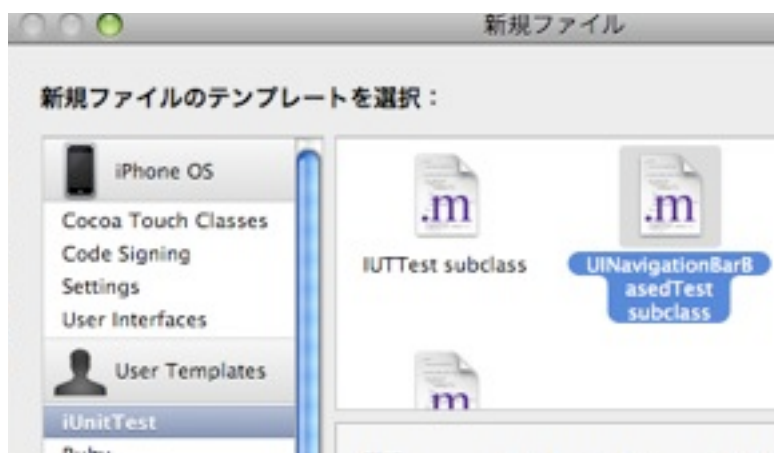
## テストクラスの追加

RootViewControllerクラスのテストを行なうRootViewControllerTestクラスを追加します。

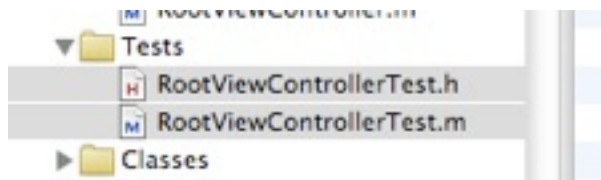
Testsグループでコンテキストメニューを表示させ、追加 -> 新規ファイル...を選択します。



iUnitTestをインストールすると、UserTemplateの中にiUnitTestが追加されています。この中からUINavigationControllerBasedTest subclassを選択しテストクラスを作成します。



RootViewControllerTestという名称にします。



iUnitTestはUITestクラスのサブクラスでtestから始まるmethodを自動的に実行します。 UINavigationControllerBasedTestはUITestクラスのサブクラスになっています。

## 初めのテスト

PrefecturesTestプロジェクトのRootViewControllerTest.hを以下の様に編集します。黄色になっている行が変更箇所です。

```
#import <Foundation/Foundation.h>
#import "UINavigationControllerBasedTest.h"
#import "RootViewController.h"

@interface RootViewControllerTest : UINavigationControllerBasedTest {

    // TODO: replace your view controller
    RootViewController *controller;

}

@end
```

RootViewControllerTest.mの-rootViewControllerを以下の様に編集します。

```
- (UIViewController *)rootViewController
{
    // this controller will be tested.
    // TODO: replace your view controller
    controller = [[RootViewController alloc] initWithStyle:UITableViewStylePlain];
    return controller;
}
```

都合上北海道と東北のみを表示する仕様にします。

RootViewController.mの- (NSInteger)numberOfSectionsInTableView:  
(UITableView \*)tableView

では北海道と東北の2セクションを返す事になりその事をテストします。

-test1を削除し、-testNumberOfSectionsInTableViewを追加します。  
ASSERT\_EQUAL\_INTは整数の比較をします。

```
- (void)testNumberOfSectionsInTableView
{
    ASSERT_EQUAL_INT(2, [controller
numberOfSectionsInTableView:controller.tableView]);
}
```

ここで一旦実行してみます。

アプリケーションが起動したらStartボタンを押します。

testNumberOfSectionsInTableViewで失敗した事が表示されます。



ここでRootViewControllerの実装をします。

RootViewController.hにデータを保持するdataSourceを追加します。

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

@interface RootViewController : UITableViewController {
    NSArray *dataSource;
}

@end
```

RootViewController.mの-didLoadViewにdataSourceの初期化コードを記述します。

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

    dataSource = [[NSArray alloc] initWithObjects:
        [NSArray arrayWithObject:@"Hokkaido"],
        [NSArray arrayWithObjects:
            @"Aomori",
            @"Iwate",
            @"Akita",
            @"Yamagata",
            @"Miyagi",
            @"Fukushima",
            nil],
        nil];
}
```

deallocで解放します。

```
- (void)dealloc {
    [dataSource release];
    [super dealloc];
}
```

- (NSInteger)numberOfSectionsInTableView:(UITableView \*)tableViewを実装します。

```
- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView {
    return [dataSource count];
}
```

