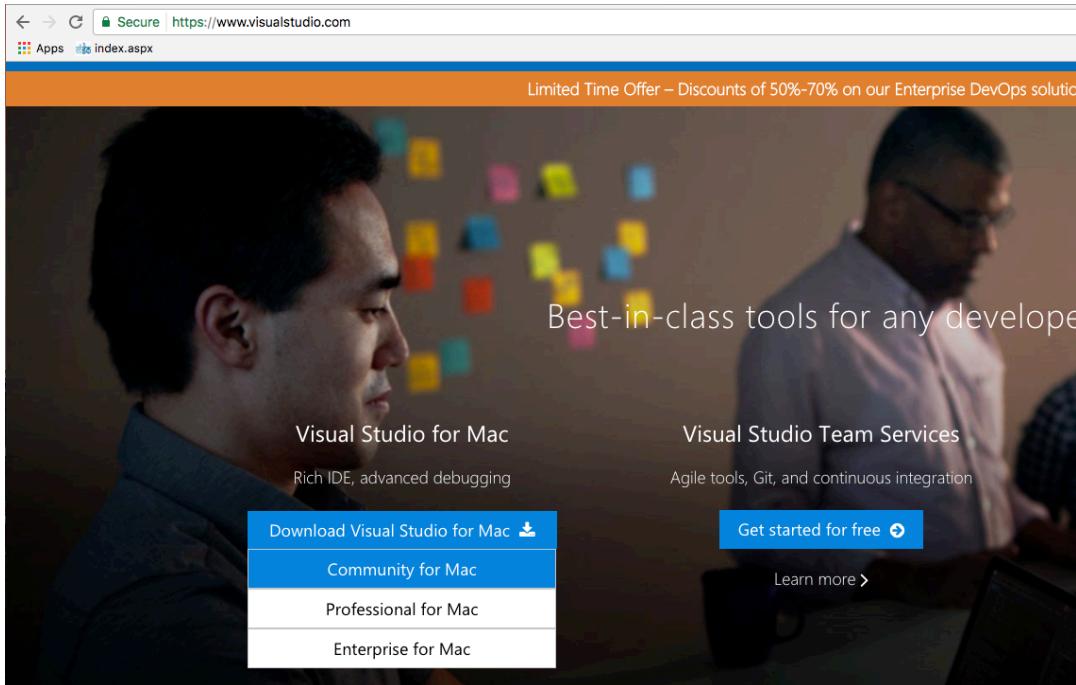


Lilitab SDK

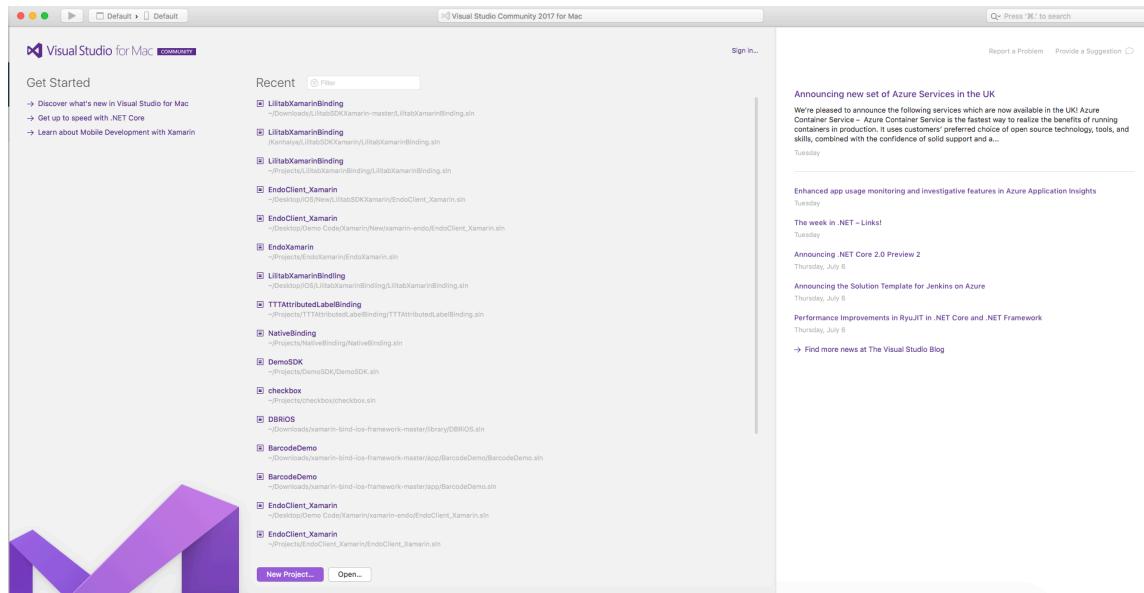
1. Download and install **Visual Studio Community Edition** for Mac.(Download Url: <https://www.visualstudio.com/>)



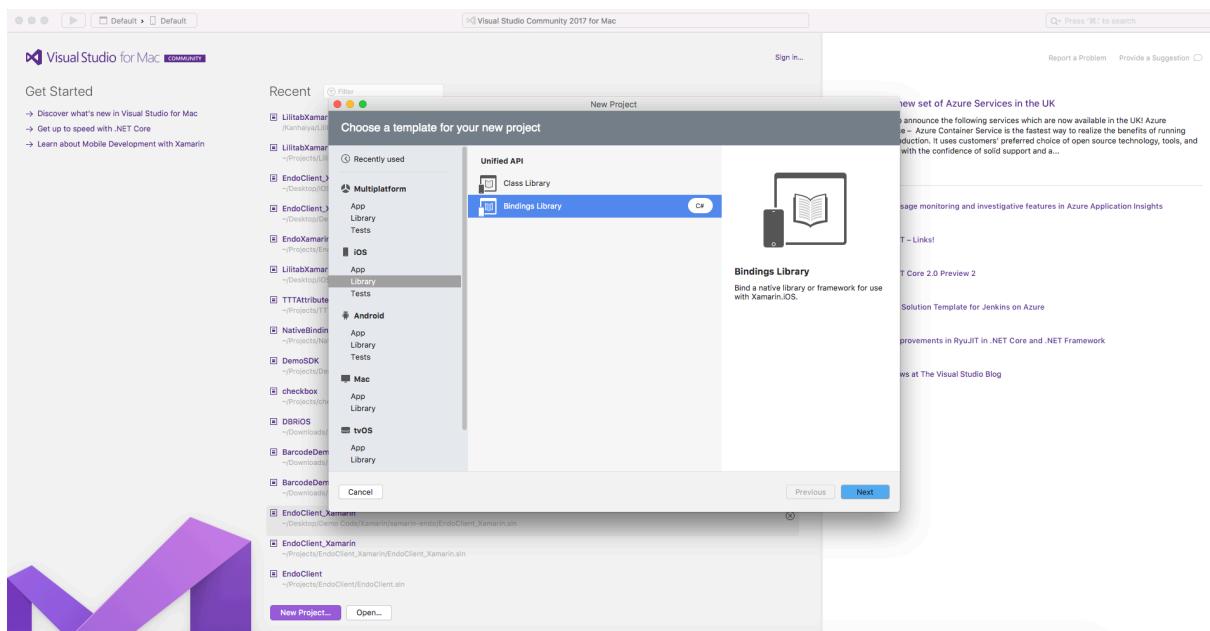
2. Download and install objective Sharpie tool.(<https://developer.xamarin.com/guides/cross-platform/macios/binding/objective-sharpie/>)

A screenshot of the Xamarin developer site. The top navigation bar includes links for Products, Customers, Pricing, Developers (which is the active tab), Support, Resources, and Sign In. The main content area shows a breadcrumb trail: Cross-Platform > iOS and Mac > Binding Objective-C > Objective Sharpie. The page title is 'Objective Sharpie' and the subtitle is 'Automated Binding Definitions using Objective Sharpie'. Below the title, there's a brief description of what Objective Sharpie is. To the right, there's a sidebar with a red box highlighting a 'Sample Code' link: 'Objective Sharpie download'. A red arrow points from the text 'Click for download Sharpie' to this link. Other sidebar links include 'PDF for offline use' and 'Related Articles'.

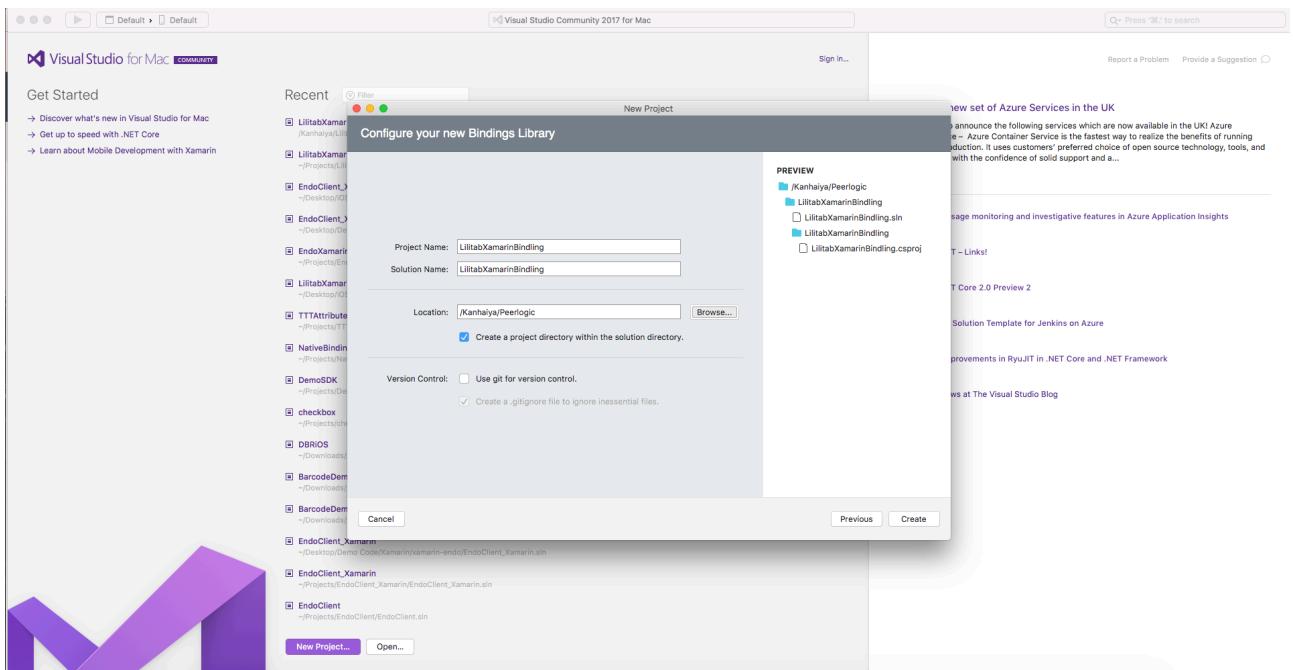
3. Open Visual Studio Application and click on New Project.



