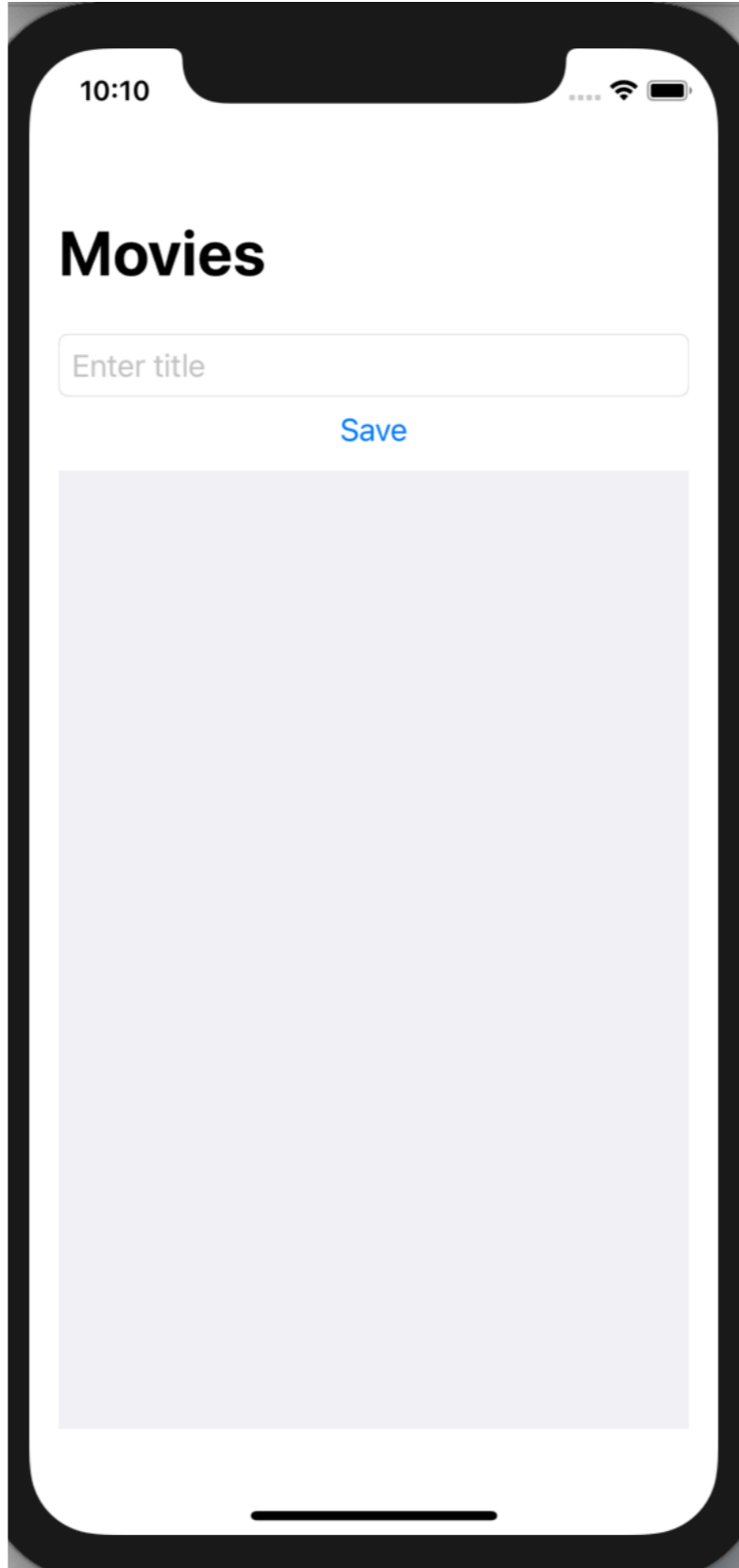


List 與 CoreData 