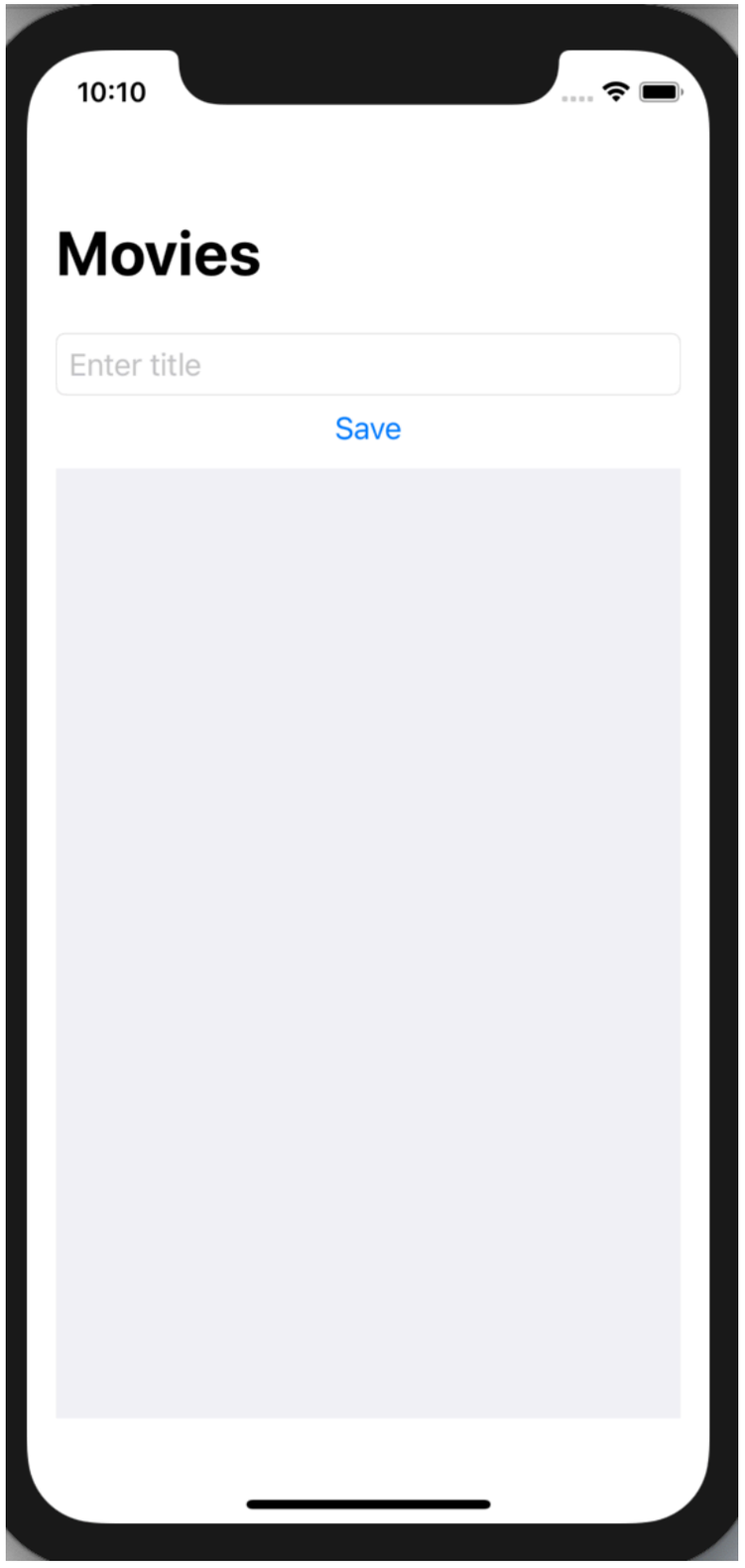