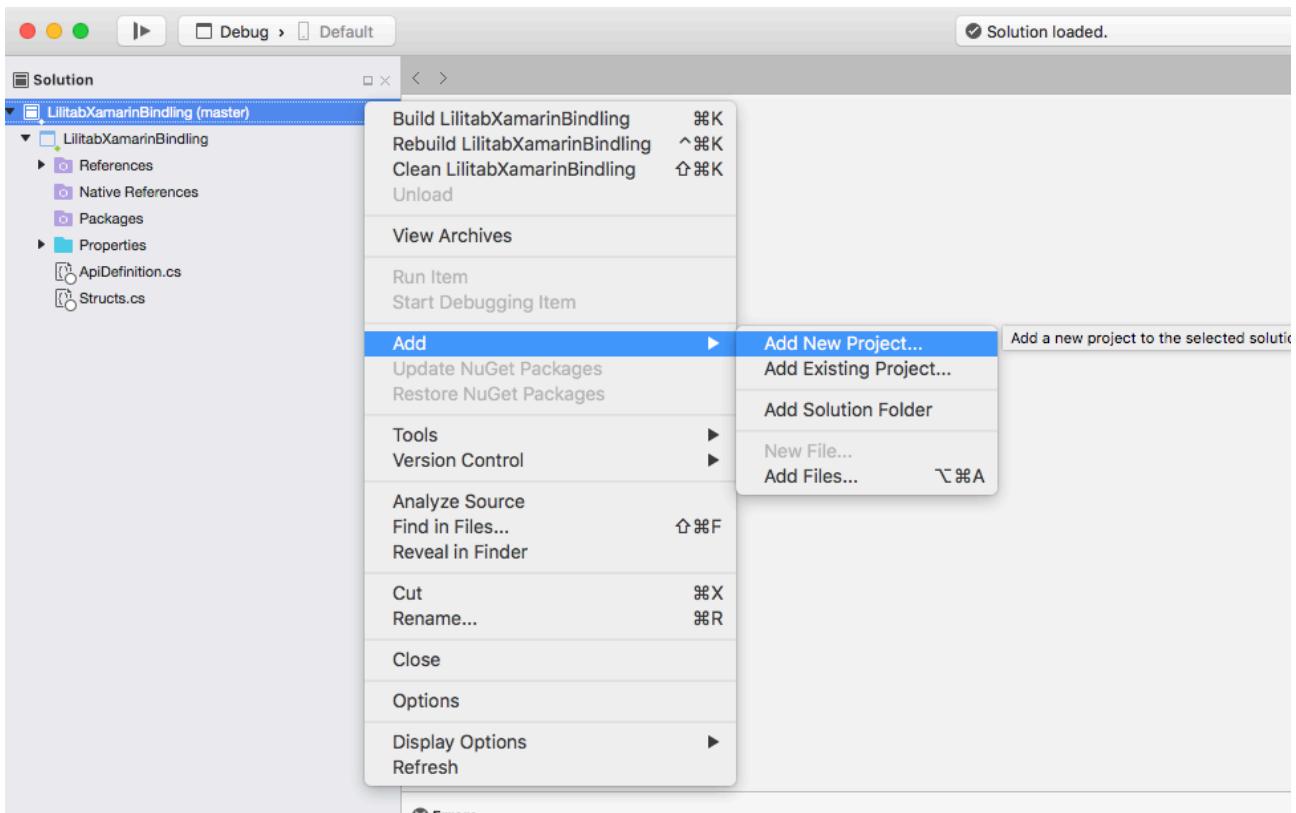
4. Choose iOS-> Library-> Bindings Library -> Next



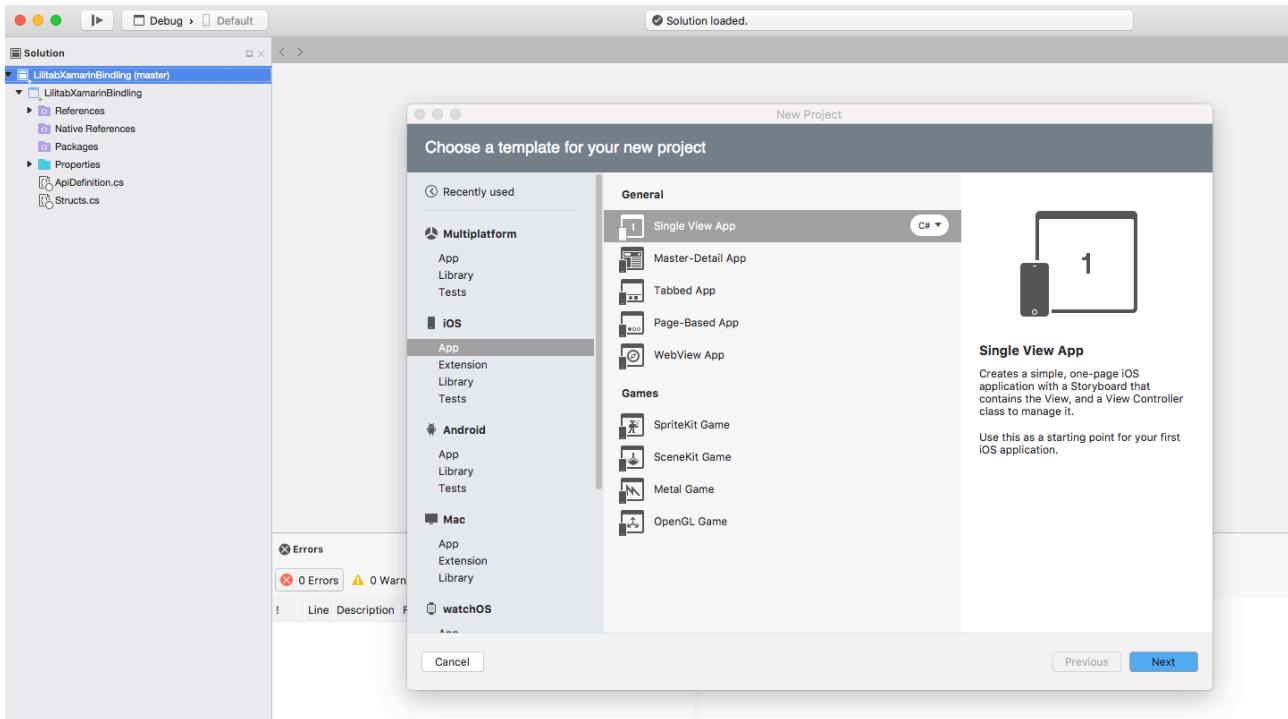
5. Type Project Name (LilitabXamarinBinding) and click on Next Button.



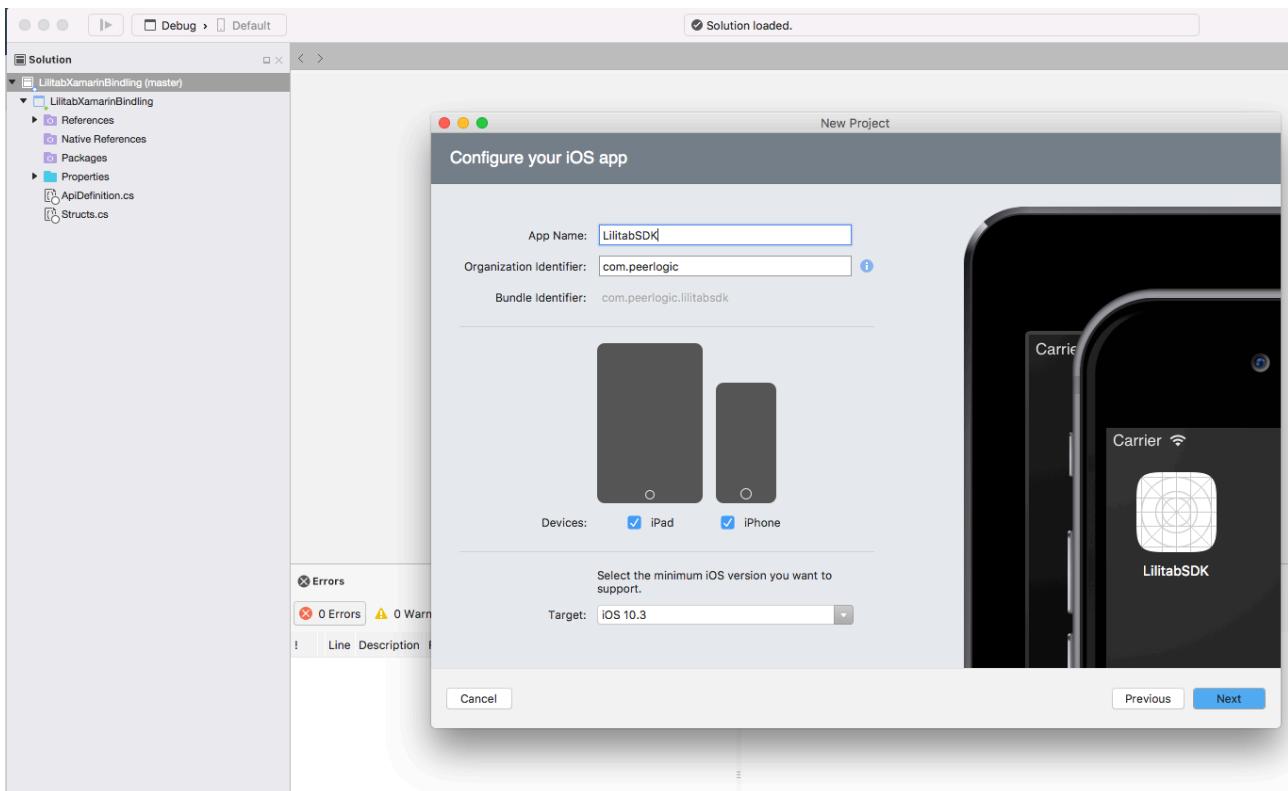
6. Right click on LilitabXamarinBinding(master) -> Add -> Add New Project



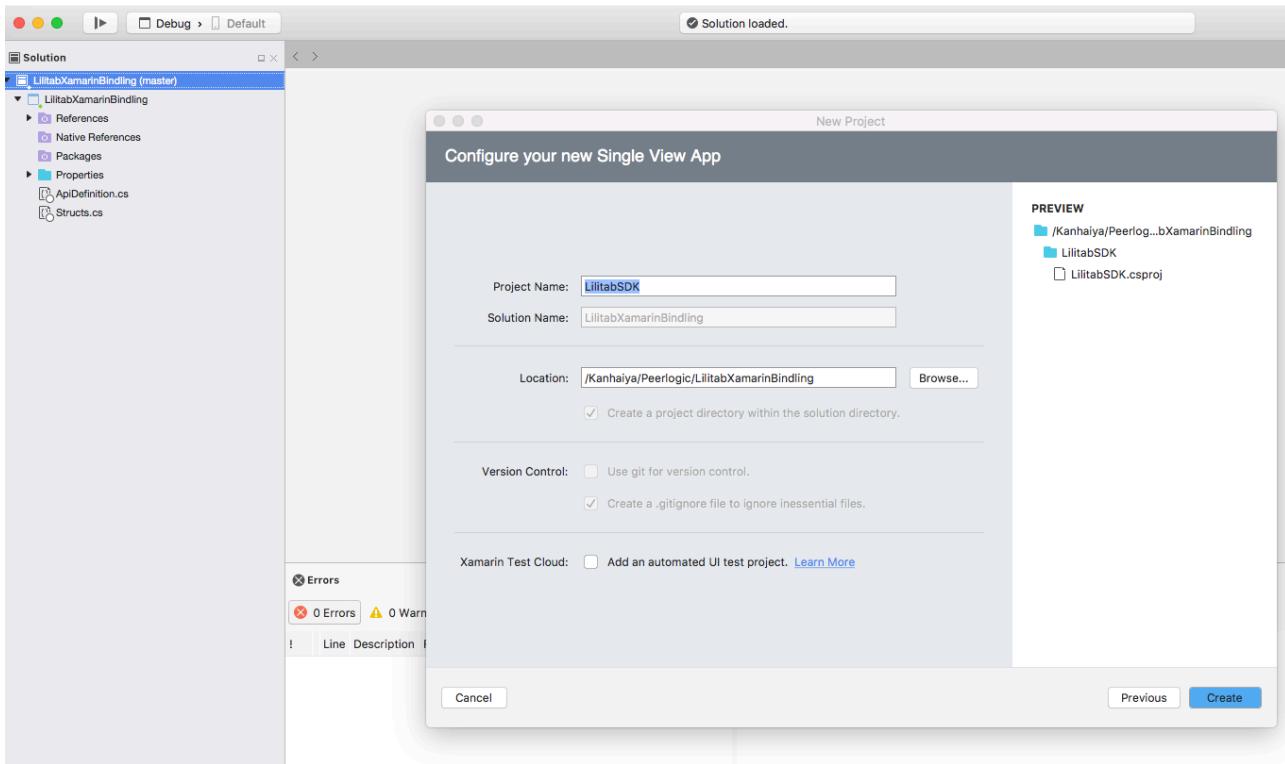
7. Choose iOS-> App->Single View App -> Next



