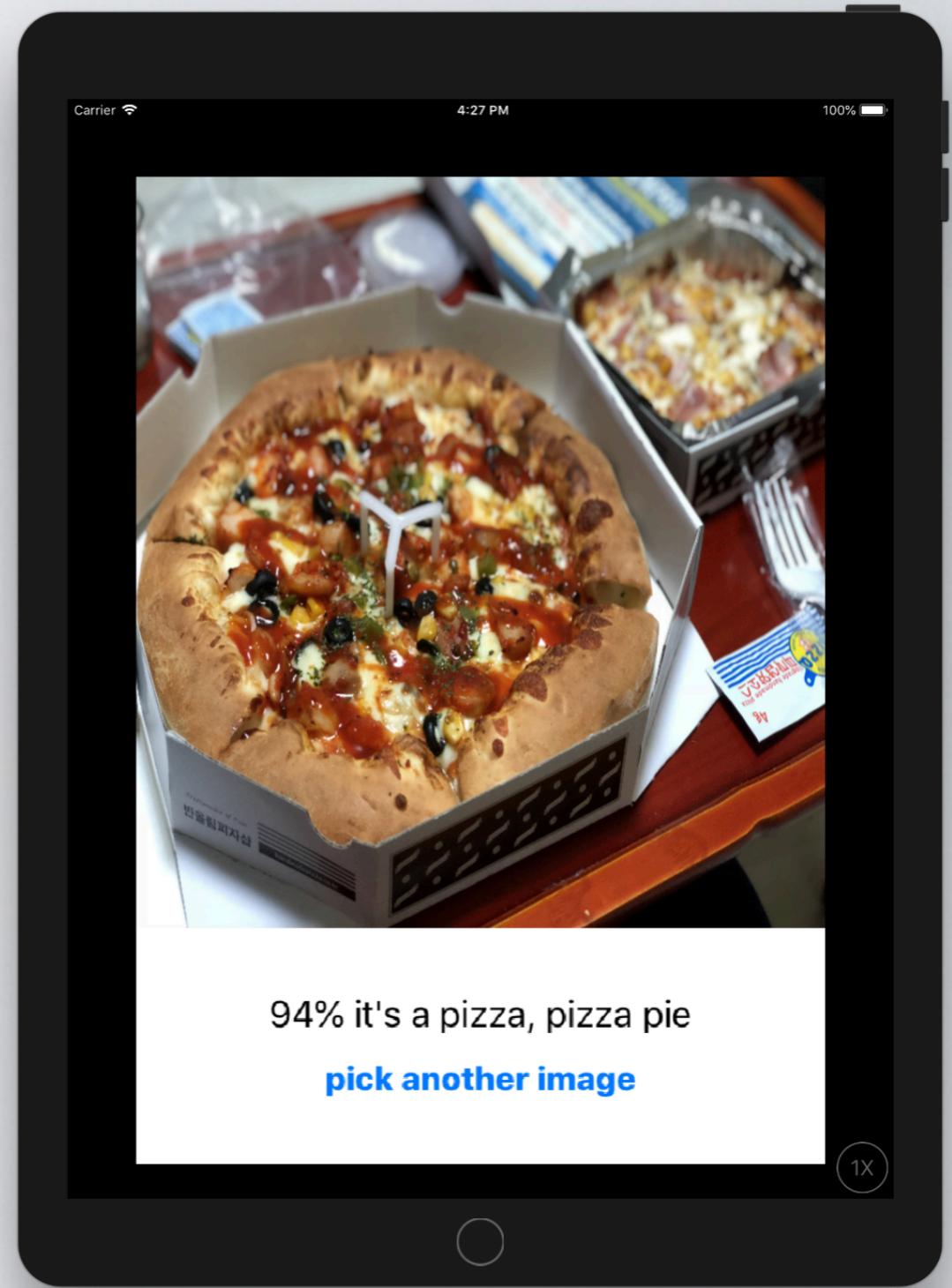


On-Device Tutorial With Core ML

ICT Application Development



iPad (5th generation) - 11.3



Developer

First, you have to assign apple developer account

Apple Inc.