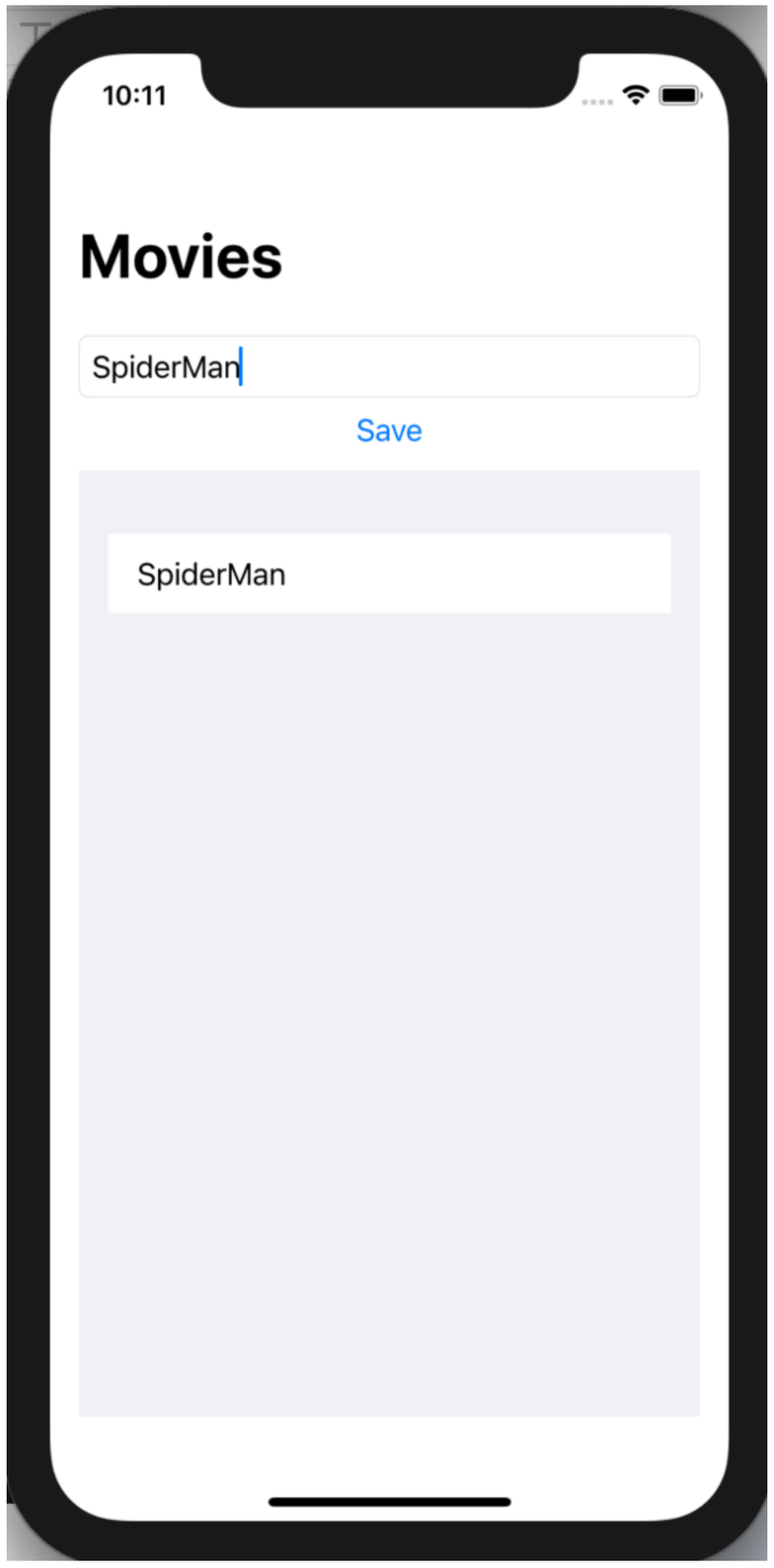
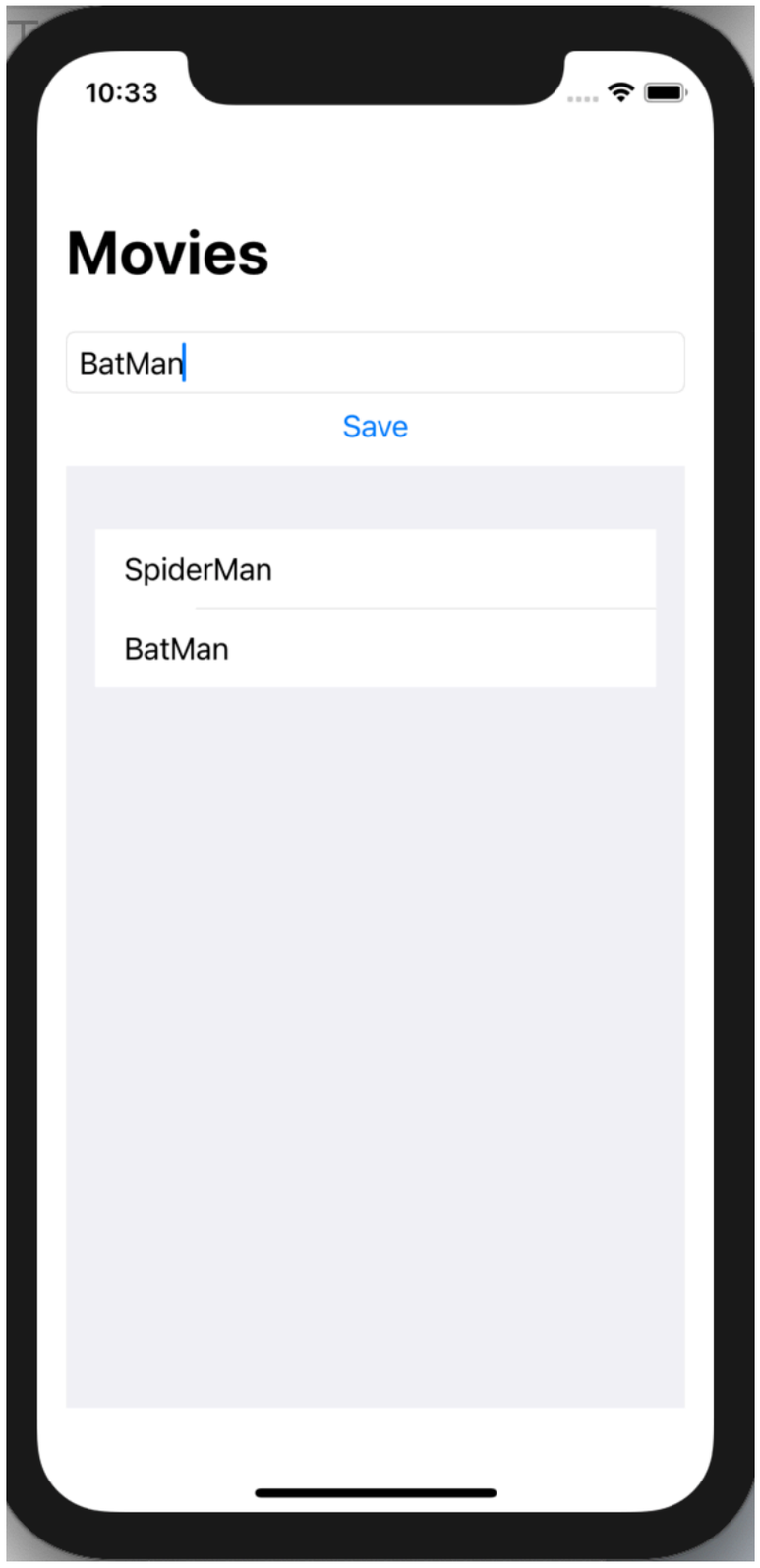