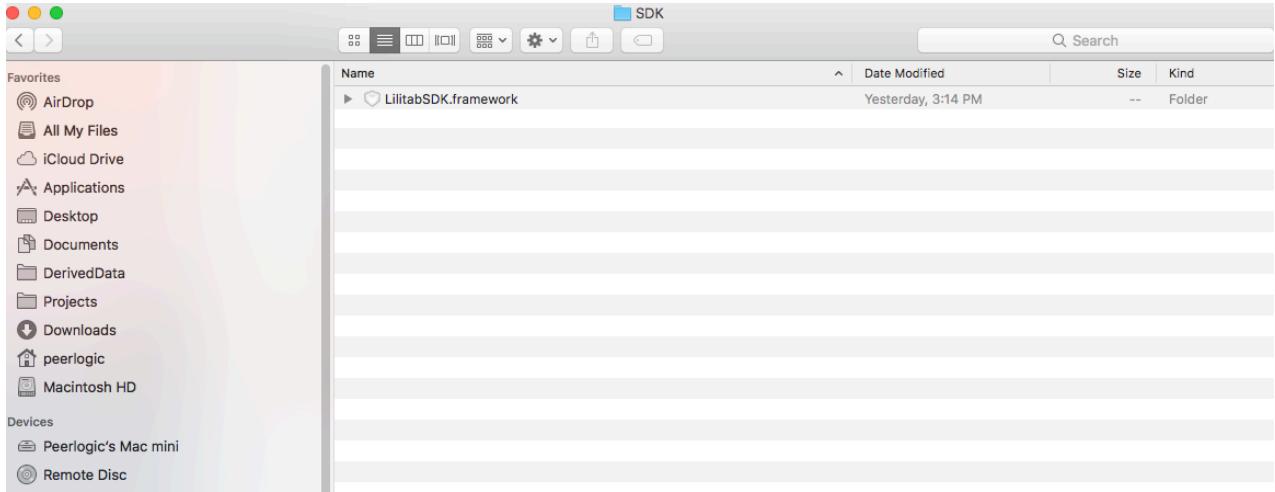
8. Type app name (LilitabSDK) and Click on Next



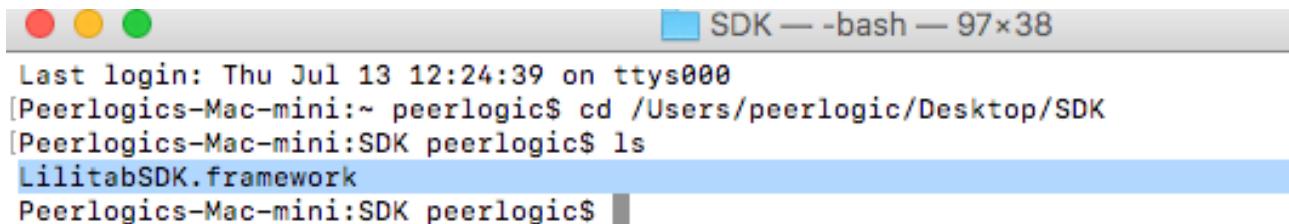
9.Click on Create button



10.We need a Universal LilitabSDK.Framework.



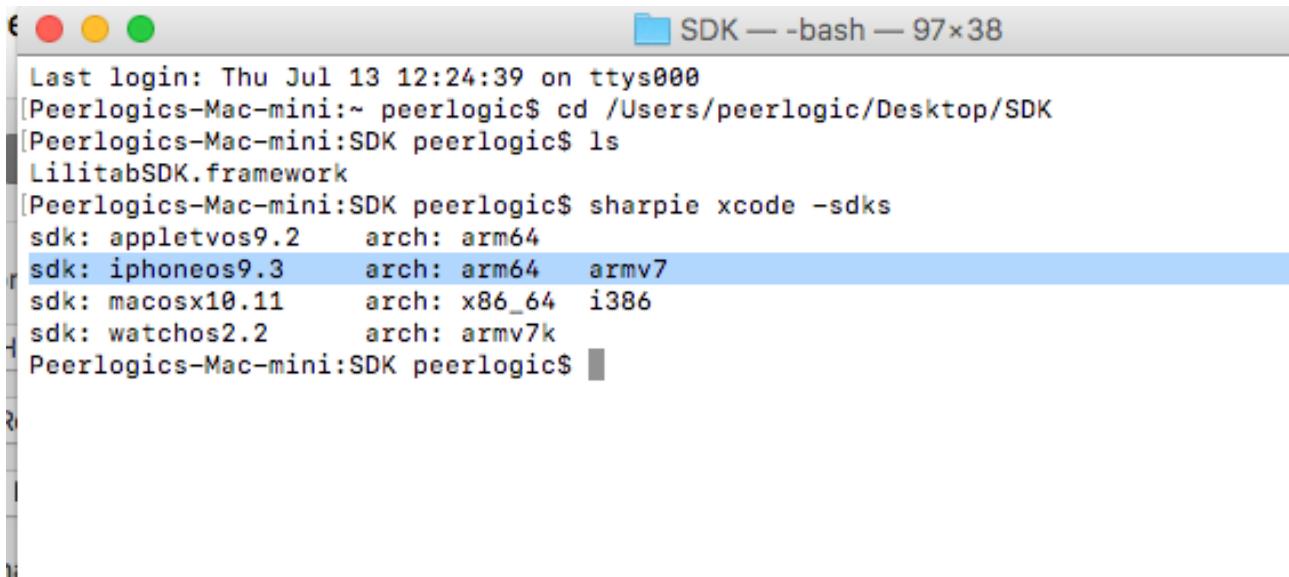
11. Open the terminal and change directory to the LilitabSDK.Framework



```
Last login: Thu Jul 13 12:24:39 on ttys000
[Peerlogics-Mac-mini:~ peerlogic$ cd /Users/peerlogic/Desktop/SDK
[Peerlogics-Mac-mini:SDK peerlogic$ ls
LilitabSDK.framework
Peerlogics-Mac-mini:SDK peerlogic$ ]
```

12. Type below command:

sharpie xcode -sdks



```
Last login: Thu Jul 13 12:24:39 on ttys000
[Peerlogics-Mac-mini:~ peerlogic$ cd /Users/peerlogic/Desktop/SDK
[Peerlogics-Mac-mini:SDK peerlogic$ ls
LilitabSDK.framework
[Peerlogics-Mac-mini:SDK peerlogic$ sharpie xcode -sdks
sdk: appletvos9.2      arch: arm64
sdk: iphoneos9.3       arch: arm64    armv7
sdk: macosx10.11        arch: x86_64   i386
sdk: watchos2.2         arch: armv7k
Peerlogics-Mac-mini:SDK peerlogic$ ]
```

Note: This machine supports iphoneos9.3 SDK; so the same version is used in the next command.

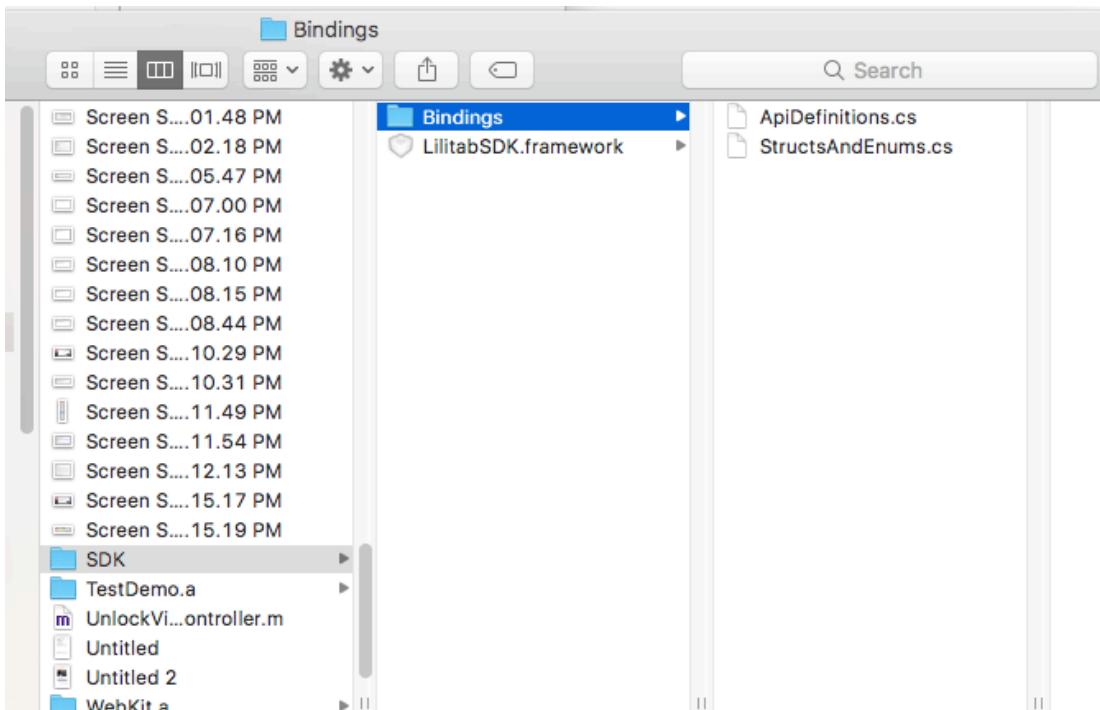
13.create objective.binding

```
sharpie bind -sdk iphoneos9.3 -output Bindings LilitabSDK.framework/Headers/  
LilitabSDK.h -c -F. -v
```