Developer 둘러보기 디자인 개발 배포 지원 Account

iPhone X

앱 준비하기 >



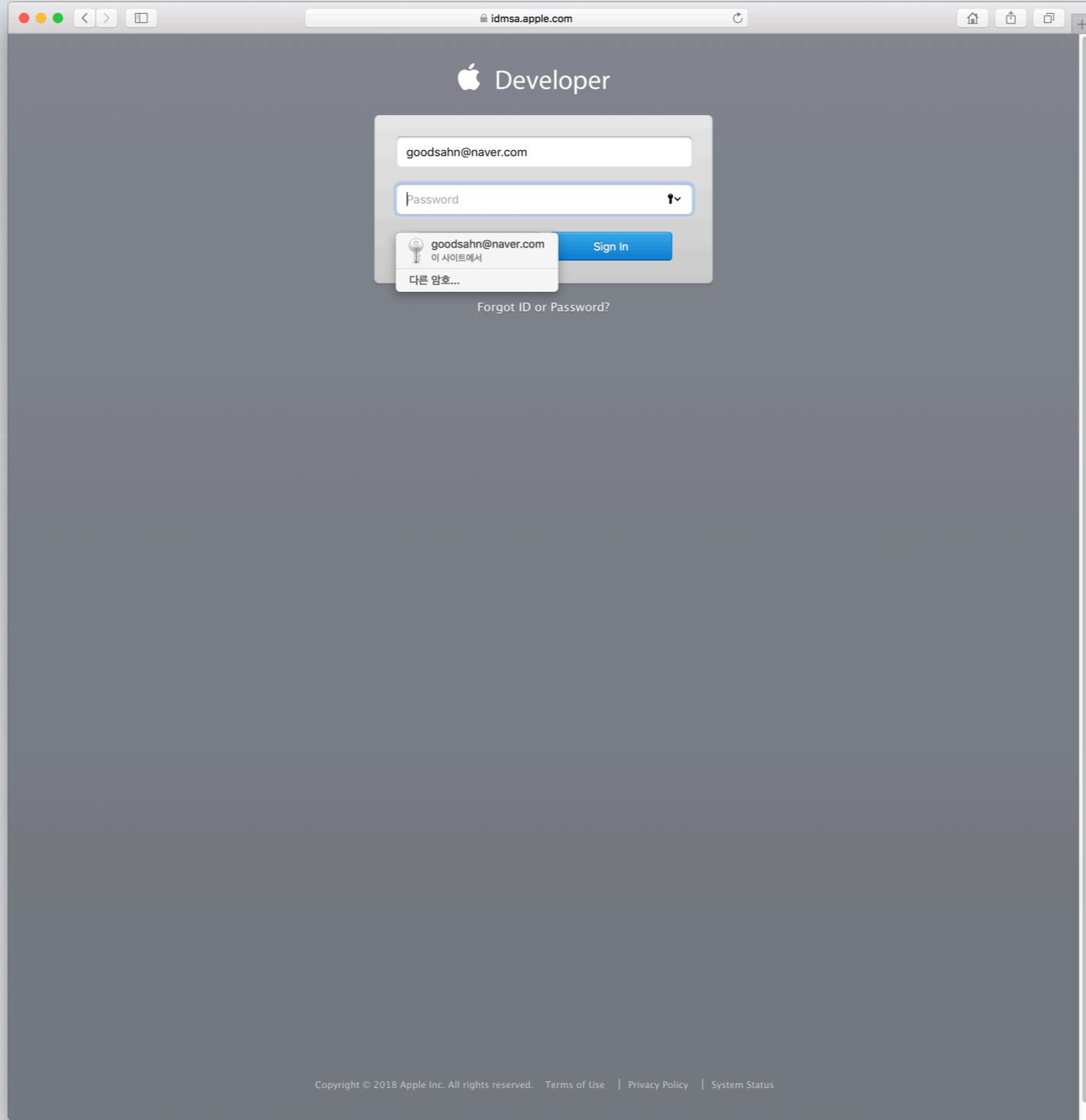
https://developer.apple.com

< >

Apple Developer

둘러보기	디자인	개발	배포	지원
macOS	Human Interface Guidelines	Xcode	App Store	Developer Forums
iOS	Resources	Swift	iTunes Connect	문의하기
watchOS	Videos	Documentation	TestFlight	Bug Reporting
tvOS	Apple Design Awards	Videos	Enterprise	License Agreements
Developer Program	Accessibility	Downloads	Safari Extensions	System Status
Enterprise	국제화			마케팅 지침
Education	액세서리 디자인			

최신 개발자 소식을 받으려면 [뉴스 및 업데이트](#)에서 구독을 신청하십시오.



You can sign in with your own Apple ID

This is a legal agreement between you and Apple.

[Download PDF](#)

THIS IS A LEGAL AGREEMENT BETWEEN YOU AND APPLE INC. ("APPLE") STATING THE TERMS THAT GOVERN YOUR PARTICIPATION AS AN APPLE DEVELOPER. PLEASE READ THIS APPLE DEVELOPER AGREEMENT ("AGREEMENT") BEFORE PRESSING THE "AGREE" BUTTON AND CHECKING THE BOX AT THE BOTTOM OF THIS PAGE. BY PRESSING "AGREE," YOU ARE AGREEING TO BE BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, PRESS "CANCEL".

Apple Developer Agreement

1. Relationship With Apple; Apple ID and Password. You understand and agree that by registering with Apple, you will become an Apple Developer ("Apple Developer"), no legal partnership or agency relationship is created between you and Apple. You agree not to represent otherwise. You also certify that you are at least thirteen years of age and you represent that you are legally permitted to register as an Apple Developer. This Agreement is void where prohibited by law and the right to register as an Apple Developer is not granted in such jurisdictions. Unless otherwise agreed or permitted by Apple, you will not resell your Apple Developer account.

By checking this box I confirm that I have read and agree to be bound by the Agreement above.

I'd like to receive marketing emails to stay up-to-date with Apple Developer news.

[Cancel](#)

Welcome

- Additional Resources
- Documentation
- Downloads
- Forums
- Bug Reporter
- News & Updates

Getting Started

Download Xcode, learn how to build an app, and install it directly on your Apple device.

Download Tools
Get started with Xcode, Apple's integrated development environment for creating apps.

Build Your First App
Use Xcode to write a simple "Hello World" app to get familiar with tools, SDKs, and the Swift language.

Apple Developer

Discover Design Develop Distribute Support Account

Additional Resources

macOS Resources Xcode App Store Developer Forum
iOS Videos Swift App Review Contact Us
watchOS Apple Design Awards Documentation iTunes Connect Bug Reporting
tvOS Accessibility Videos TestFlight License Agreements
Developer Program Internationalization Downloads Enterprise System Status
Enterprise Accessories Downloads Safari Extensions
Education Marketing Guidelines

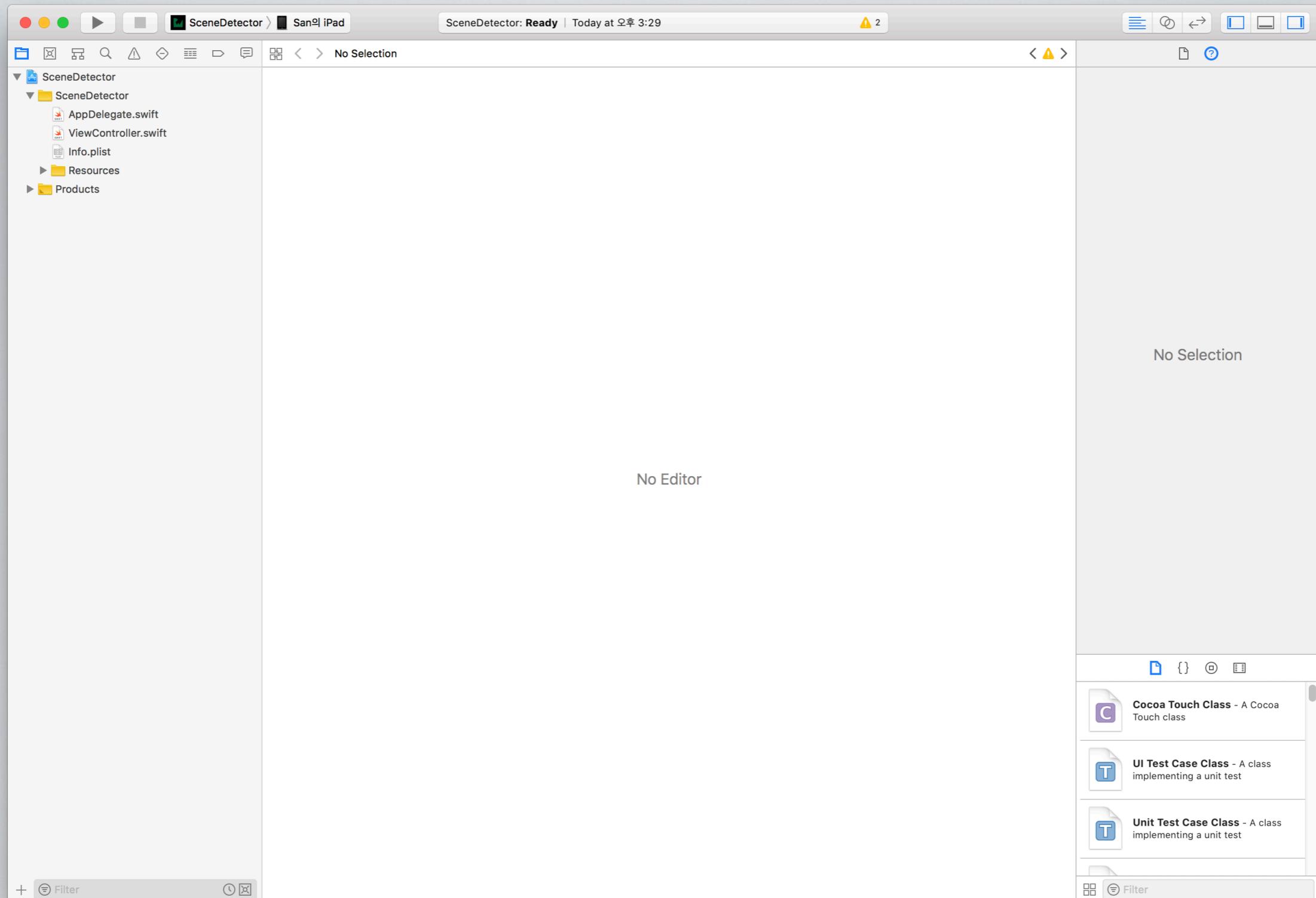
To receive the latest developer news, visit and subscribe to our [News and Updates](#).