資料建立與刪除



10:11



Movies

Save

SpiderMan

10:33

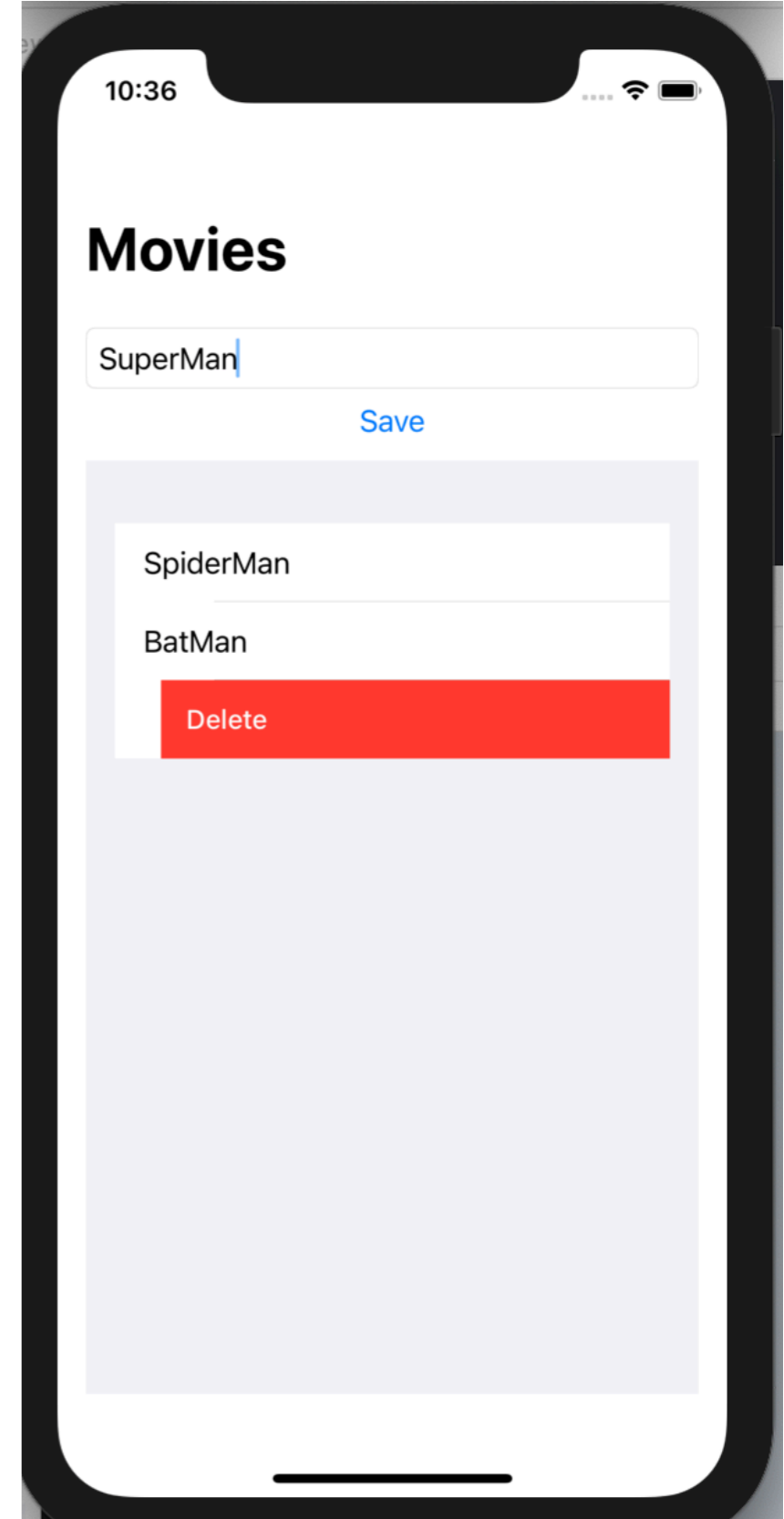
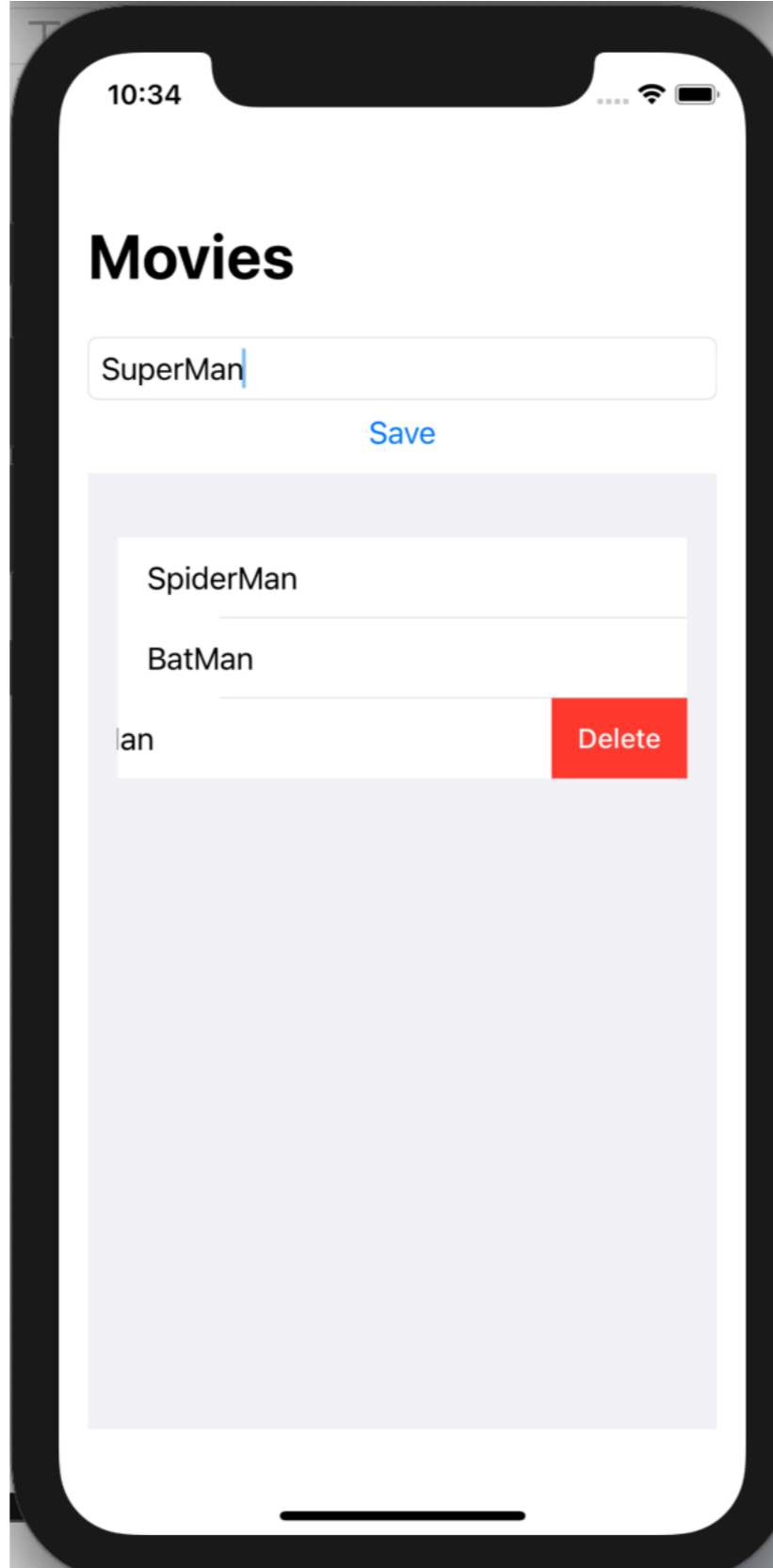