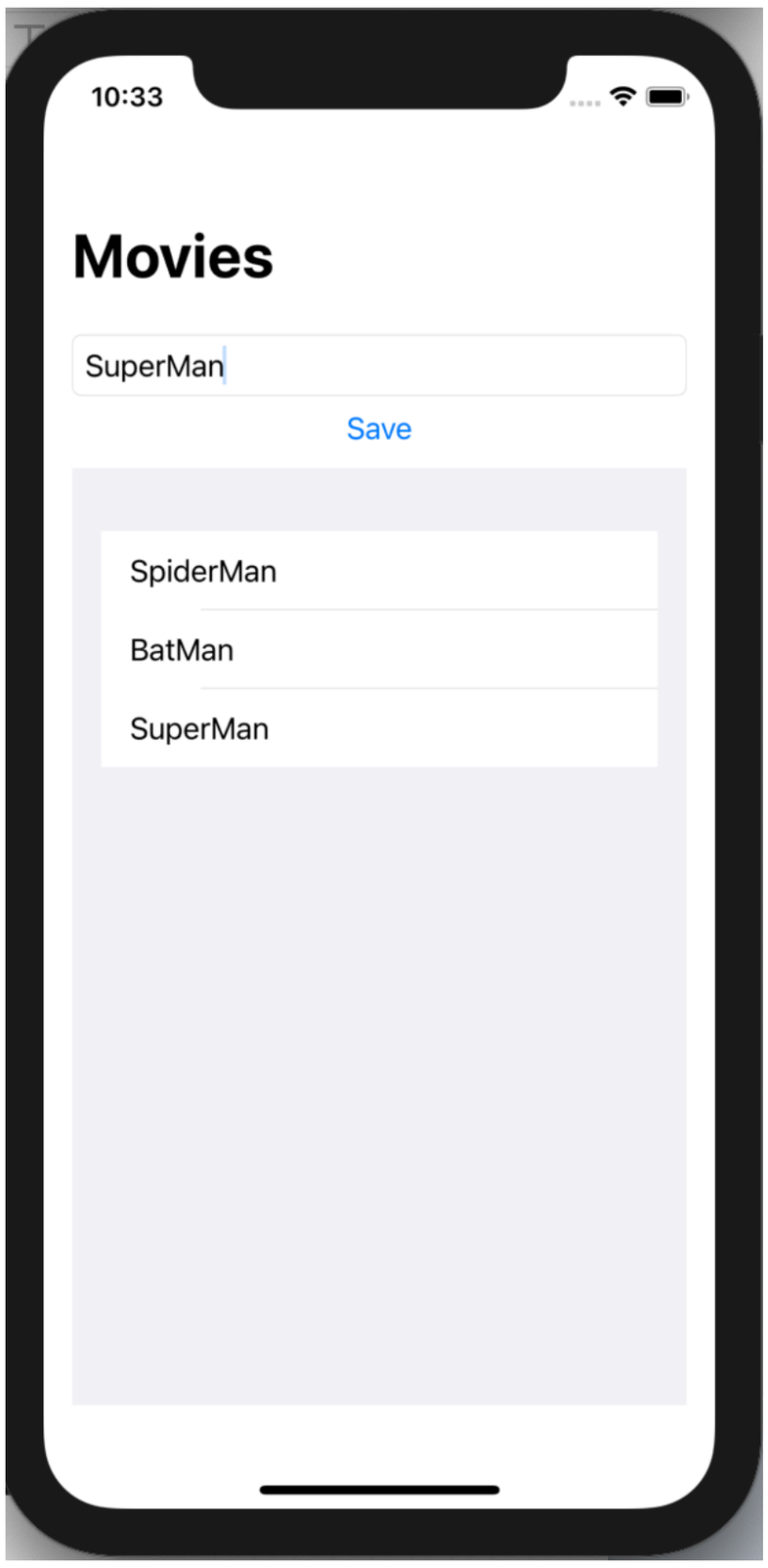