Copyright © 2018 Apple Inc. All rights reserved. [Terms of Use](#) | [Privacy Policy](#) | [Report Bugs](#) | [Feedback](#)

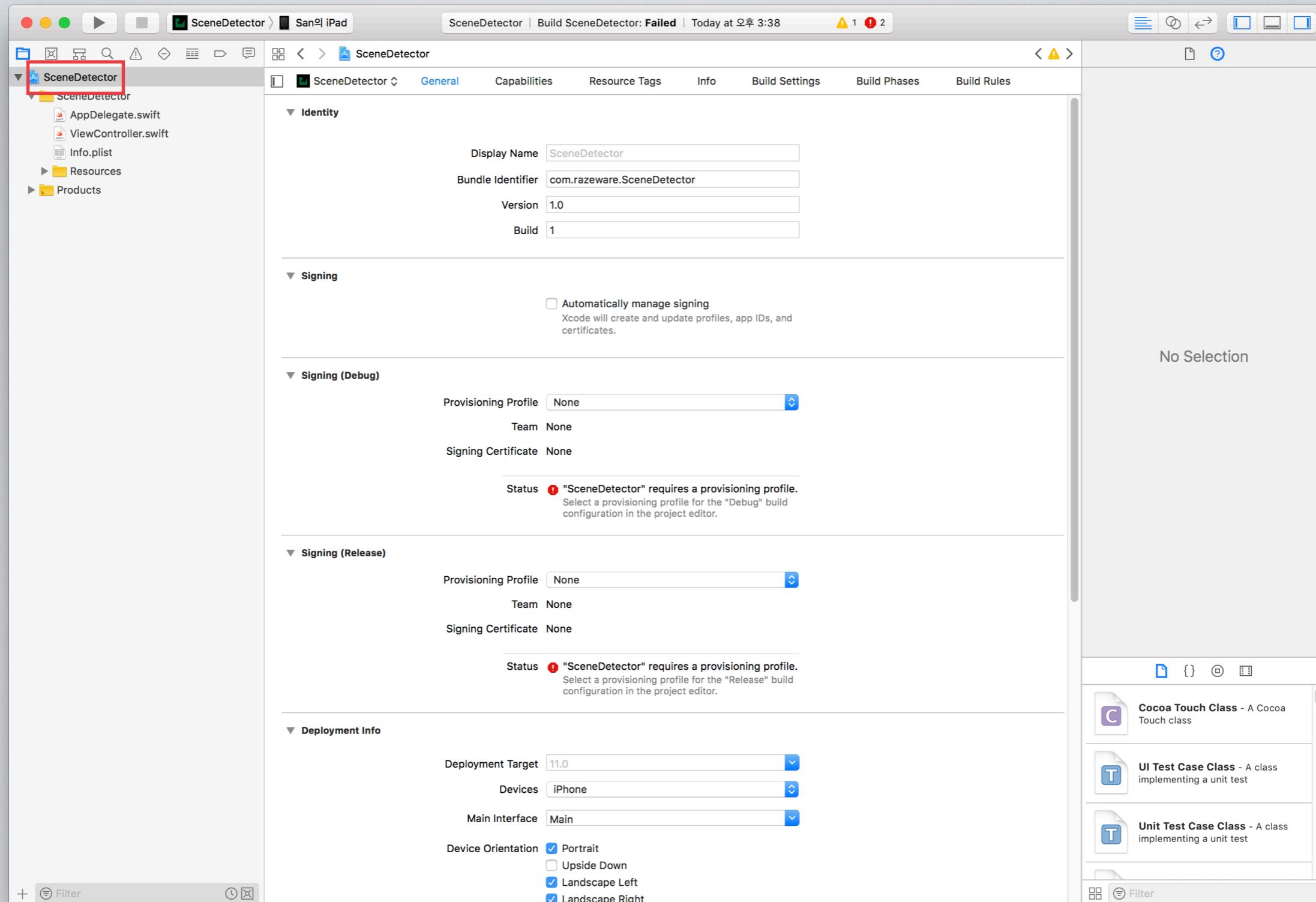
Join the Apple Developer Program

Membership in the Apple Developer Program includes everything you need to develop, distribute, and manage your apps on the App Store. You'll also gain access to beta software, advanced app capabilities, beta testing tools, and app analytics.

You can download various kit information and programs from apple developer site

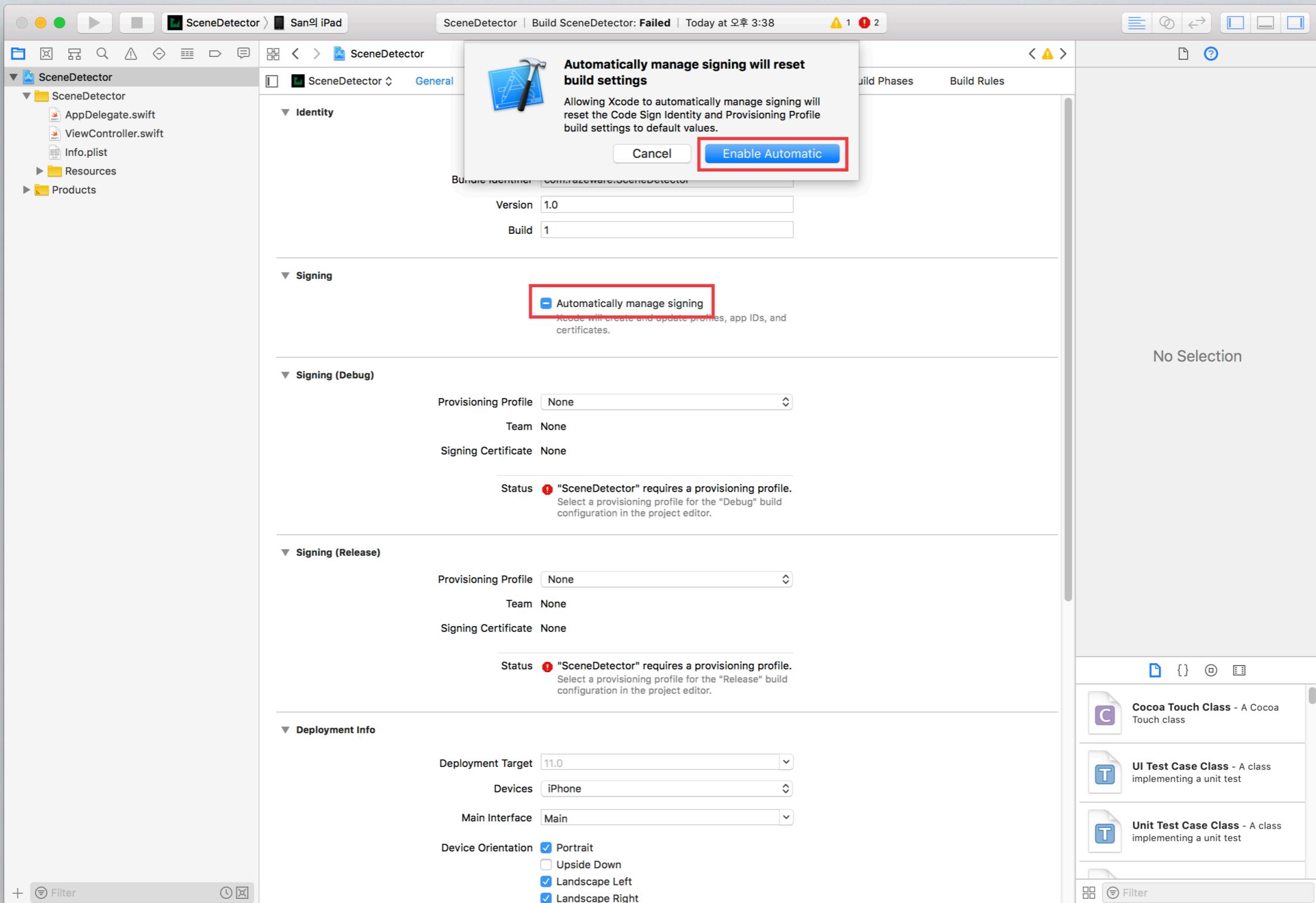


Run the Xcode
(your project or example project)