ここでアプリケーションを実行します。

今度は成功しました。



## 残りのテストの実装

残りのテストもRootViewControllerTestに実装します。

ASSERT\_EQUALはオブジェクトの比較をします。

```
- (void)testTitleForHeaderInSection
{
    ASSERT_EQUAL(@"Hokkaido", [controller tableView:controller.tableView
titleForHeaderInSection:0]);
    ASSERT_EQUAL(@"Tohoku", [controller tableView:controller.tableView
titleForHeaderInSection:1]);
}

- (void)testNumberOfRowsInSection
{
    ASSERT_EQUAL_INT(1, [controller tableView:controller.tableView
numberOfRowsInSection:0]);
    ASSERT_EQUAL_INT(6, [controller tableView:controller.tableView
numberOfRowsInSection:1]);
}

- (void)testCellForRowAtIndexPath
{
    ASSERT_EQUAL(@"Hokkaido", [[controller tableView:controller.tableView
cellForRowAtIndexPath:[NSIndexPath indexPathForRow:0 inSection:0]] text]);
    ASSERT_EQUAL(@"Aomori", [[controller tableView:controller.tableView
cellForRowAtIndexPath:[NSIndexPath indexPathForRow:0 inSection:1]] text]);
    ASSERT_EQUAL(@"Iwate", [[controller tableView:controller.tableView
cellForRowAtIndexPath:[NSIndexPath indexPathForRow:1 inSection:1]] text]);
    ASSERT_EQUAL(@"Akita", [[controller tableView:controller.tableView
cellForRowAtIndexPath:[NSIndexPath indexPathForRow:2 inSection:1]] text]);
    ASSERT_EQUAL(@"Yamagata", [[controller tableView:controller.tableView
cellForRowAtIndexPath:[NSIndexPath indexPathForRow:3 inSection:1]] text]);
    ASSERT_EQUAL(@"Miyagi", [[controller tableView:controller.tableView
cellForRowAtIndexPath:[NSIndexPath indexPathForRow:4 inSection:1]] text]);
    ASSERT_EQUAL(@"Fukushima", [[controller tableView:controller.tableView
cellForRowAtIndexPath:[NSIndexPath indexPathForRow:5 inSection:1]] text]);
}
```



## 残りのコードの実装

RootViewControllerも実装します。

```
- (NSString *)tableView:(UITableView *)tableView titleForHeaderInSection:
(NSInteger)section {
    if (section == 0) {
        return @"Hokkaido";
    } else {
        return @"Tohoku";
    }
}

// Customize the number of rows in the table view.
- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:
(NSInteger)section {
    return [[dataSource objectAtIndex:section] count];
}

// Customize the appearance of table view cells.
- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:
(NSIndexPath *)indexPath {

    static NSString *CellIdentifier = @"Cell";

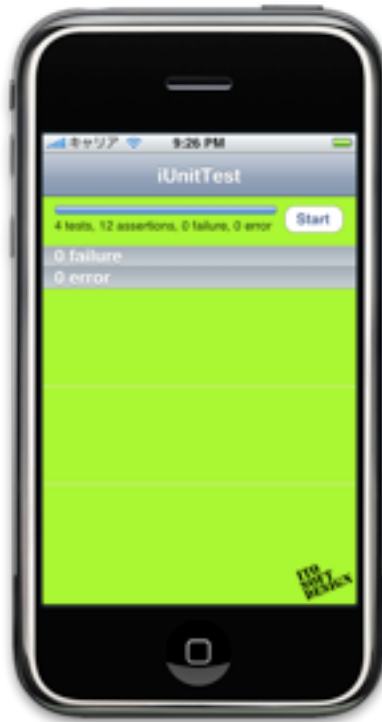
    UITableViewCell *cell = [tableView
    dequeueReusableCellWithIdentifier:CellIdentifier];
    if (cell == nil) {
        cell = [[[UITableViewCell alloc] initWithFrame:CGRectZero
        reuseIdentifier:CellIdentifier] autorelease];
    }

    // Set up the cell...
    cell.text = [[dataSource objectAtIndex:indexPath.section]
    objectAtIndex:indexPath.row];
    return cell;
}
```

## テスト実行

テストコードと、ソースコードの実装が終わったのでPrefecturesTestをテストします。

Startボタンを押すと、全てのテストが通った事が表示されました。



## ターゲットアプリケーションの動作確認

テストが通ったので、Prefecturesプロジェクトに戻り実行してみます。

予定通り北海道と東北の2つのセクションにそれぞれの都道府県が表示されています。