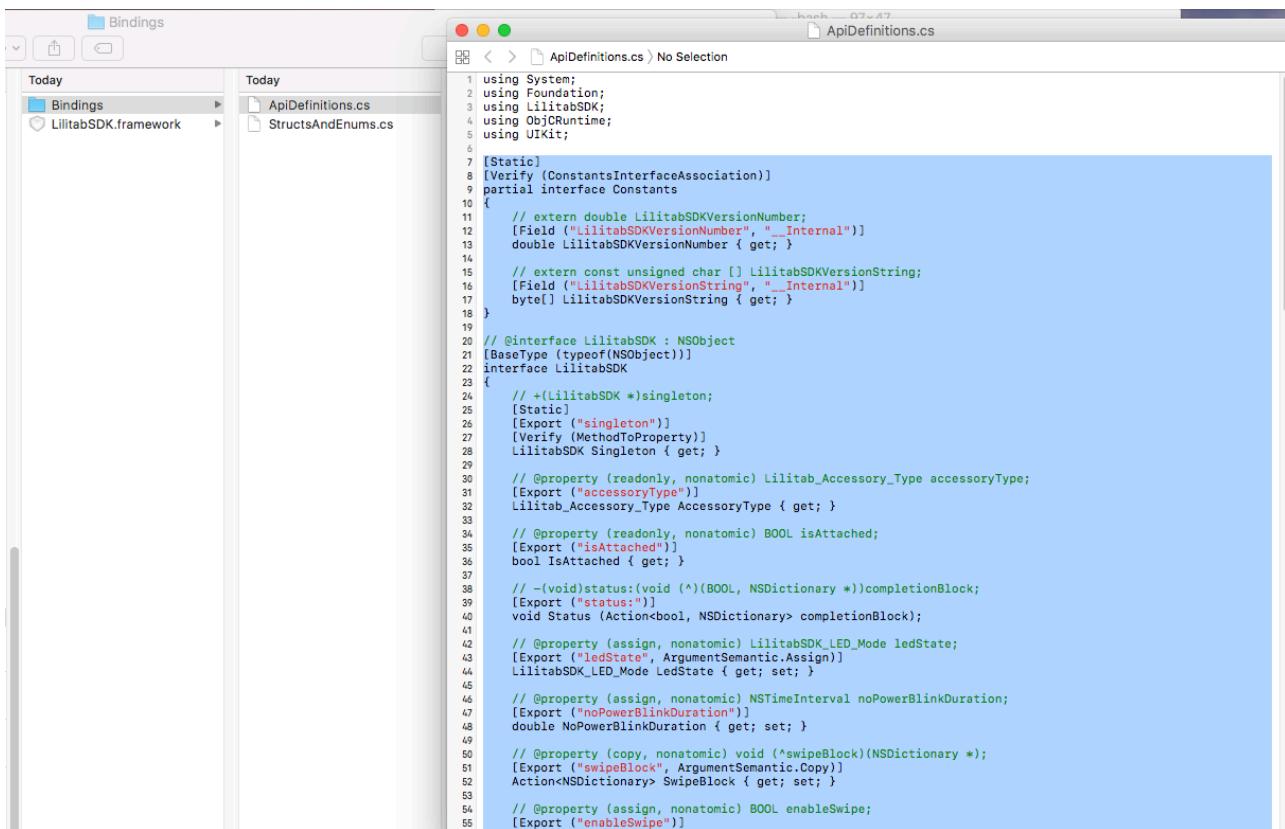
```
Last login: Thu Jul 13 12:24:39 on ttys000
[Peerlogics-Mac-mini:~ peerlogic$ cd /Users/peerlogic/Desktop/SDK
[Peerlogics-Mac-mini:SDK peerlogic$ ls
LilitabSDK.framework
[Peerlogics-Mac-mini:SDK peerlogic$ sharpie xcode -sdks
sdk: appletvos9.2      arch: arm64
sdk: iphoneos9.3       arch: arm64    armv7
sdk: macosx10.11       arch: x86_64   i386
sdk: watchos2.2        arch: armv7k
[Peerlogics-Mac-mini:SDK peerlogic$ sharpie bind -sdk iphoneos9.3 -output Bindings LilitabSDK.framework/Headers/LilitabSDK.h -v
Parsing 1 header files...
clang version 4.0.0
Target: aarch64-apple-darwin16.1.0
Thread model: posix
InstalledDir:
clang version 4.0.0
Target: aarch64-apple-darwin16.1.0
Thread model: posix
InstalledDir:
clang Invocation:
"clang-tool" "--cc1" "-triple" "arm64-apple-ios9.3.0" "-Wdeprecated-objc-is-a-usage" "-Werror=deprecated-objc-is-a-usage" "-Werror=implicit-function-declaration" "-fsyntax-only" "-disable-free" "-disable-llvm-verifier" "-discard-value-names" "-main-file-name" "LilitabSDK.h" "-mrelocation-model" "pic" "-pic-level" "2" "-mthread-model" "posix" "-mdisable-fp-elim" "-masm-verbose" "-target-cpu" "cyclone" "-target-feature" "+neon" "-target-feature" "+crypto" "-target-feature" "+zcm" "-target-feature" "+zzc" "-target-abi" "darwinppcs" "-target-linker-version" "253.3" "-v" "-dwarf-column-info" "-debugger-tuning=lldb" "-resource-dir" "/Library/Frameworks/ObjectiveSharpie.framework/Versions/Current/clang-resources" "-isysroot" "/Applications/Xcode 2.app/Contents/Developer/Platforms/iPhoneOS.platform/DeveloperSDKs/iPhoneOS 3.sdk" "-f" "-D" "NFC_BUILID_22_LIKE_44" "-I" "/Ap
```

Note: `iphoneos9.3` -> “Your installed `iphoneos9.3` SDK”

14. Automatically a folder named ‘Binding’ will be created with two files in your LilitabSDK.framework path.



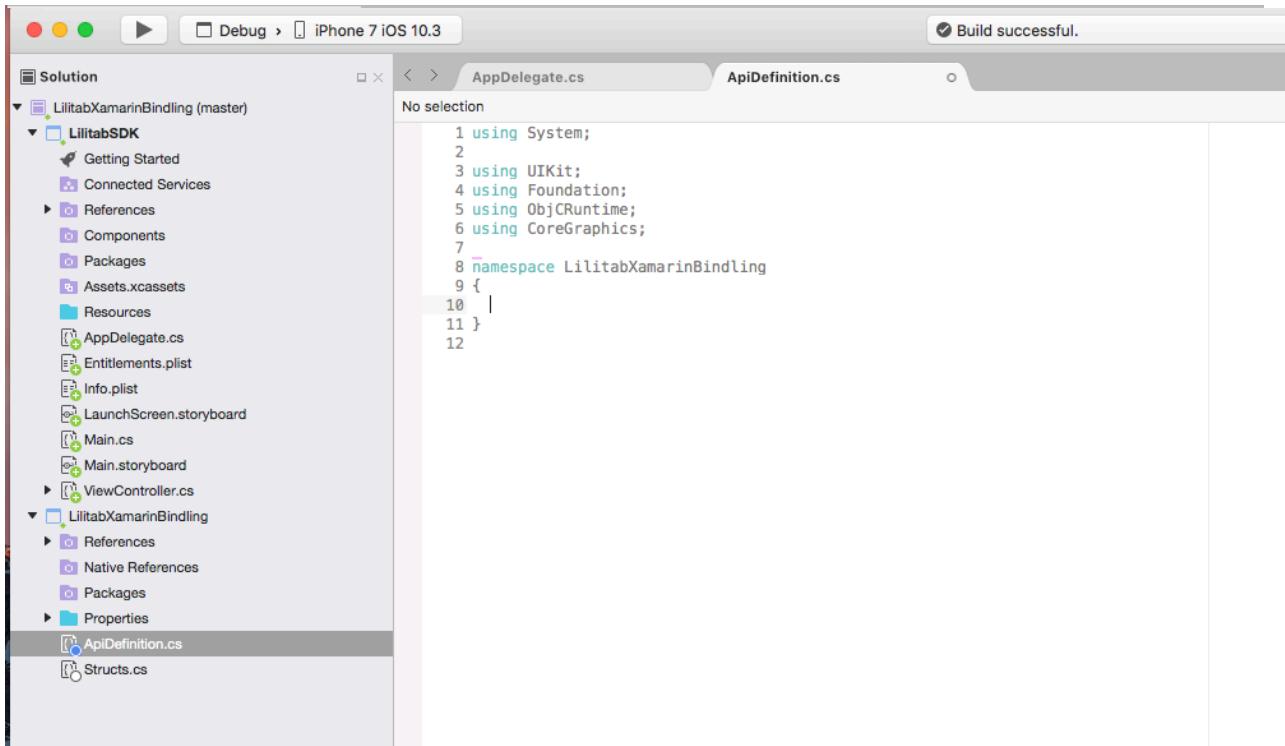
15. Open your created ApiDefinitions.cs in the **Binding folder** and copy all contents (Except imported or using files)



The screenshot shows the Xcode interface with the 'Bindings' folder selected in the left sidebar. Inside the 'Bindings' folder, there are two files: 'ApiDefinitions.cs' and 'StructsAndEnums.cs'. The 'ApiDefinitions.cs' file is open in the main editor window, displaying C# code. The code includes several 'using' statements at the top, followed by a partial interface named 'Constants'. It contains declarations for external variables like 'LilitabSDKVersionNumber' and 'LilitabSDKVersionString', and an interface named 'LilitabSDK'. The code uses attributes such as [Static], [Verify], [Field], [Export], and [MethodToProperty] to map Objective-C methods to C# properties and methods.

```
1 using System;
2 using Foundation;
3 using LilitabSDK;
4 using ObjCRuntime;
5 using UIKit;
6
7 [Static]
8 [Verify(ConstantsInterfaceAssociation)]
9 partial interface Constants
10 {
11     // extern double LilitabSDKVersionNumber;
12     [Field("LilitabSDKVersionNumber", "__Internal")]
13     double LilitabSDKVersionNumber { get; }
14
15     // extern const unsigned char [] LilitabSDKVersionString;
16     [Field("LilitabSDKVersionString", "__Internal")]
17     byte[] LilitabSDKVersionString { get; }
18 }
19
20 // @interface LilitabSDK : NSObject
21 [BaseType(typeof(NSObject))]
22 interface LilitabSDK
23 {
24     // +(LilitabSDK *)singleton;
25     [Static]
26     [Export("singleton")]
27     [Verify(MethodToProperty)]
28     LilitabSDK Singleton { get; }
29
30     // @property (readonly, nonatomic) Lilitab_Accessory_Type accessoryType;
31     [Export("accessoryType")]
32     Lilitab_Accessory_Type AccessoryType { get; }
33
34     // @property (readonly, nonatomic) BOOL isAttached;
35     [Export("isAttached")]
36     bool IsAttached { get; }
37
38     // -(void)status:(void (^)(BOOL, NSDictionary *))completionBlock;
39     [Export("status:")]
40     void Status (Action<bool, NSDictionary> completionBlock);
41
42     // @property (assign, nonatomic) LilitabSDK_LED_Mode ledState;
43     [Export("ledState", ArgumentSemantic.Assign)]
44     LilitabSDK_LED_Mode LedState { get; set; }
45
46     // @property (assign, nonatomic) NSTimeInterval noPowerBlinkDuration;
47     [Export("noPowerBlinkDuration")]
48     double NoPowerBlinkDuration { get; set; }
49
50     // @property (copy, nonatomic) void (^swipeBlock)(NSDictionary *);
51     [Export("swipeBlock", ArgumentSemantic.Copy)]
52     Action<NSDictionary> SwipeBlock { get; set; }
53
54     // @property (assign, nonatomic) BOOL enableSwipe;
55     [Export("enableSwipe")]
56 }
```