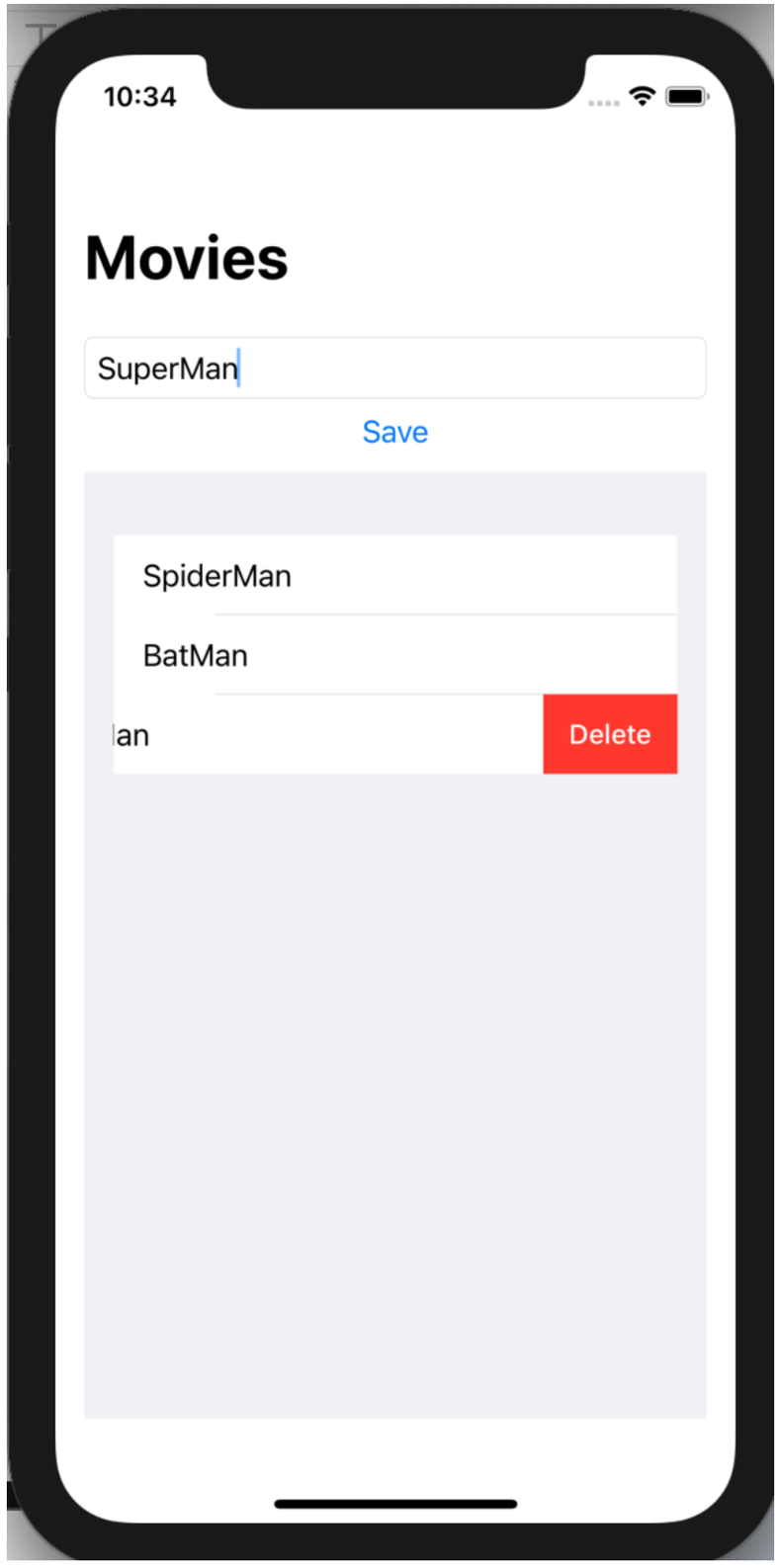
List 與 CoreData
資料建立與刪除