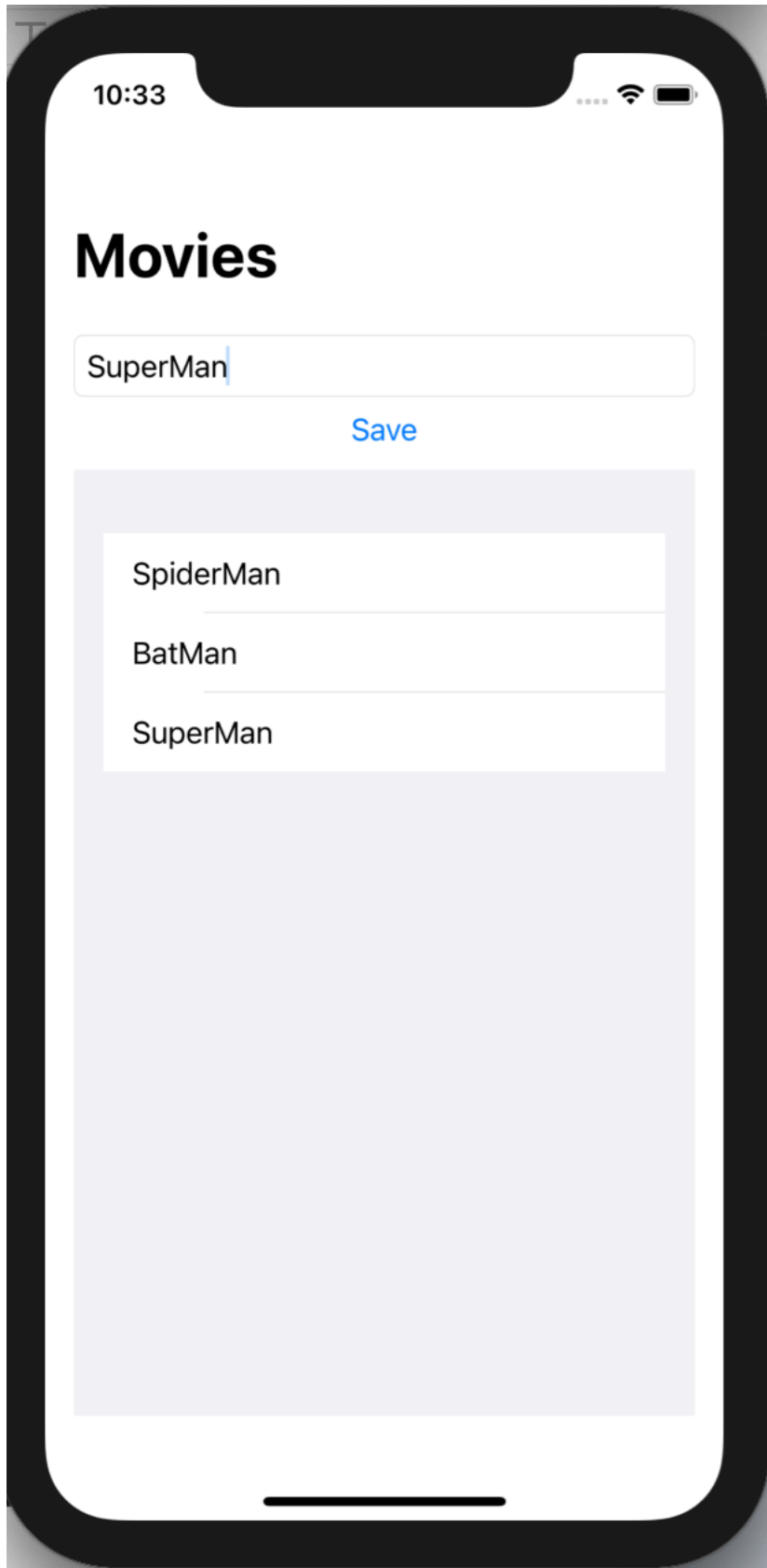