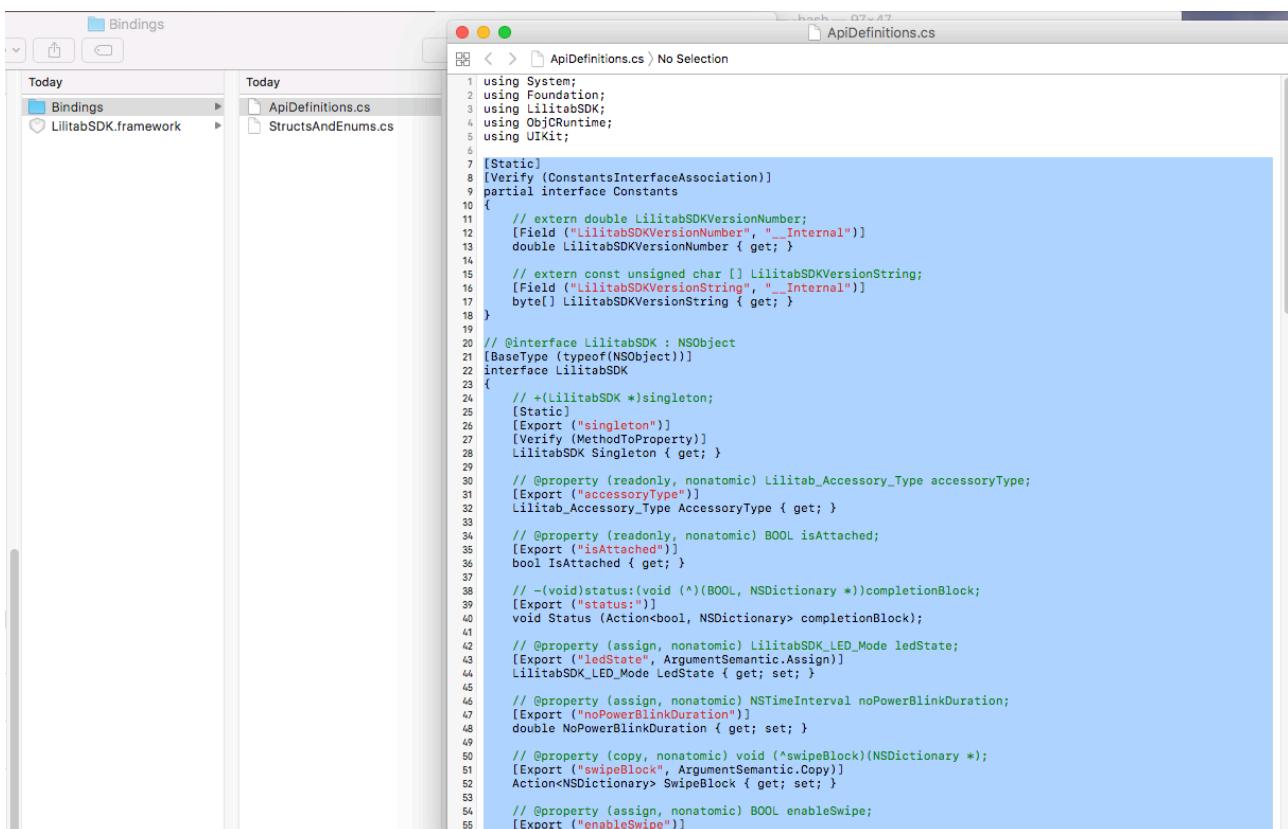
16 Open Visual Studio and Open ApiDification.cs and Remove all Commented code.(Excepts imported or using files)



The screenshot shows the Visual Studio interface with the 'Solution Explorer' on the left. The 'LilitabXamarinBinding' project is selected, showing its structure. In the 'AppDelegate.cs' file, the build status is shown as 'Build successful'. The 'ApiDefinition.cs' file is open in the main editor window. The code in 'ApiDefinition.cs' consists of a single line: '1 using System;'. This indicates that the commented-out code from the Xcode screenshot has been removed.

```
1 using System;
```

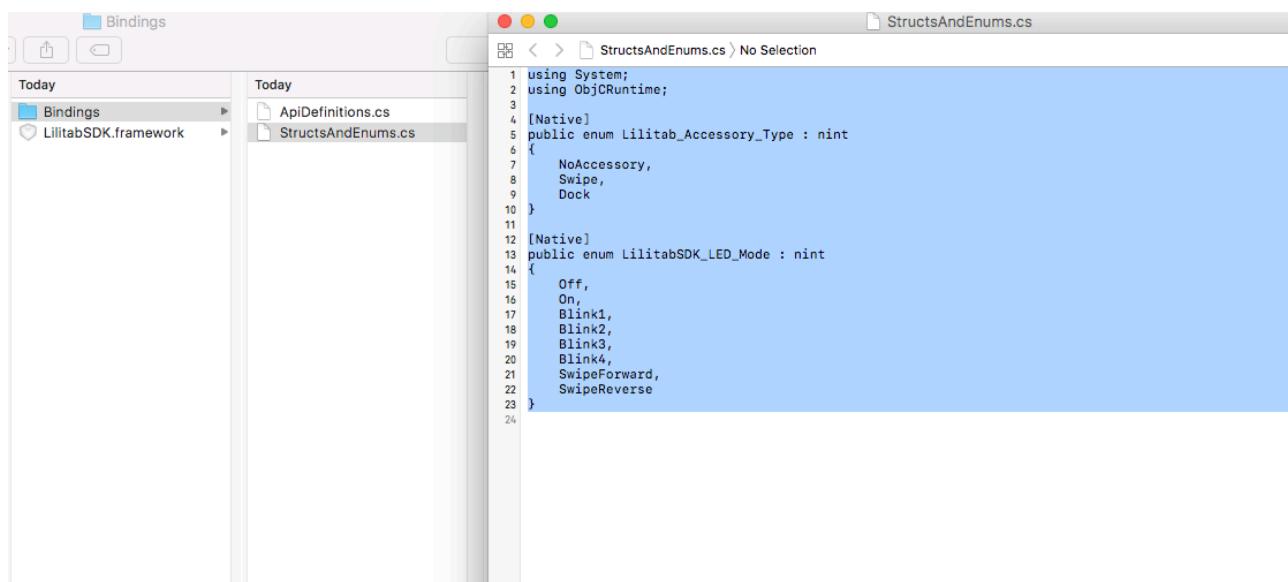
17. Paste all copied contents in the **LilitabXamarinBinding** method of Aplication.cs file



The screenshot shows the Xcode interface with the project navigation bar at the top. Below it, the 'Bindings' folder is selected in the left sidebar. In the main workspace, the 'ApiDefinitions.cs' file is open. The code in the file is a C# class definition for the LilitabSDK interface, containing various properties and methods. The code is highlighted in blue and green, indicating syntax and annotations.

```
1 using System;
2 using Foundation;
3 using LilitabSDK;
4 using ObjCRuntime;
5 using UIKit;
6
7 [Static]
8 [Verify (ConstantsInterfaceAssociation)]
9 partial interface Constants
10 {
11     // extern double LilitabSDKVersionNumber;
12     [Field ("LilitabSDKVersionNumber", "__Internal")]
13     double LilitabSDKVersionNumber { get; }
14
15     // extern const unsigned char [] LilitabSDKVersionString;
16     [Field ("LilitabSDKVersionString", "__Internal")]
17     byte[] LilitabSDKVersionString { get; }
18 }
19
20 // @interface LilitabSDK : NSObject
21 [BaseType (typeof(NSObject))]
22 interface LilitabSDK
23 {
24     // +(LilitabSDK *)singleton;
25     [Static]
26     [Export ("singleton")]
27     [Verify (MethodToProperty)]
28     LilitabSDK Singleton { get; }
29
30     // @property (readonly, nonatomic) Lilitab_Accessory_Type accessoryType;
31     [Export ("accessoryType")]
32     Lilitab_Accessory_Type AccessoryType { get; }
33
34     // @property (readonly, nonatomic) BOOL isAttached;
35     [Export ("isAttached")]
36     bool IsAttached { get; }
37
38     // -(void)status:(void (^)(BOOL, NSDictionary *))completionBlock;
39     [Export ("status:")]
40     void Status (Action<bool, NSDictionary> completionBlock);
41
42     // @property (assign, nonatomic) LilitabSDK_LED_Mode ledState;
43     [Export ("ledState", ArgumentSemantic.Assign)]
44     LilitabSDK_LED_Mode LedState { get; set; }
45
46     // @property (assign, nonatomic) NSTimeInterval noPowerBlinkDuration;
47     [Export ("noPowerBlinkDuration")]
48     double NoPowerBlinkDuration { get; set; }
49
50     // @property (copy, nonatomic) void (^swipeBlock)(NSDictionary *);
51     [Export ("swipeBlock", ArgumentSemantic.Copy)]
52     Action<NSDictionary> SwipeBlock { get; set; }
53
54     // @property (assign, nonatomic) BOOL enableSwipe;
55     [Export ("enableSwipe")]
56 }
```

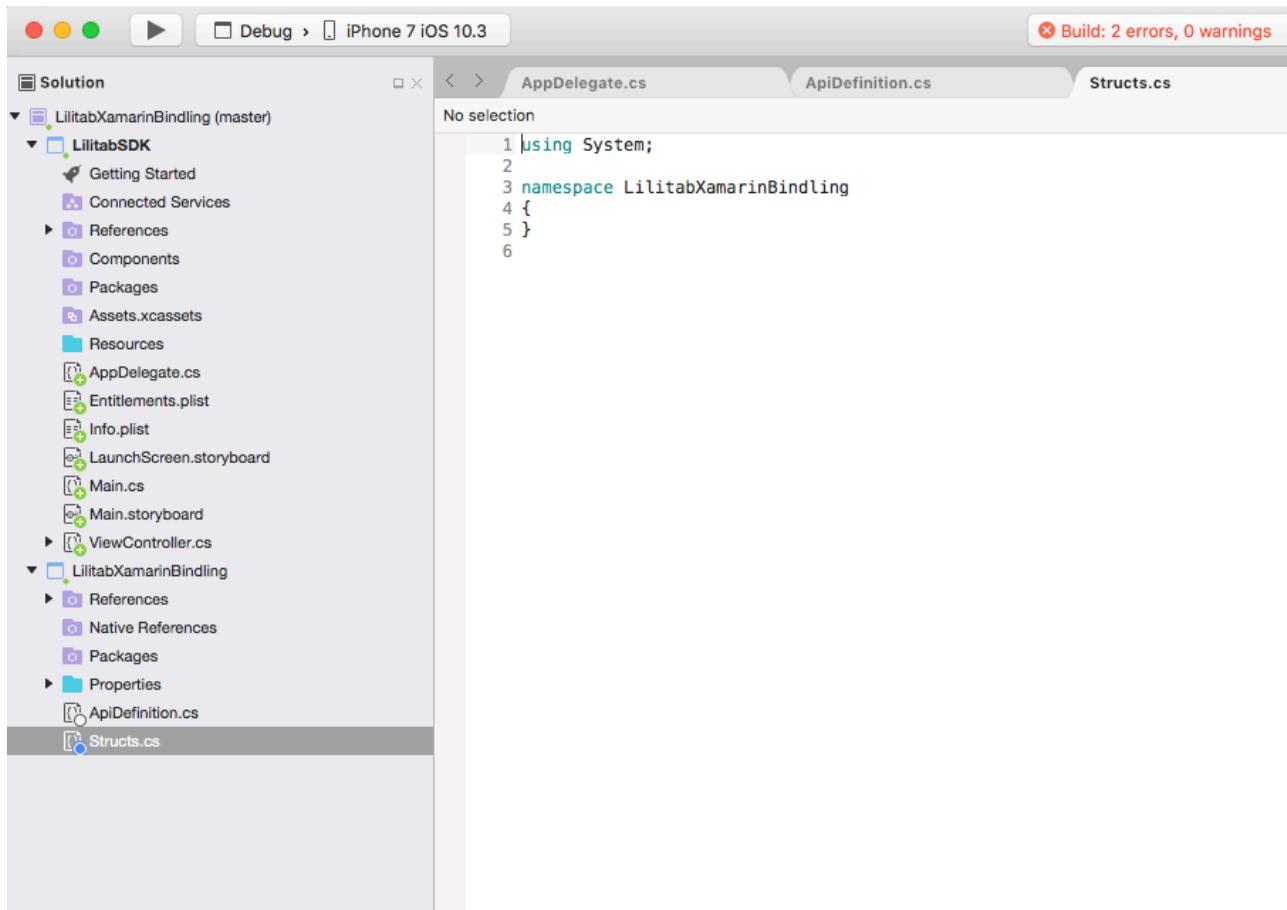
18. Now open your created **StructsAndEnums.cs** in the **Binding** folder. And copy all contents