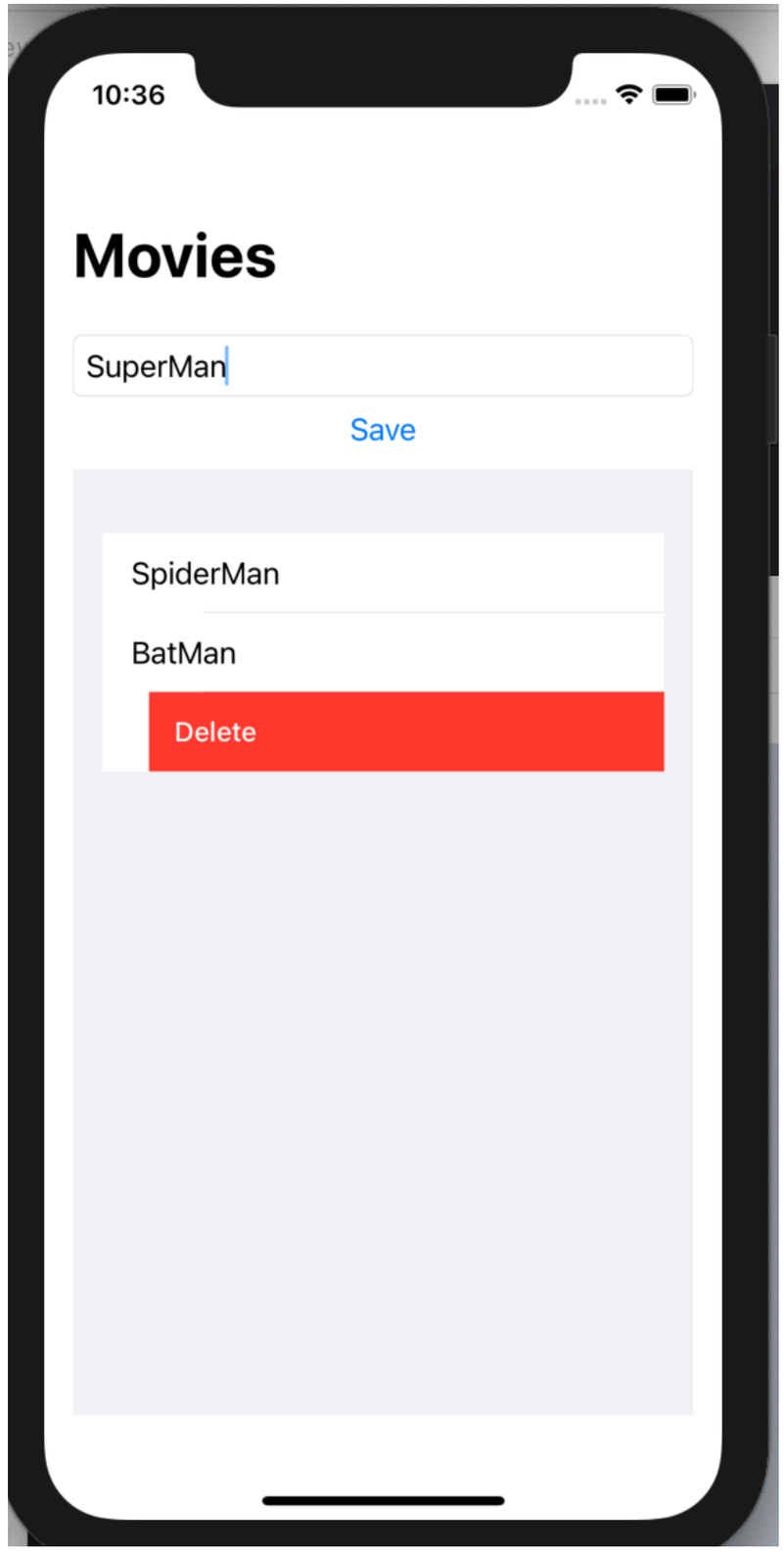

10:10
Movies
Enter title
Save

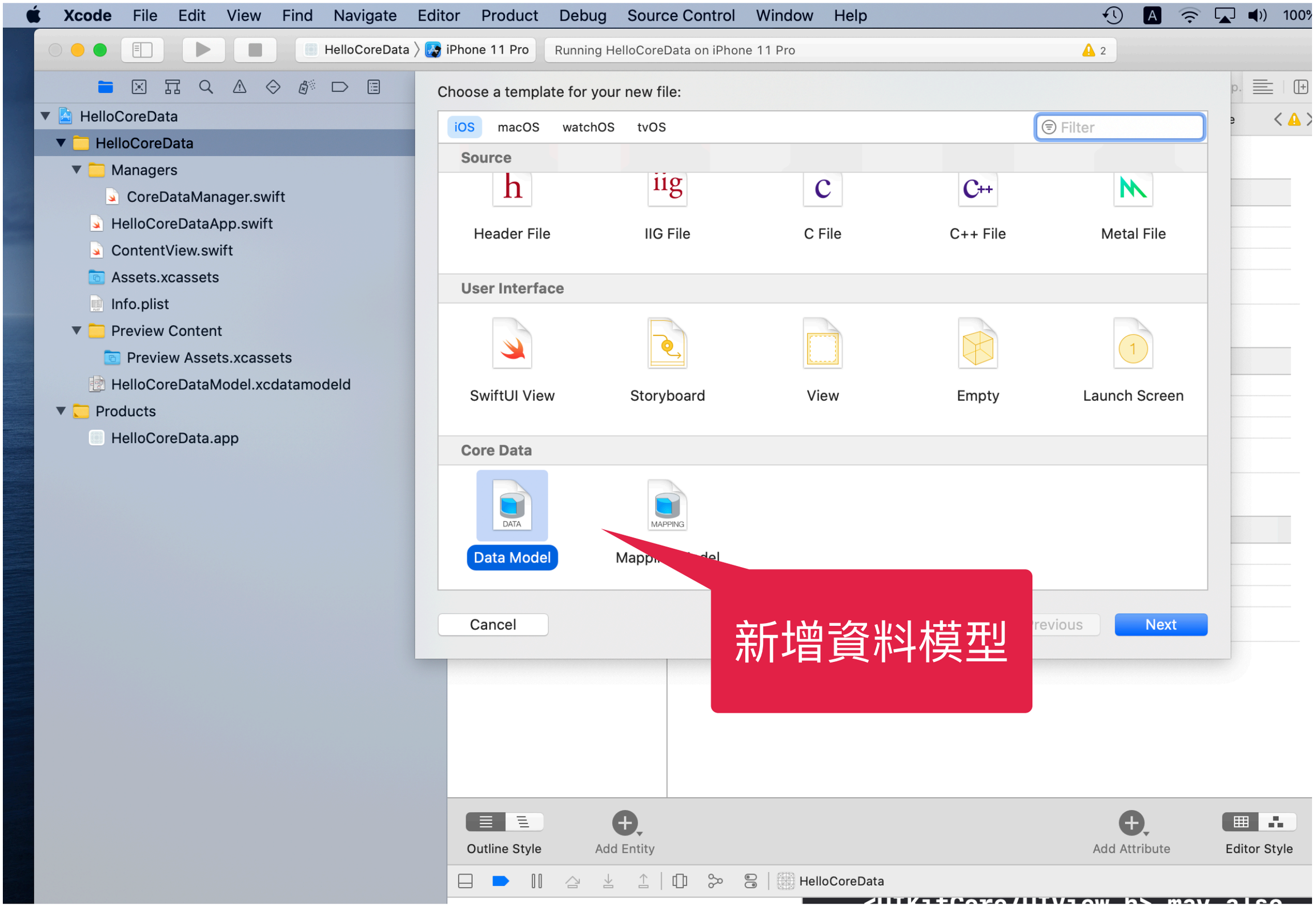

10:11
Movies
SpiderMan
Save
SpiderMan


10:33
Movies
BatMan
Save
SpiderMan
BatMan


10:33
Movies
SuperMan
Save
SpiderMan
BatMan
SuperMan


10:34
Movies
SuperMan
Save
SpiderMan
BatMan
an
Delete


10:36
Movies
SuperMan
Save
SpiderMan
BatMan
Delete


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
HelloCoreData iPhone 11 Pro
Running HelloCoreData on iPhone 11 Pro
HelloCoreData
HelloCoreData
Managers
CoreDataManager.swift
HelloCoreDataApp.swift
ContentView.swift
Assets.xcassets
Info.plist
Preview Content
Preview Assets.xcassets
HelloCoreDataModel.xcdatamodeld
Products
HelloCoreData.app
Choose a template for your new file:
iOS
macOS
watchOS
tvOS
Filter
Source
Header File
IIG File
C File
C++ File
Metal File
User Interface
SwiftUI View
Storyboard
View
Empty
Launch Screen
Core Data
DATA
Data Model
MAPPING
Cancel
Next
新增資料模型
Outline Style
Add Entity
Add Attribute
Editor Style
HelloCoreData

File Edit View Find Navigate Editor Product Debug elp 100% 週日 下午

HelloCoreData 〉 iPhone 11 Pro Pro 2

資料實體

ntView.swift HelloCoreD...cdatamodel HelloCoreDataApp.

HelloCoreD eDataModel.xcdatamodeld 〉 HelloCoreDataModel.xcdatamodel 〉 E Movie

HelloCoreData M
HelloCoreData
Managers
CoreDataManager.swift A
HelloCoreDataApp.swift M
ContentView.swift M
Assets.xcassets
Info.plist
Preview Content
Preview Assets.xcassets
HelloCoreDataModel.xcdatamodeld M
Products
HelloCoreData.app