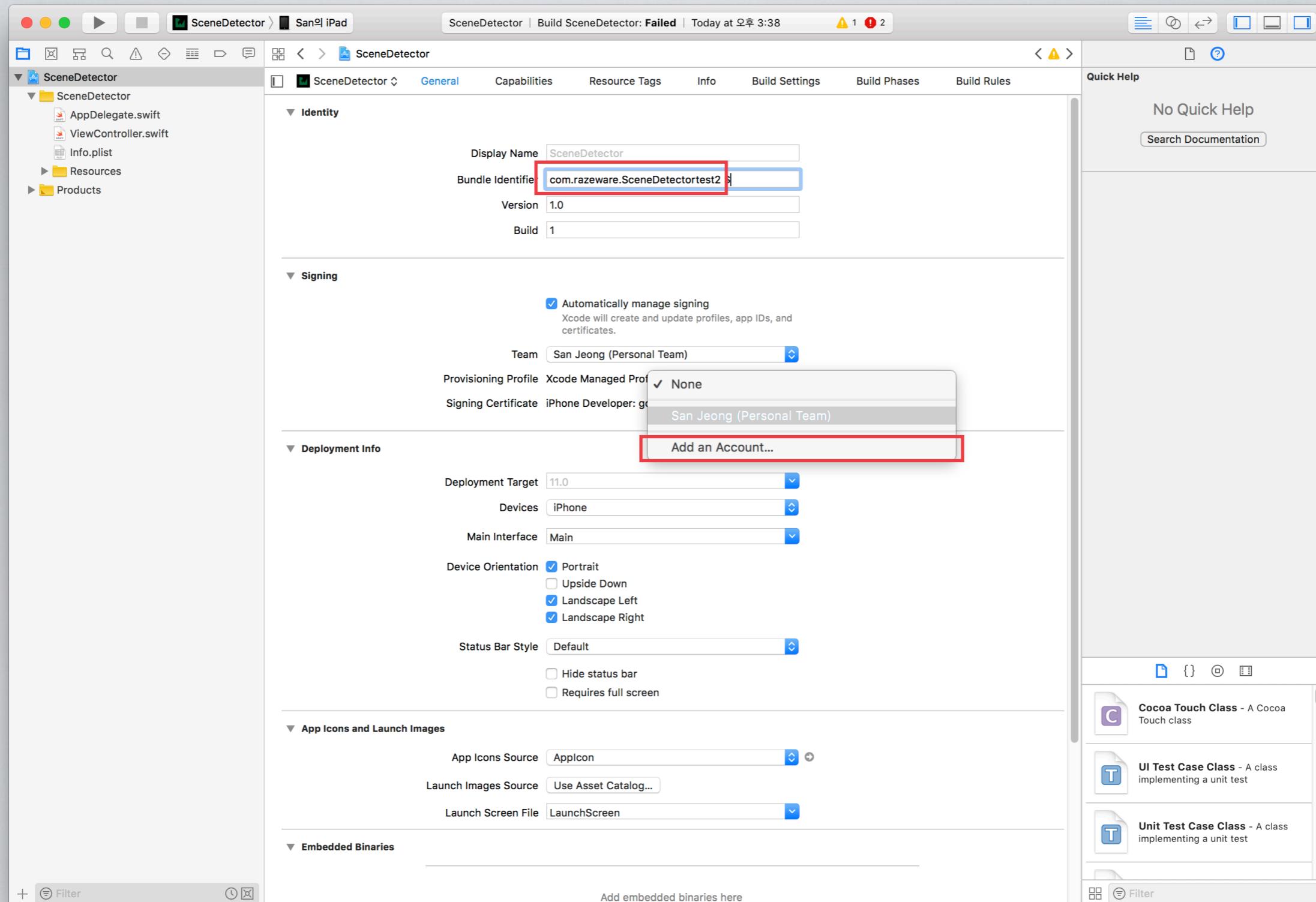


Some things to do before you project your application

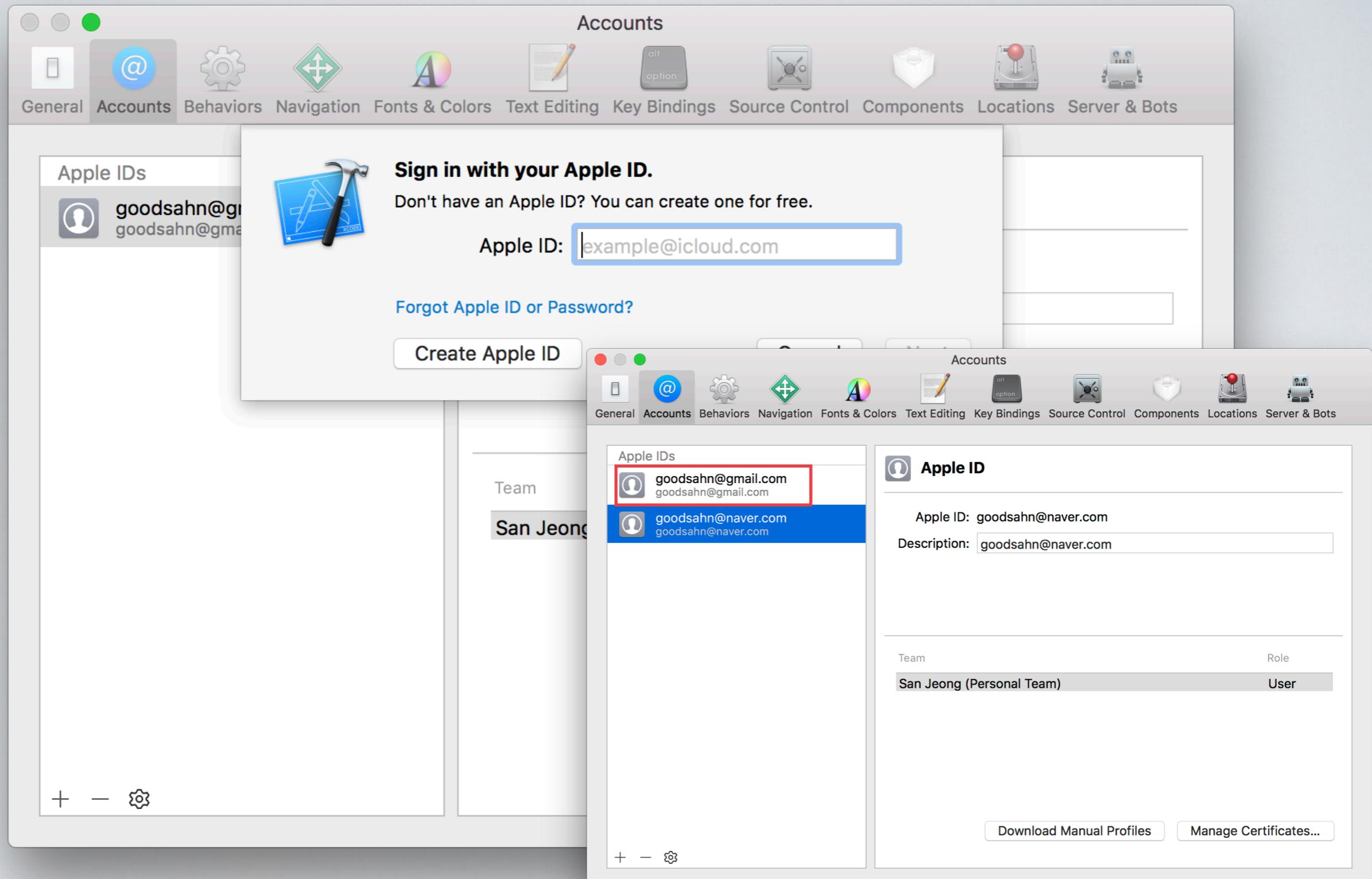
1. Your project needs a unique identifier and a unique team
2. Your project and device must have the same IOS version