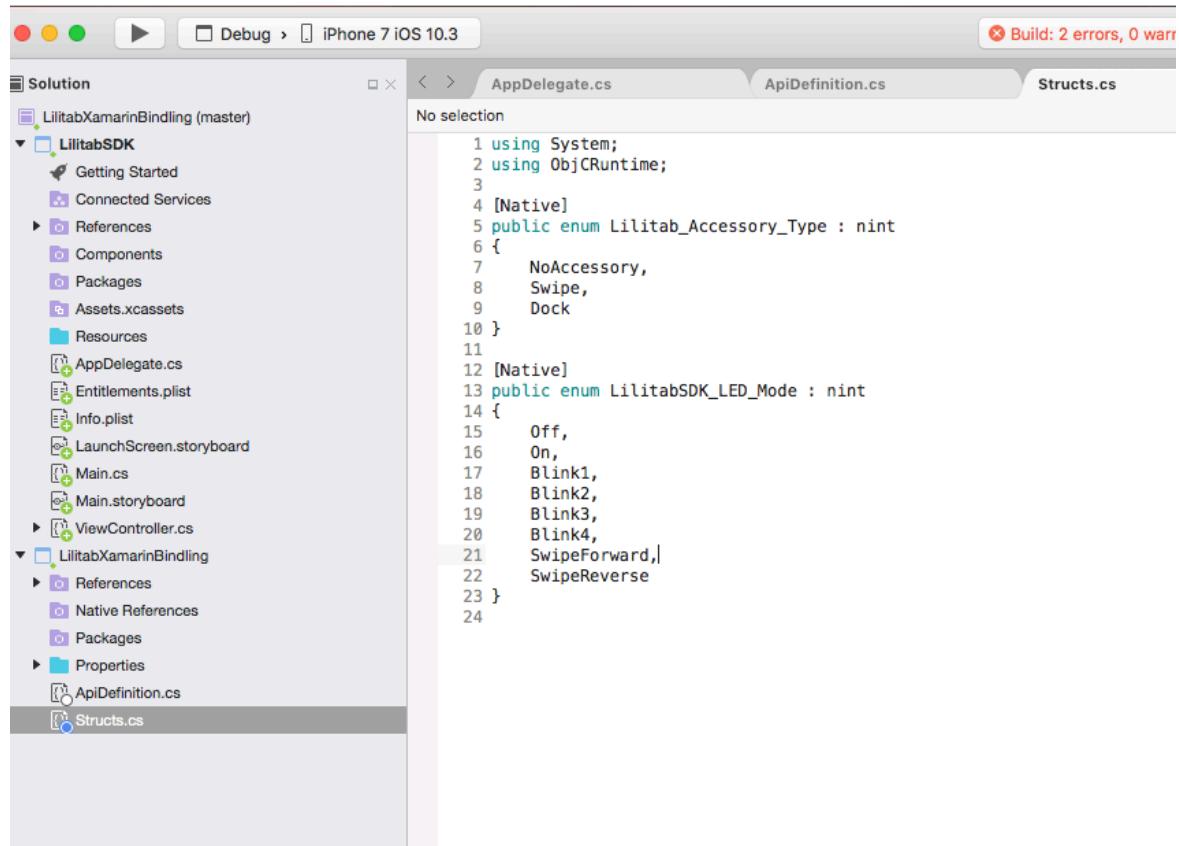
The screenshot shows the Xcode interface with the project navigation bar at the top. Below it, the 'Bindings' folder is selected in the left sidebar. In the main workspace, the 'StructsAndEnums.cs' file is open. The code defines two enum classes: Lilitab_Accessory_Type and LilitabSDK_LED_Mode, both using the [Native] attribute. The Lilitab_Accessory_Type enum has values NoAccessory, Swipe, and Dock. The LilitabSDK_LED_Mode enum has values Off, On, Blink1, Blink2, Blink3, Blink4, SwipeForward, and SwipeReverse.

```
1 using System;
2 using ObjCRuntime;
3
4 [Native]
5 public enum Lilitab_Accessory_Type : nint
6 {
7     NoAccessory,
8     Swipe,
9     Dock
10 }
11
12 [Native]
13 public enum LilitabSDK_LED_Mode : nint
14 {
15     Off,
16     On,
17     Blink1,
18     Blink2,
19     Blink3,
20     Blink4,
21     SwipeForward,
22     SwipeReverse
23 }
```

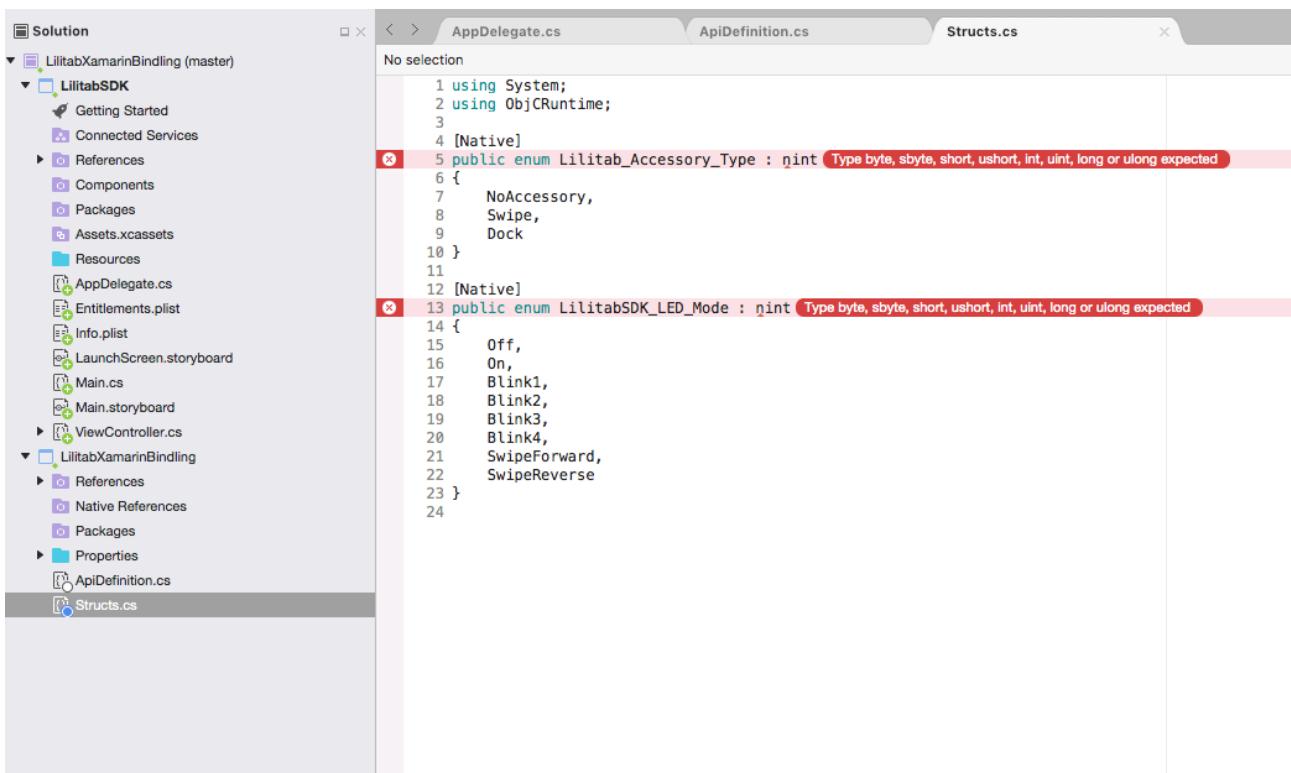
19 Open Visual Studio and Open **Structs.cs**



18. Paste all copied contents in the **LilitabXamarinBinding** method of **Structs.cs** file



19. Build the application with the simulator.



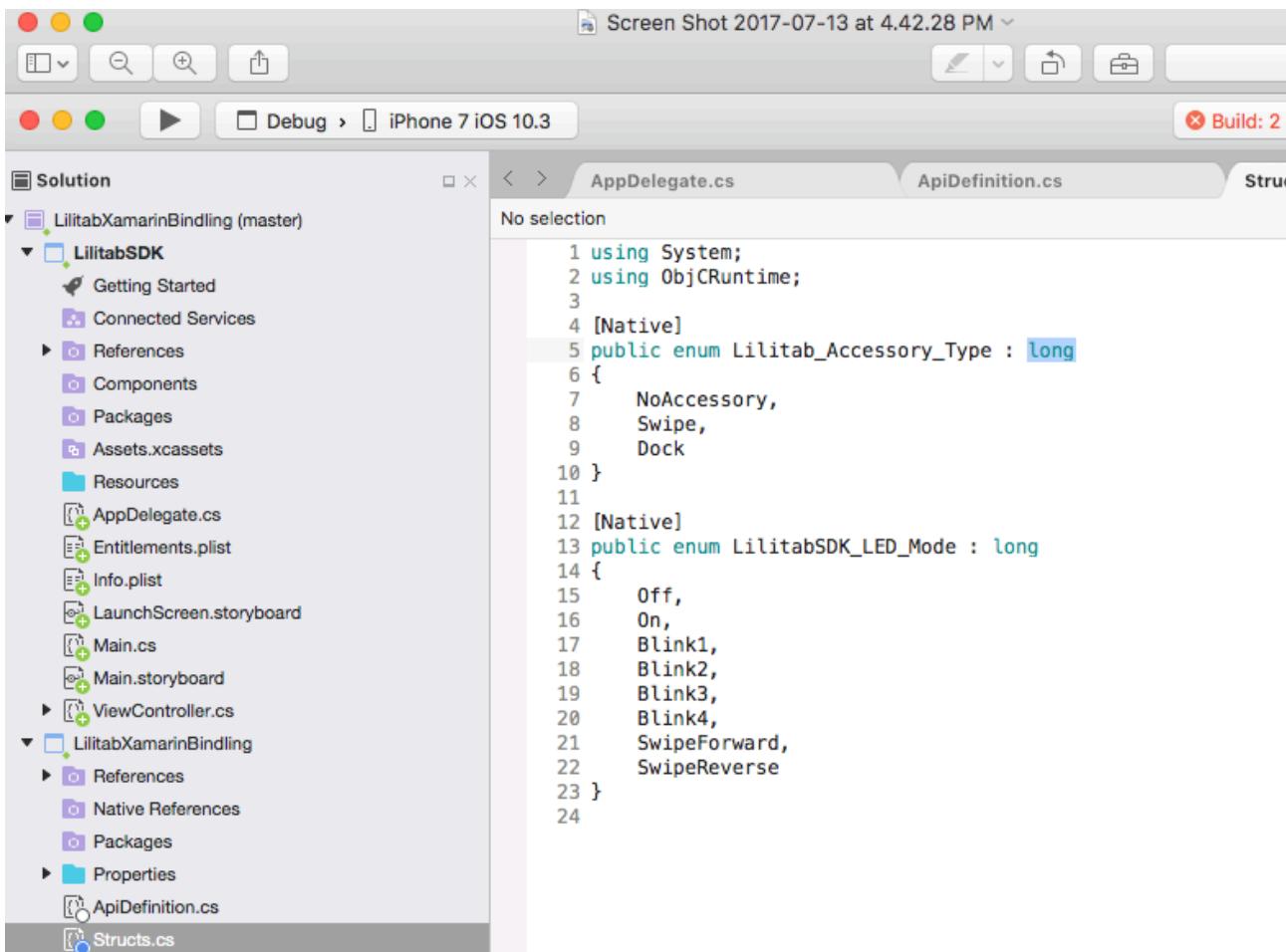
The screenshot shows the Visual Studio IDE interface. On the left is the Solution Explorer pane, which lists the project structure for 'LilitabXamarinBinding' (master). The 'Structs.cs' file is currently selected. The main code editor window displays the following C# code:

```
1 using System;
2 using ObjCRuntime;
3
4 [Native]
5 public enum Lilitab_Accessory_Type : nint
6 {
7     NoAccessory,
8     Swipe,
9     Dock
10 }
11
12 [Native]
13 public enum LilitabSDK_LED_Mode : nint
14 {
15     Off,
16     On,
17     Blink1,
18     Blink2,
19     Blink3,
20     Blink4,
21     SwipeForward,
22     SwipeReverse
23 }
24
```

Two syntax errors are highlighted with red boxes and crossed-out text:

- Line 5: `nint` is underlined with a red squiggle, and a tooltip says "Type byte, sbyte, short, ushort, int, uint, long or ulong expected".
- Line 13: `nint` is underlined with a red squiggle, and a tooltip says "Type byte, sbyte, short, ushort, int, uint, long or ulong expected".