核資料模型

ENTITIES
E Movie
FETCH REQUESTS
CONFIGURATIONS
C Default

Attributes

Attribute	Type
S title	String

屬性欄位

Relationships

Relationship	Destination	Inver

Fetched Properties

Fetched Property	Predicate

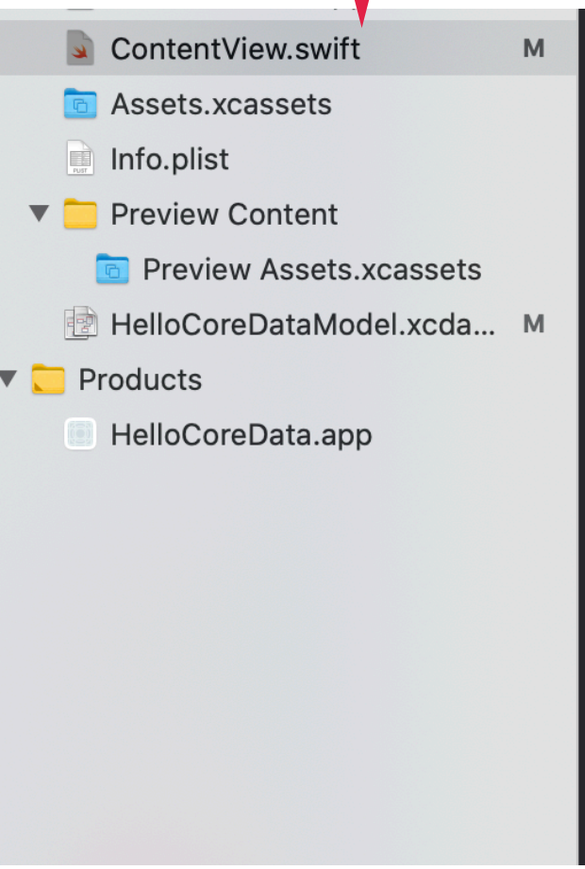
Outline Style Add Entity Add Attribute Editor Style