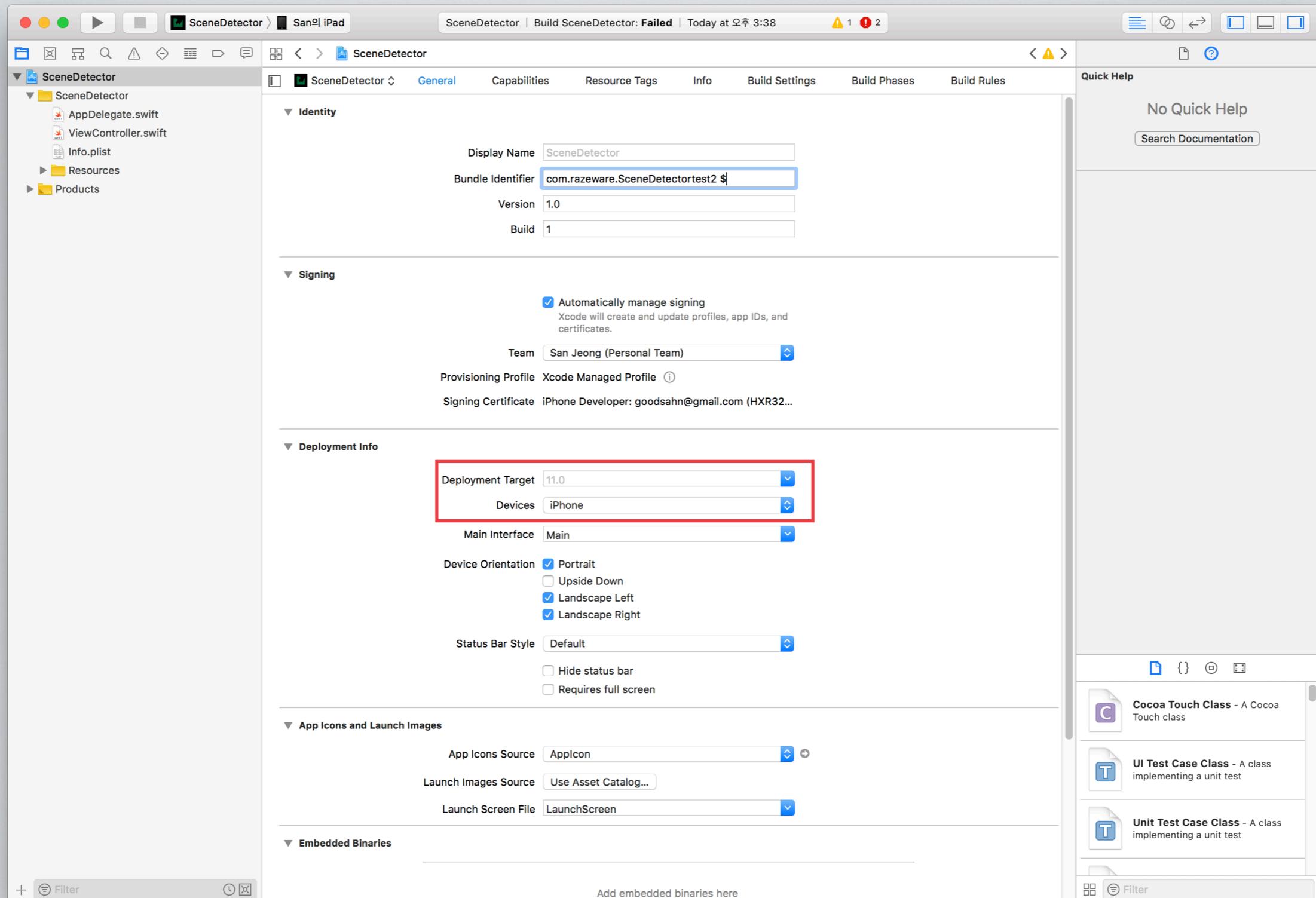
I. Your project needs a unique identifier and a unique team



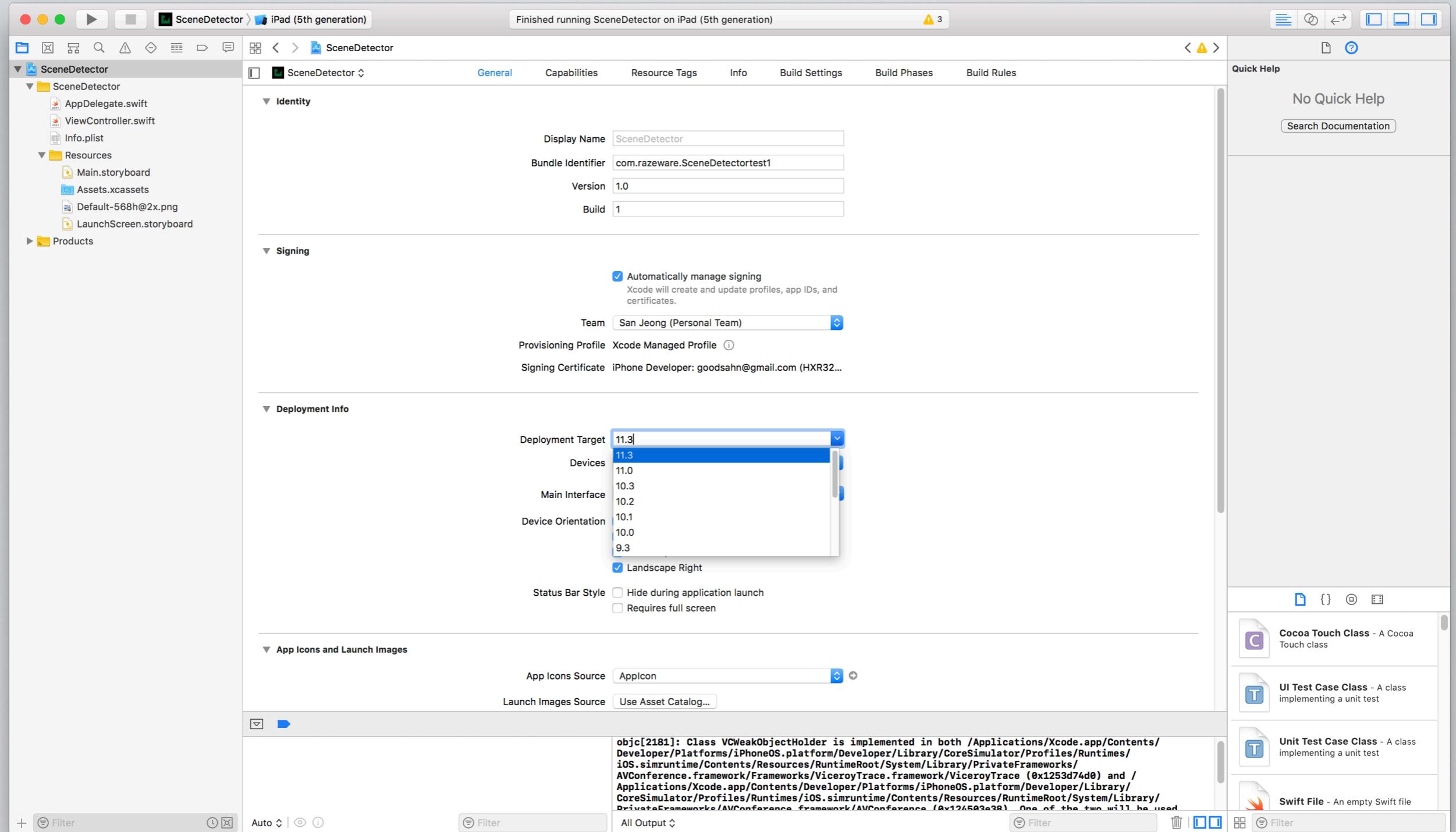
The name of the *Bundle identifier* must be a unique name, and the *Team* must be a personal or a specific account.