20. 'nint' type in the Structs.cs file is not supported; replace **nint ->: long** and again build the application.



The screenshot shows the Xcode IDE interface. At the top, the status bar indicates "Screen Shot 2017-07-13 at 4.42.28 PM" and the device is set to "iPhone 7 iOS 10.3". The main window has tabs for "AppDelegate.cs", "ApiDefinition.cs", and "Structs.cs". The "Structs.cs" tab is active. The code editor shows the same C# code as the previous screenshot, but the 'nint' types have been replaced with 'long'.

```
1 using System;
2 using ObjCRuntime;
3
4 [Native]
5 public enum Lilitab_Accessory_Type : long
6 {
7     NoAccessory,
8     Swipe,
9     Dock
10 }
11
12 [Native]
13 public enum LilitabSDK_LED_Mode : long
14 {
15     Off,
16     On,
17     Blink1,
18     Blink2,
19     Blink3,
20     Blink4,
21     SwipeForward,
22     SwipeReverse
23 }
24
```

21. Now open ApiDefinition.cs in the **LilitabXamarinBinding**. there are two more error. So just **remove or comment** error lines. because **Verify** is optional in the Xamarin.

```

1 using System;
2
3 using UIKit;
4 using Foundation;
5 using ObjCRuntime;
6 using CoreGraphics;
7
8 namespace LilitabXamarinBinding
9 {
10     [Static]
11     [Verify(ConstantsInterfaceAssociation)] The type or namespace name 'Verify' could not be found. Are you missing an assembly reference?
12     partial interface Constants
13     {
14         // extern double LilitabSDKVersionNumber;
15         [Field("LilitabSDKVersionNumber", "__Internal")]
16         double LilitabSDKVersionNumber { get; }
17
18         // extern const unsigned char [] LilitabSDKVersionString;
19         [Field("LilitabSDKVersionString", "__Internal")]
20         byte[] LilitabSDKVersionString { get; }
21     }
22
23     // @interface LilitabSDK : NSObject
24     [BaseType(typeof(NSObject))]
25     interface LilitabSDK
26     {
27         // +(LilitabSDK *)singleton;
28         [Static]
29         [Export("singleton")]
30         [Verify(MethodToProperty)] The type or namespace name 'Verify' could not be found. Are you missing an assembly reference?
31         LilitabSDK Singleton { get; }
32
33         // @property (readonly, nonatomic) Lilitab_Accessory_Type accessoryType;
34         [Export("accessoryType")]
35         Lilitab_Accessory_Type AccessoryType { get; }
36
37         // @property (readonly, nonatomic) BOOL isAttached;
38         [Export("isAttached")]
39         bool IsAttached { get; }
40
41         // -(void)status:(void (^)(BOOL, NSDictionary *))completionBlock;
42         [Export("status:")]

```

```

//#[Verify(ConstantsInterfaceAssociation)]
//#[Verify(MethodToProperty)]

```

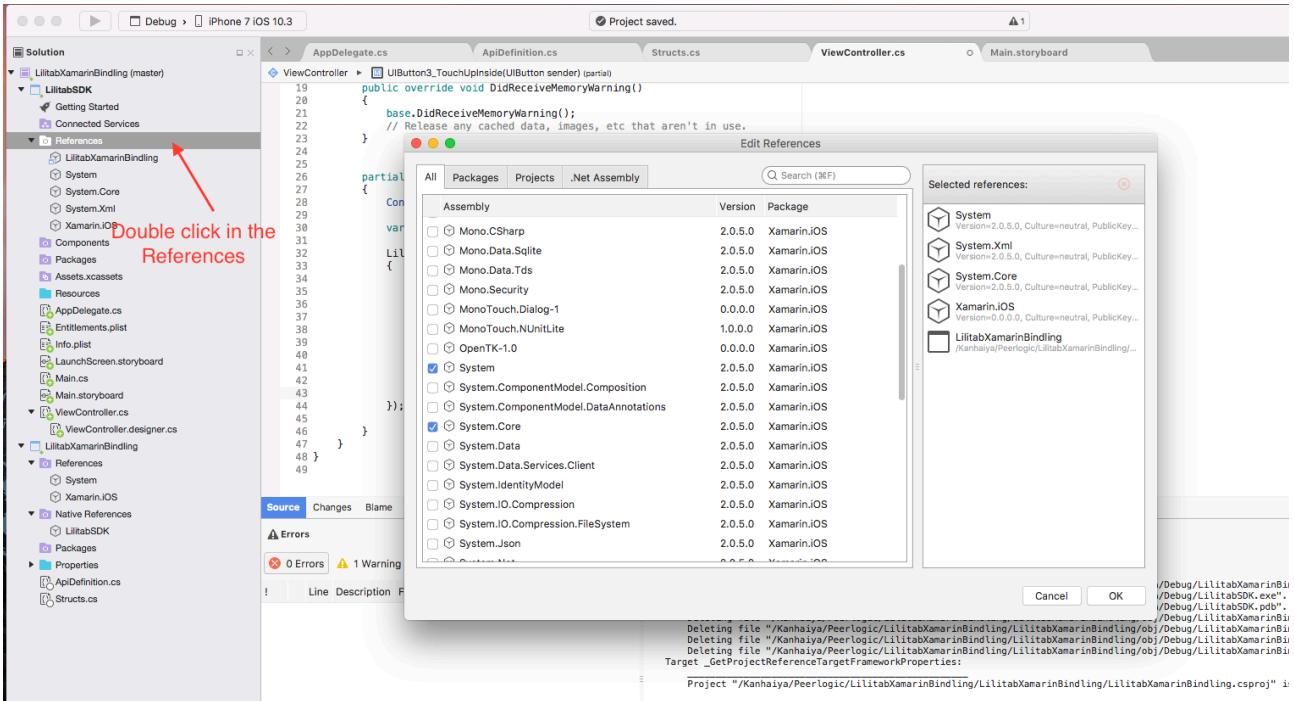
22. Build the application with the simulator.

```

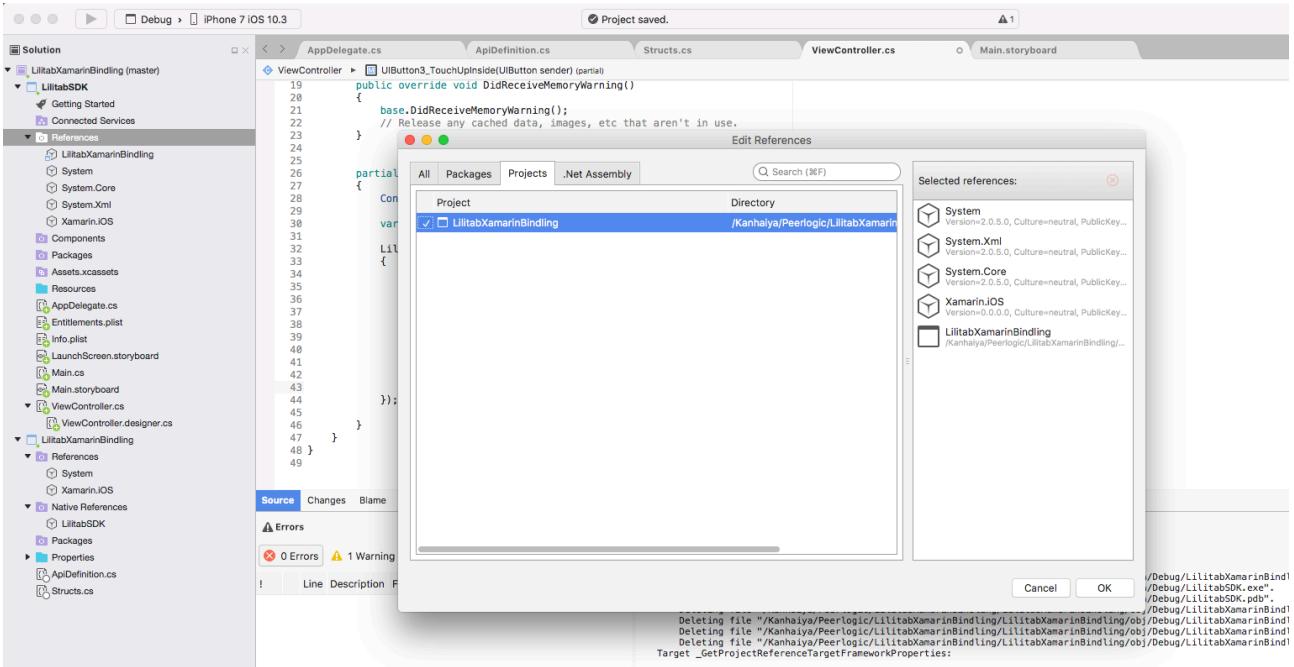
1 using System;
2
3 using UIKit;
4 using Foundation;
5 using ObjCRuntime;
6 using CoreGraphics;
7
8 namespace LilitabXamarinBinding
9 {
10     [BaseType(typeof(NSObject))]
11     interface LilitabSDK
12     {
13         // +(LilitabSDK *)singleton;
14         [Static]
15         [Export("singleton")]
16         //#[Verify(MethodToProperty)]
17         LilitabSDK Singleton { get; }
18
19         // @property (readonly, nonatomic) Lilitab_Accessory_Type accessoryType;
20         [Export("accessoryType")]
21         Lilitab_Accessory_Type AccessoryType { get; }
22
23         // @property (readonly, nonatomic) BOOL isAttached;
24         [Export("isAttached")]
25         bool IsAttached { get; }
26
27         // -(void)status:(void (^)(BOOL, NSDictionary *))completionBlock;
28         [Export("status:")]
29         void Status(Action<bool, NSDictionary> completionBlock);
30
31         // @property (assign, nonatomic) LilitabSDK_LED_Mode ledState;
32         [Export("ledState", ArgumentSemantic.Assign)]
33         LilitabSDK_LED_Mode LedState { get; set; }
34
35         // @property (assign, nonatomic) NSTimeInterval noPowerBlinkDuration;
36         [Export("noPowerBlinkDuration")]
37         double NoPowerBlinkDuration { get; set; }
38