HelloCoreData

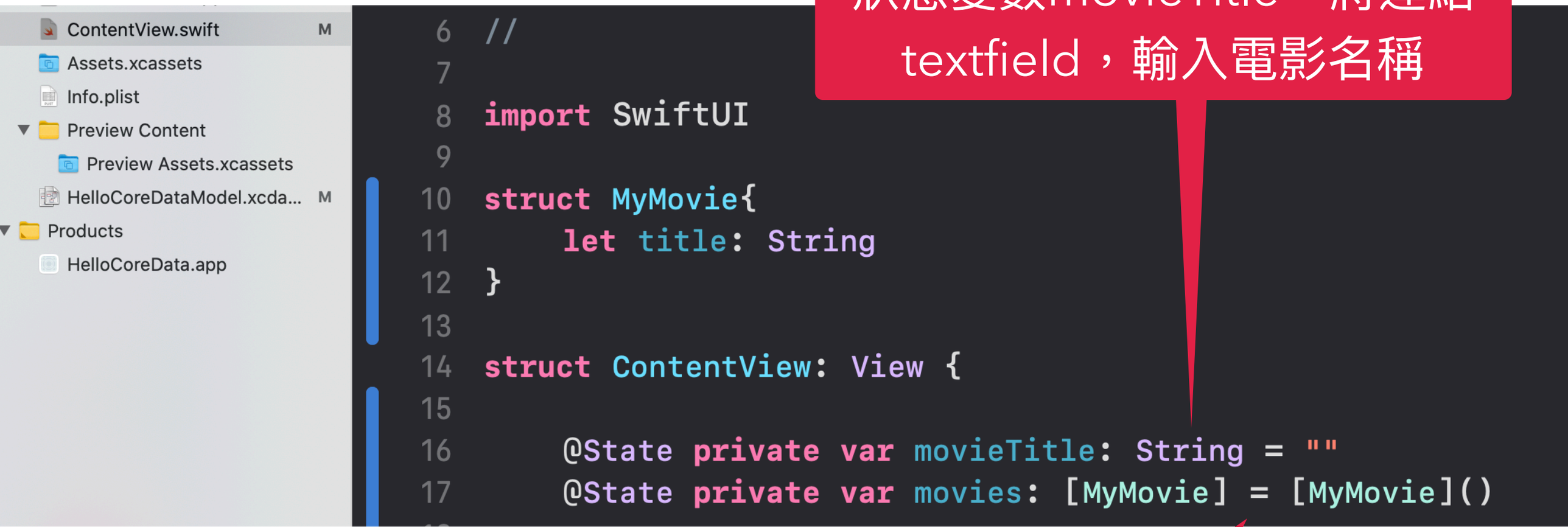
contentview

本題先使用MyMovie，將進一步改為資料模型中的Movie類別

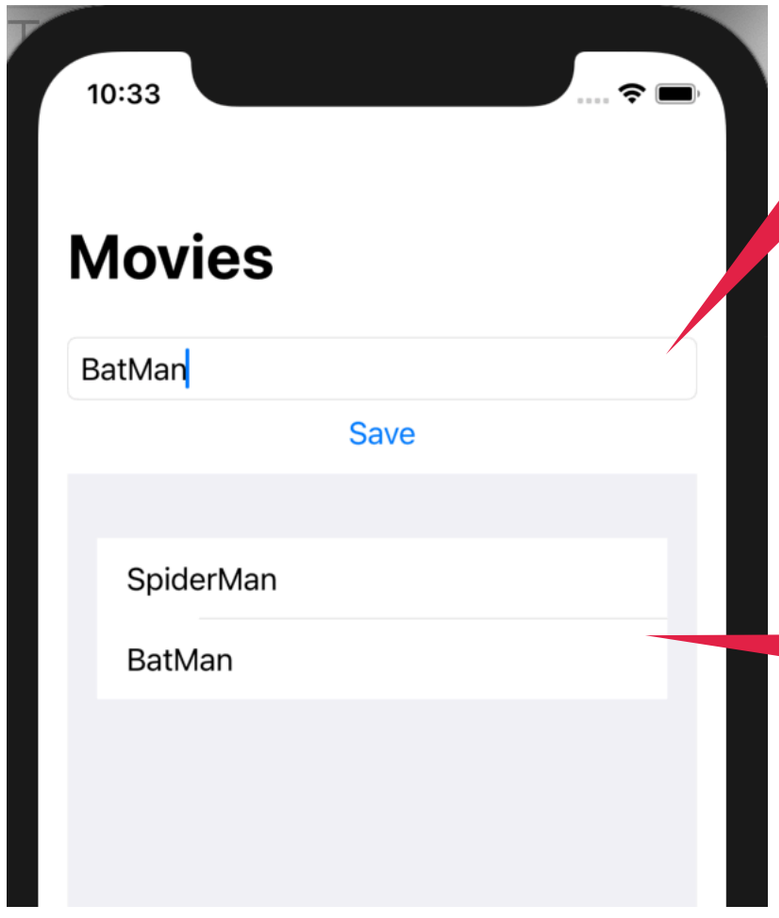
結構 struct MyMovie




```
6  //
7
8  import SwiftUI
9
10 struct MyMovie{
11     let title: String
12 }
13
14 struct ContentView: View {
15
16     @State private var movieTitle: String = ""
17     @State private var movies: [MyMovie] = [MyMovie]()
```


ContentView.swift M
Assets.xcassets
Info.plist
Preview Content
Preview Assets.xcassets
HelloCoreDataModel.xcda... M
Products
HelloCoreData.app
6 //
7
8 import SwiftUI
9
10 struct MyMovie{
11 let title: String
12 }
13
14 struct ContentView: View {
15
16 @State private var movieTitle: String = ""
17 @State private var movies: [MyMovie] = [MyMovie]()
狀態變數movieTitle，將連結textfield，輸入電影名稱
宣告串列movie，包含的物件結構為MyMovie，初始設定為空串列

```
13
14  struct ContentView: View {
15
16      @State private var movieTitle: String = ""
17      @State private var movies: [MyMovie] = [MyMovie]()
```



狀態變數movieTitle，將連結textfield，輸入電影名稱

串列movie，包含的物件結構為MyMovie，將連結到List的表列內容

導航view

使用NavigationView

```
struct ContentView: View {

    @State private var movieTitle: String = ""
    @State private var movies: [MyMovie] = [MyMovie]()


    var body: some View {
        NavigationView {
            VStack {

            }.padding()
             .navigationTitle("Movies")
        } // navigation view
    } //some view
} // contentview
```

```
    var body: some View {
        NavigationView {
            VStack {
                TextField("Enter title", text: $movieTitle)
                    .textFieldStyle(RoundedBorderTextFieldStyle())
                Button("Save"){


                }
                List{

                }
```

使用垂直堆疊

輸入文字連結
movieTitle

儲存按鍵

```
    var body: some View {
        NavigationView {
            VStack {
                TextField("Enter title", text: $movieTitle)
                    .textFieldStyle(RoundedBorderTextFieldStyle())
                Button("Save"){

                    saveMovie(title: movieTitle)
                }
                List{

                }
```

執行方法
saveMovie

```
struct ContentView: View {

    @State private var movieTitle: String = ""
    @State private var movies: [MyMovie] = [MyMovie]()

    private func saveMovie(title: String){
        let movie = MyMovie(title: title)
        movies.append(movie)
    }

    var body: some View {
```

將常數movie附加在串列movies中

宣告常數movie，結構為MyMovie，初始化設定為輸入參數title

使用List表列串列movies中的電影名稱

```
    var body: some View {
        NavigationView {
            VStack {
                TextField("Enter title", text: $movieTitle)
                    .textFieldStyle(RoundedBorderTextFieldStyle())
                Button("Save"){

                    saveMovie(title: movieTitle)
                }
                List{
                    ForEach(movies, id: \.title){(movie) in
                        Text(movie.title)
                }
```

使用ForEach表列串列movies中的每個元素，指定title欄位為id

```
32                List{
33                    ForEach(movies, id: \.title){(movie) in
34                            Text(movie.title)
35                     }
36                    .onDelete(perform: {indexSet in
37                        indexSet.forEach{ index in
38                            movies.remove(at:index)
39
40                        }
41                    })
42            }
43            Spacer()
```

增加空白

增加list的刪除功能，將index所指的串列元素從串列中移除

```
NavigationView {
    VStack {
        TextField("Enter title", text: $movieTitle)
            .textFieldStyle(RoundedBorderTextFieldStyle())
        Button("Save"){

            saveMovie(title: movieTitle)
        }
        List{
            ForEach(movies, id: \.title){(movie) in
                    Text(movie.title)
            }
            .onDelete(perform: {indexSet in
                indexSet.forEach{ index in
                    movies.remove(at:index)

                }
            })
        } //List
        Spacer()
    }.padding()
    .navigationTitle("Movies")
} // navigation view
```

設定navigationTitle