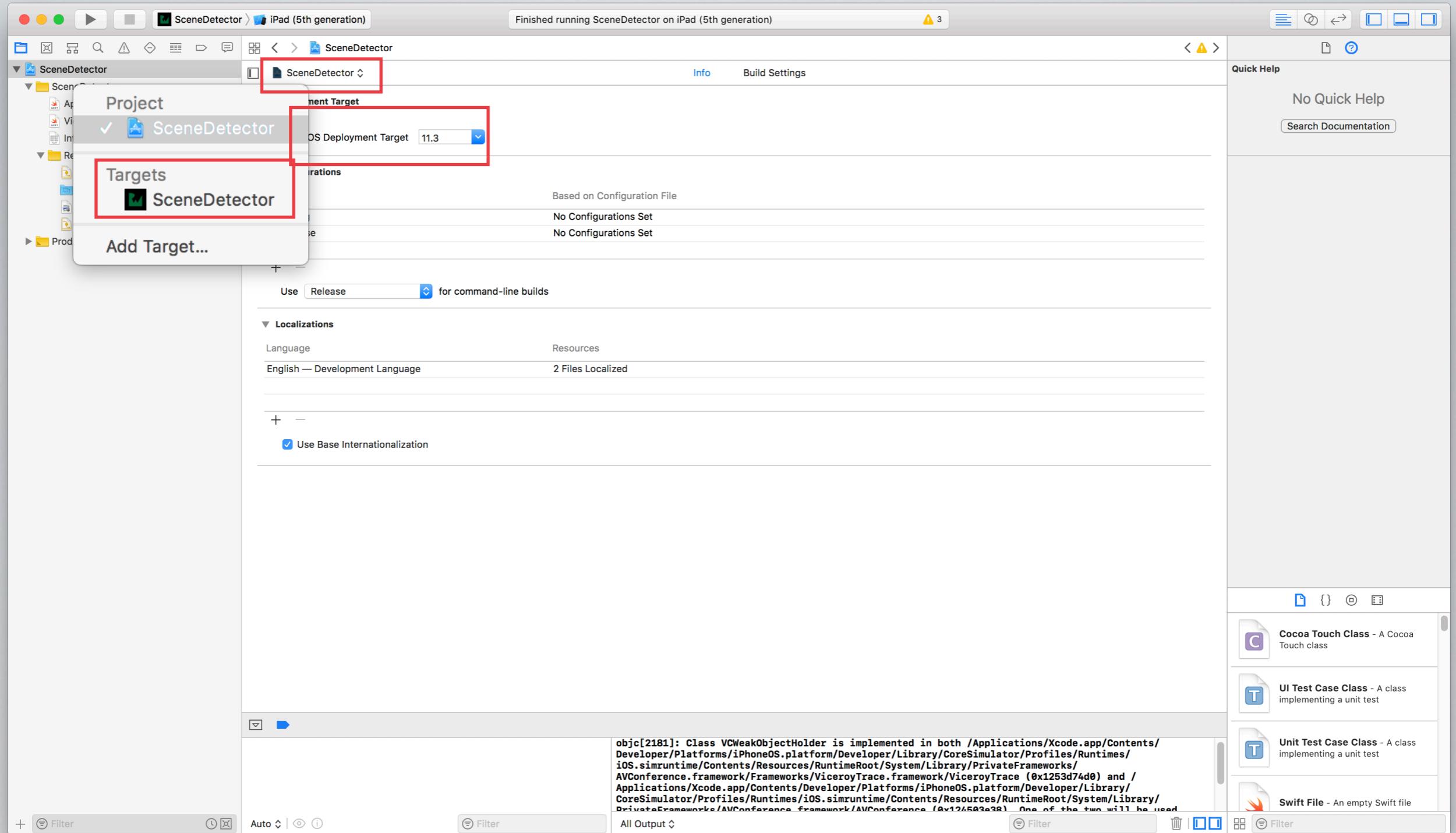
You have to add your own developer account to your Xcode



2. Your project and device must have the same IOS version



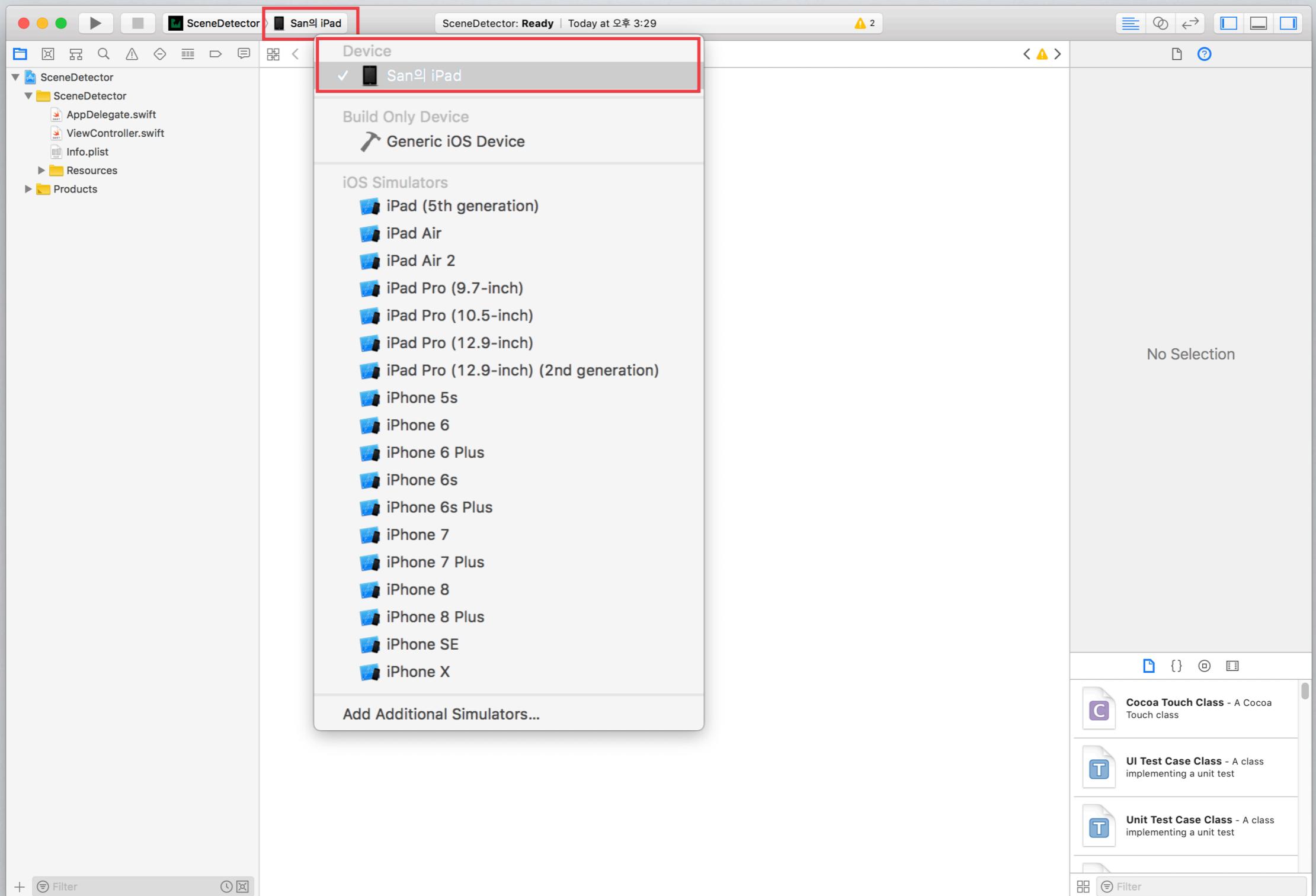
It is recommended that both the project and the target be unified with the iOS version