```

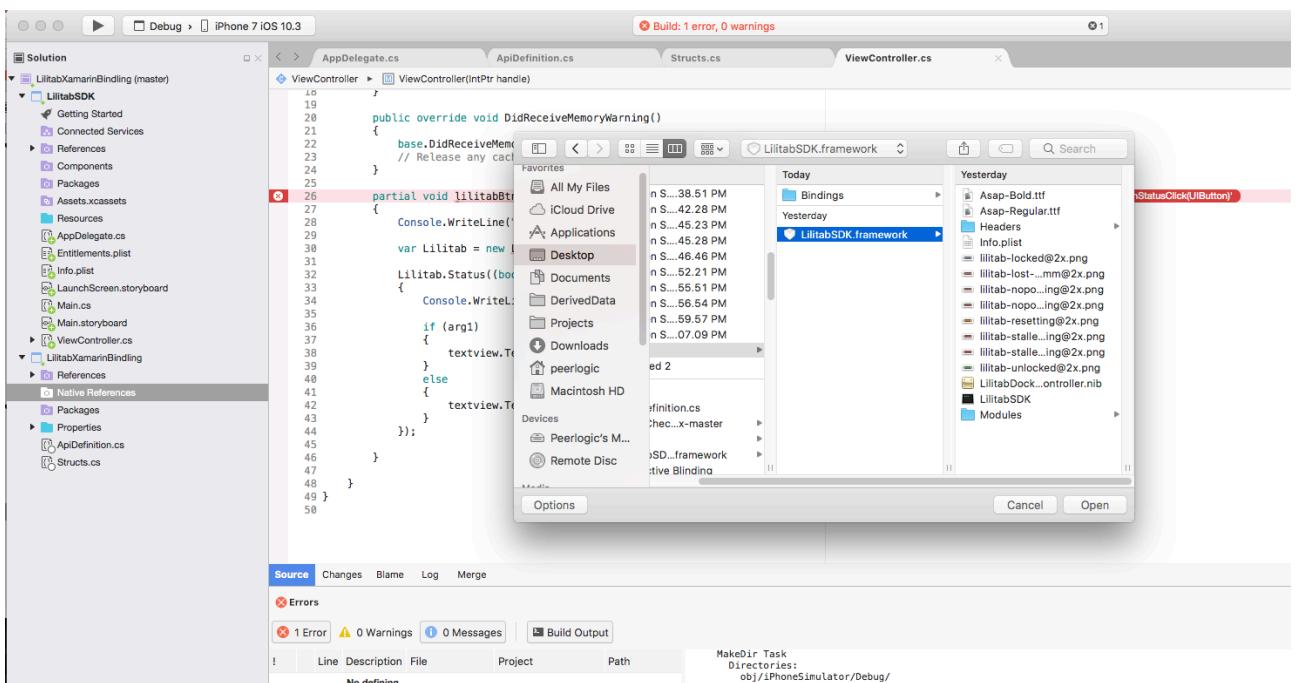
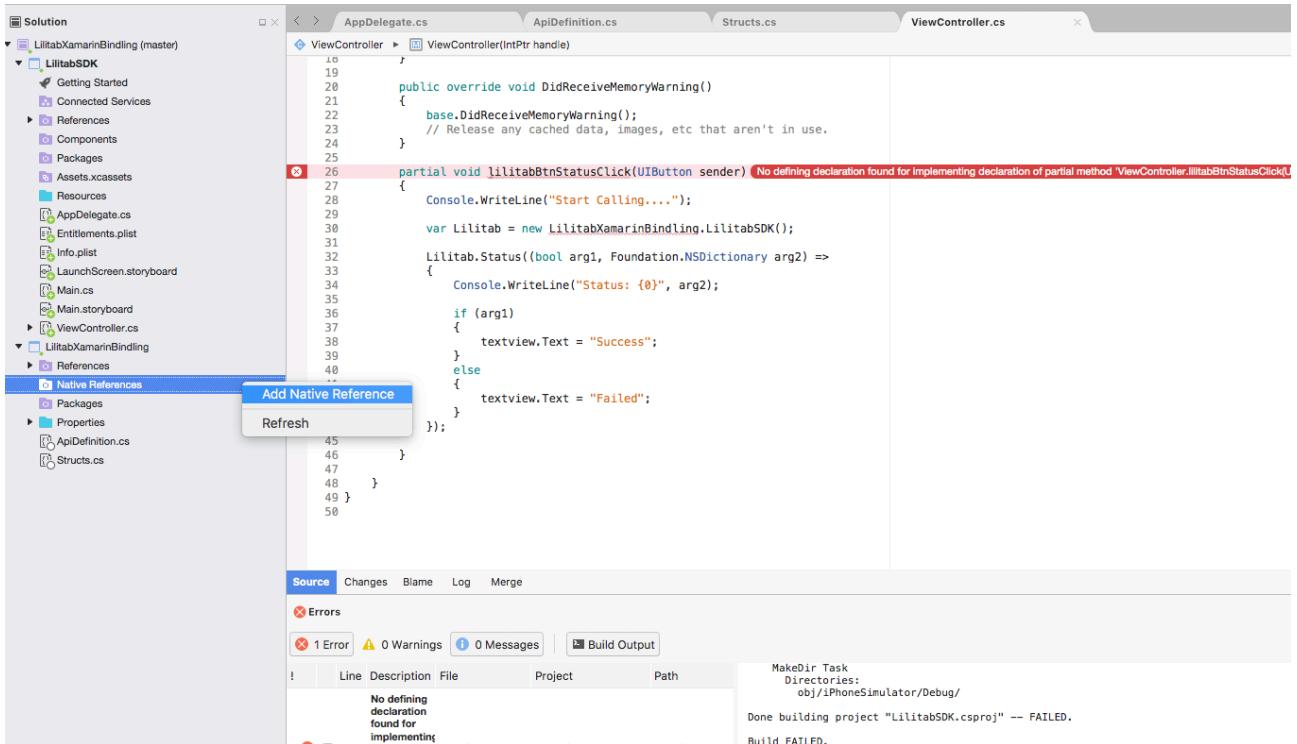
23. Double click on References in the LilitabSDK target.



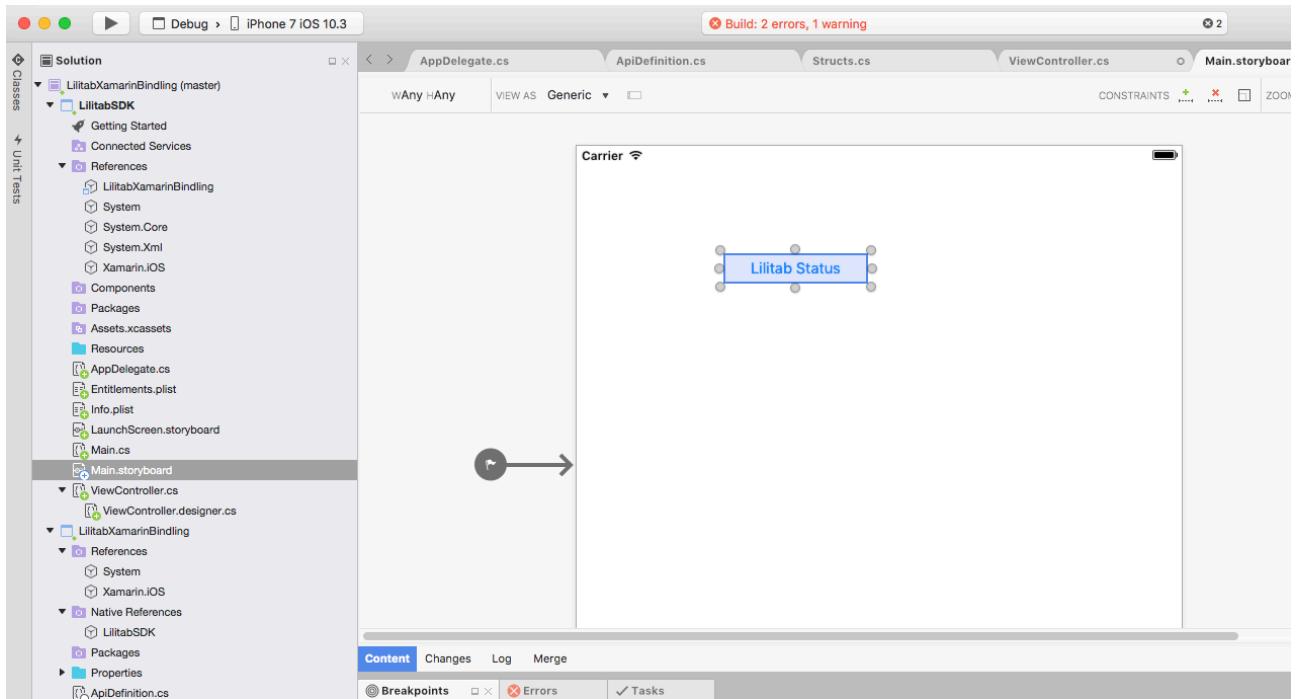
24. Click on Projects tab in the top and select the check box in the option and click on OK button.



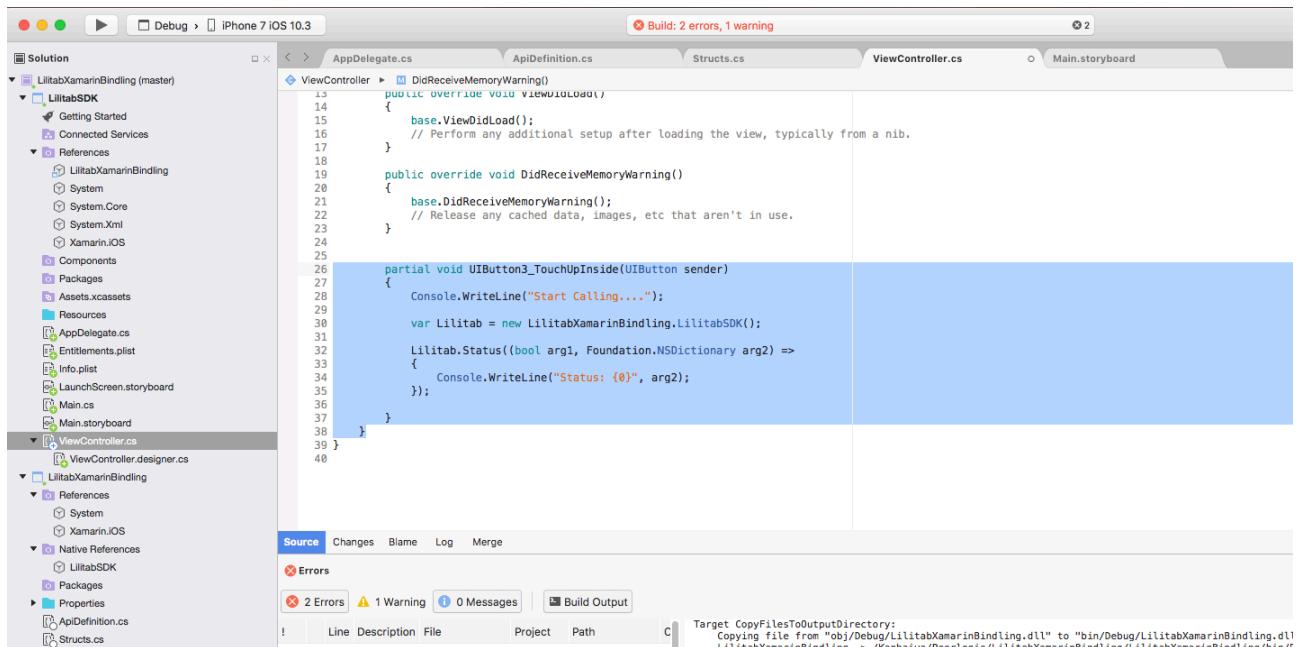
25. Right click on Native Reference option and import LilitabSDK.framework.



26. Go to main.storyBoard -> add one button in the screen. And double click in the button to make a connection.



26. Go to ViewControlller and paste below code in the Button action method.



```
Console.WriteLine("Start Calling....");

    var Lilitab = new LilitabXamarinBindling.LilitabSDK();

        Lilitab.Status((bool arg1, Foundation.NSDictionary
arg2) =>
    {
        Console.WriteLine("Status: {0}", arg2);
    });
}
```