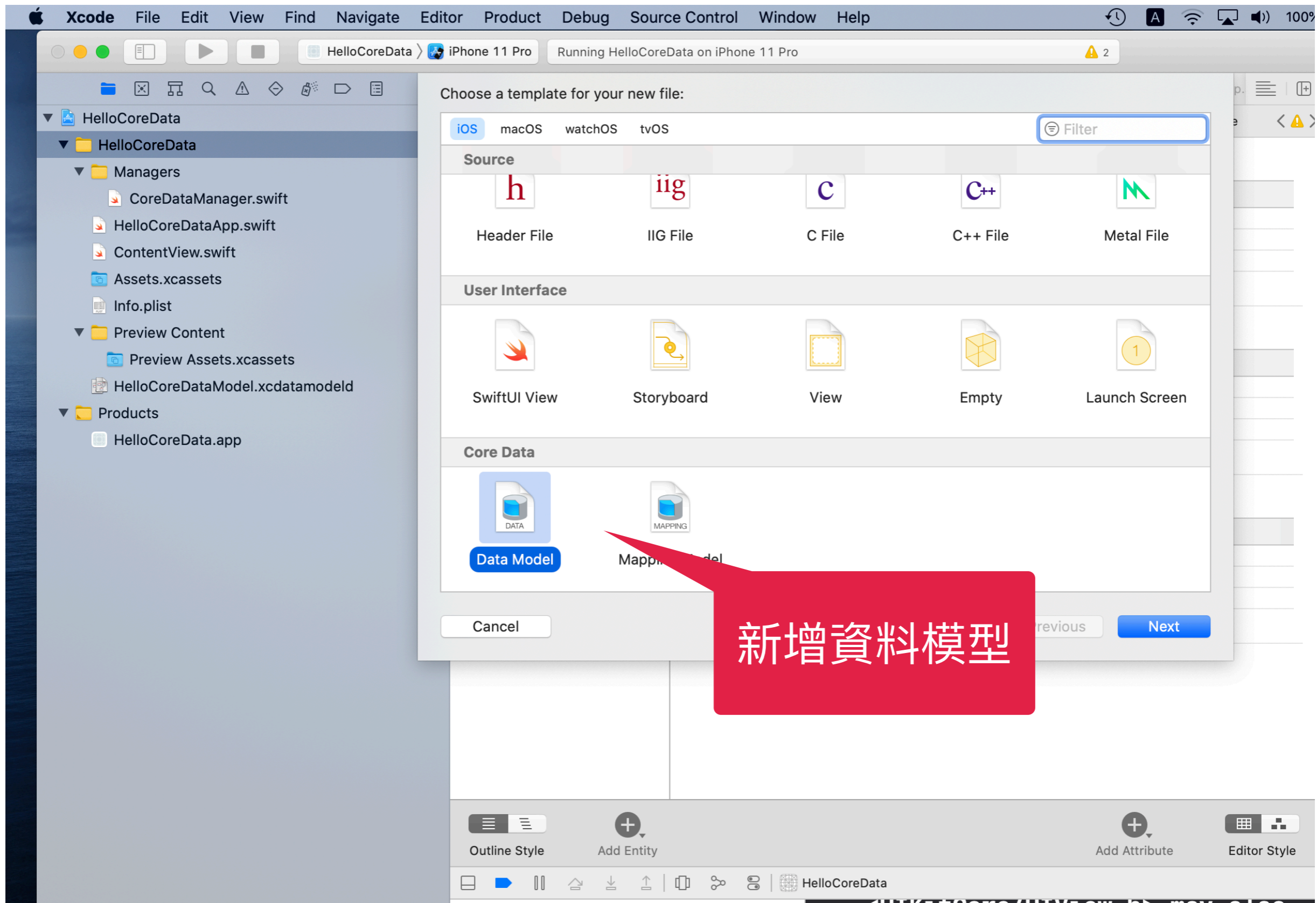
Movies

Save

SpiderMan

BatMan





資料實體

File Explorer (Left Panel):

- HelloCoreData
 - Managers
 - CoreDataManager.swift
 - HelloCoreDataApp.swift
 - ContentView.swift
 - Assets.xcassets
 - Info.plist
 - Preview Content
 - Preview Assets.xcassets
 - HelloCoreDataModel.xcdatamodeld** (Selected)
- Products
 - HelloCoreData.app

核資料模型

Entity Editor (Left Panel):

- ENTITIES
 - Movie** (Selected)
- FETCH REQUESTS
- CONFIGURATIONS
 - Default

Entity Editor (Right Panel):

- Attributes

Attribute	Type
title	String
- Relationships

Relationship	Destination	Invert
--------------	-------------	--------
- Fetches Properties