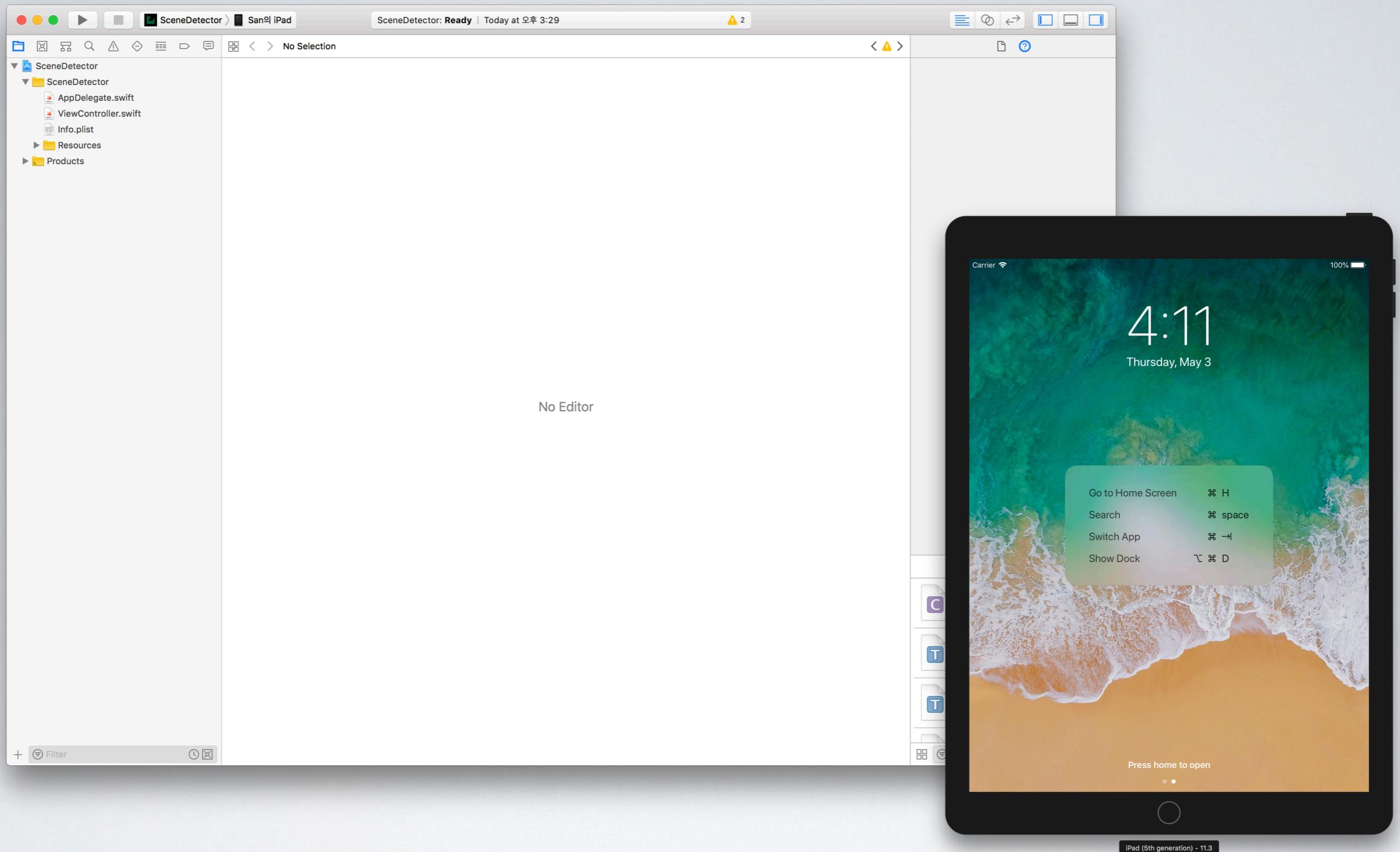
It is recommended that both the project and the target be unified with the iOS version