Fetches Property	Predicate
------------------	-----------

屬性欄位

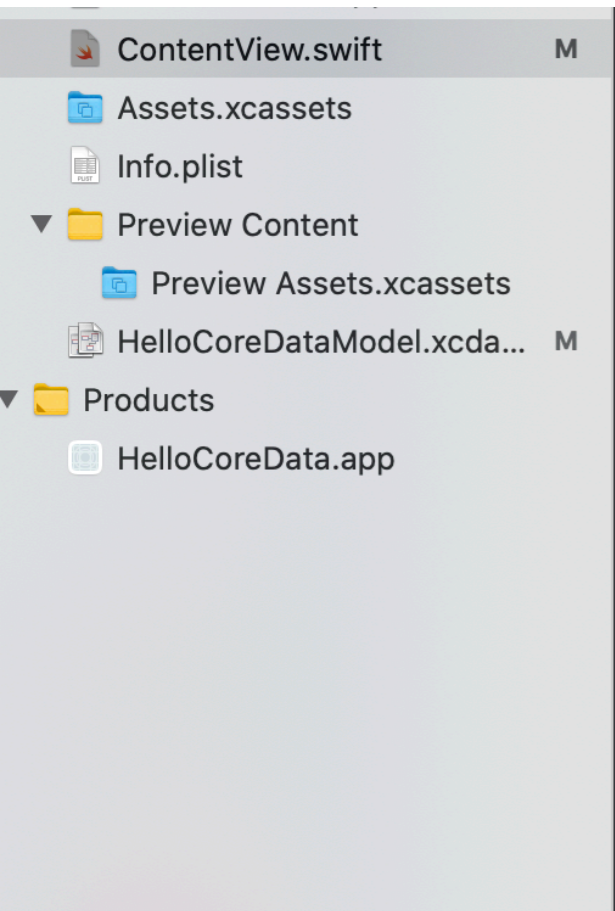
contentview

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

```
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]()
```

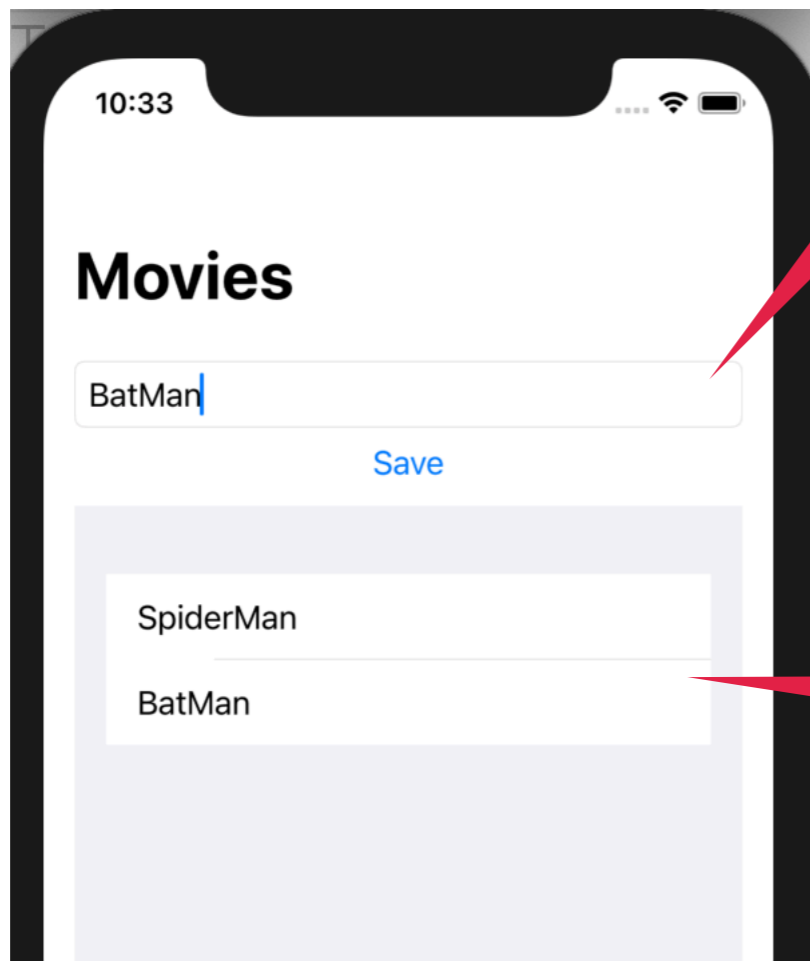


```
6 //
7
8 import SwiftUI
9
10 struct MyMovie{
11     let title: String
12 }
13
14 struct ContentView: View {
15
16     @State private var movieTitle: String = ""
17     @State private var movies: [MyMovie] = [MyMovie]()
```

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movies，包含的物件結構為
MyMovie，將連結到List的表列內容

導航view

使用Navigation View

```
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
```

使用垂直堆疊

輸入文字連結
movieTitle

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

儲存按鍵

```
        }  
        List{
```

```
    }  
}
```

```
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