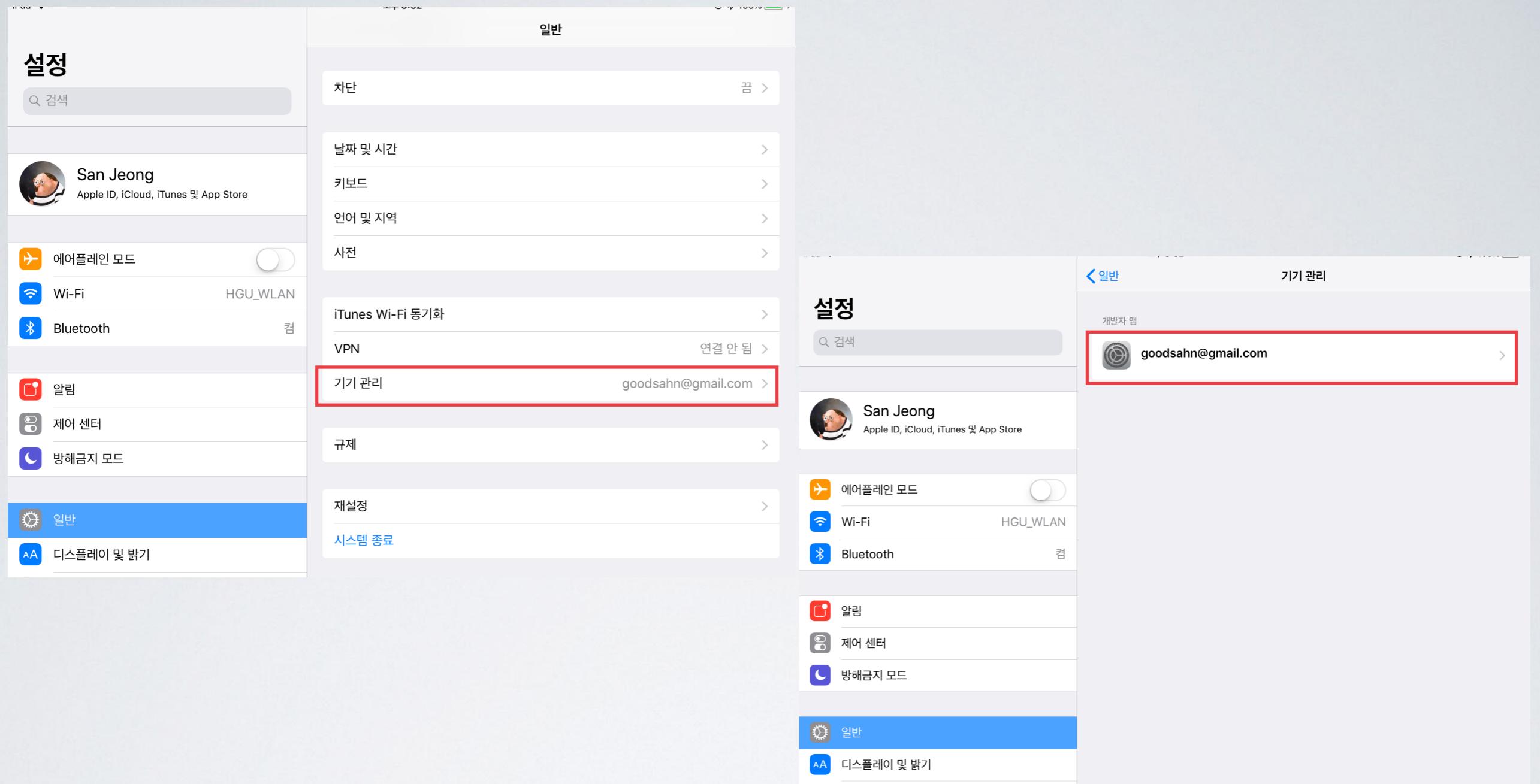
Connect iPad and Mac with lightning cable



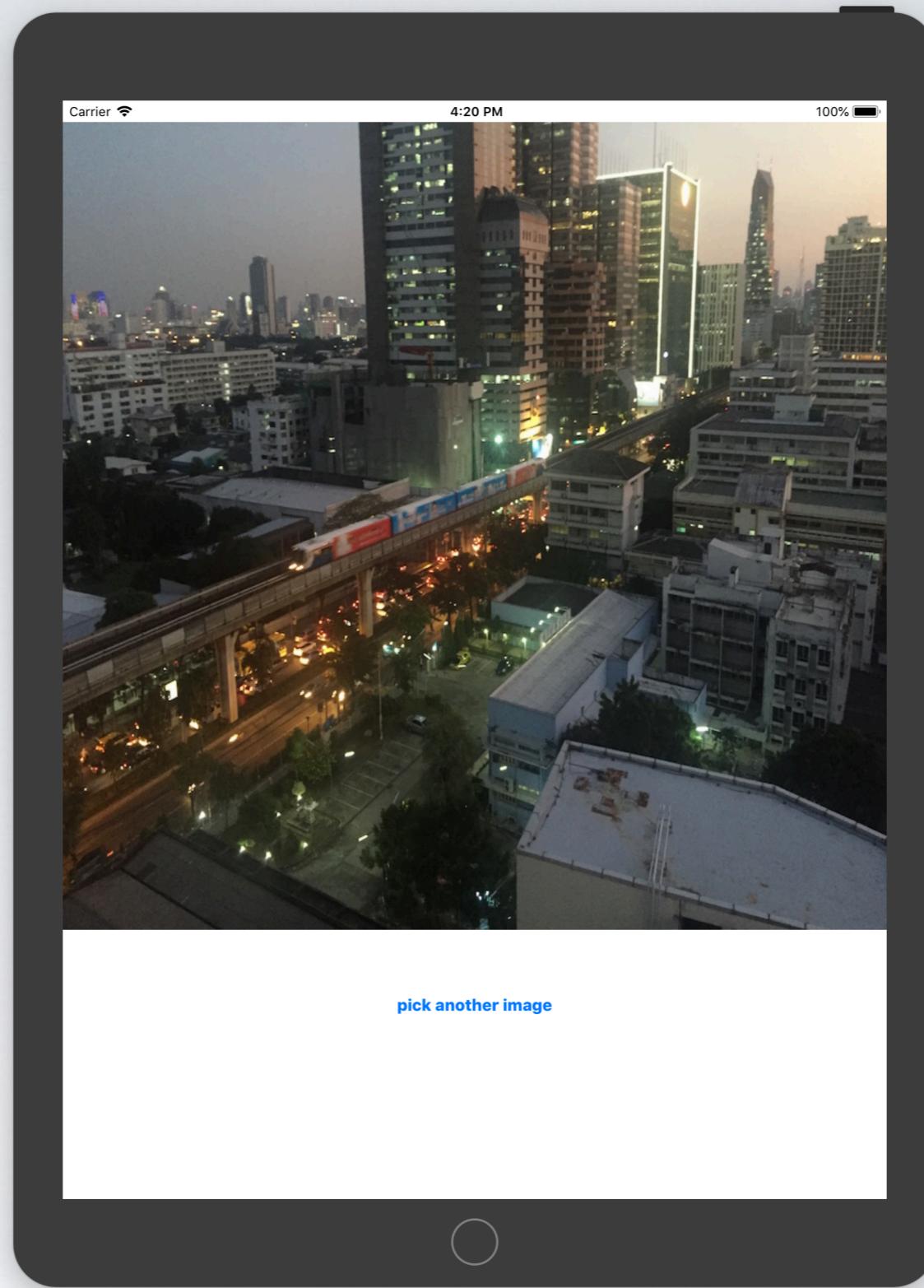
Click top left and select IPAD from Device tab



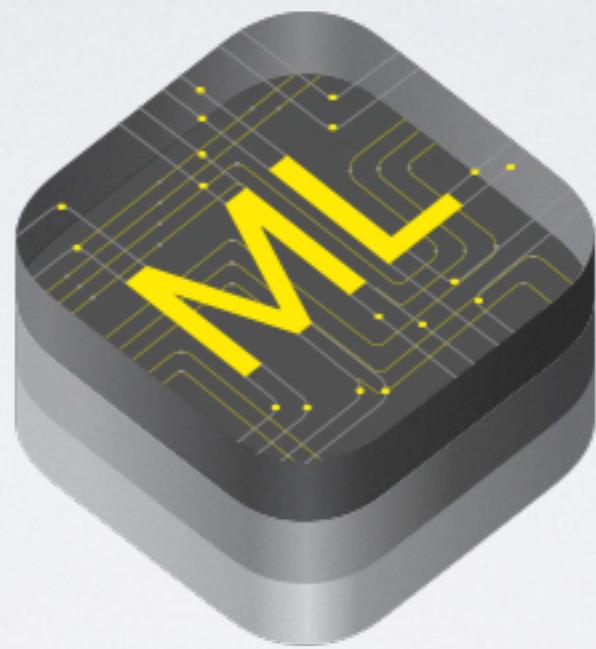
Unlock the iPad and run the project



In the Settings - General - Device Management tab,
You must give permission for the application



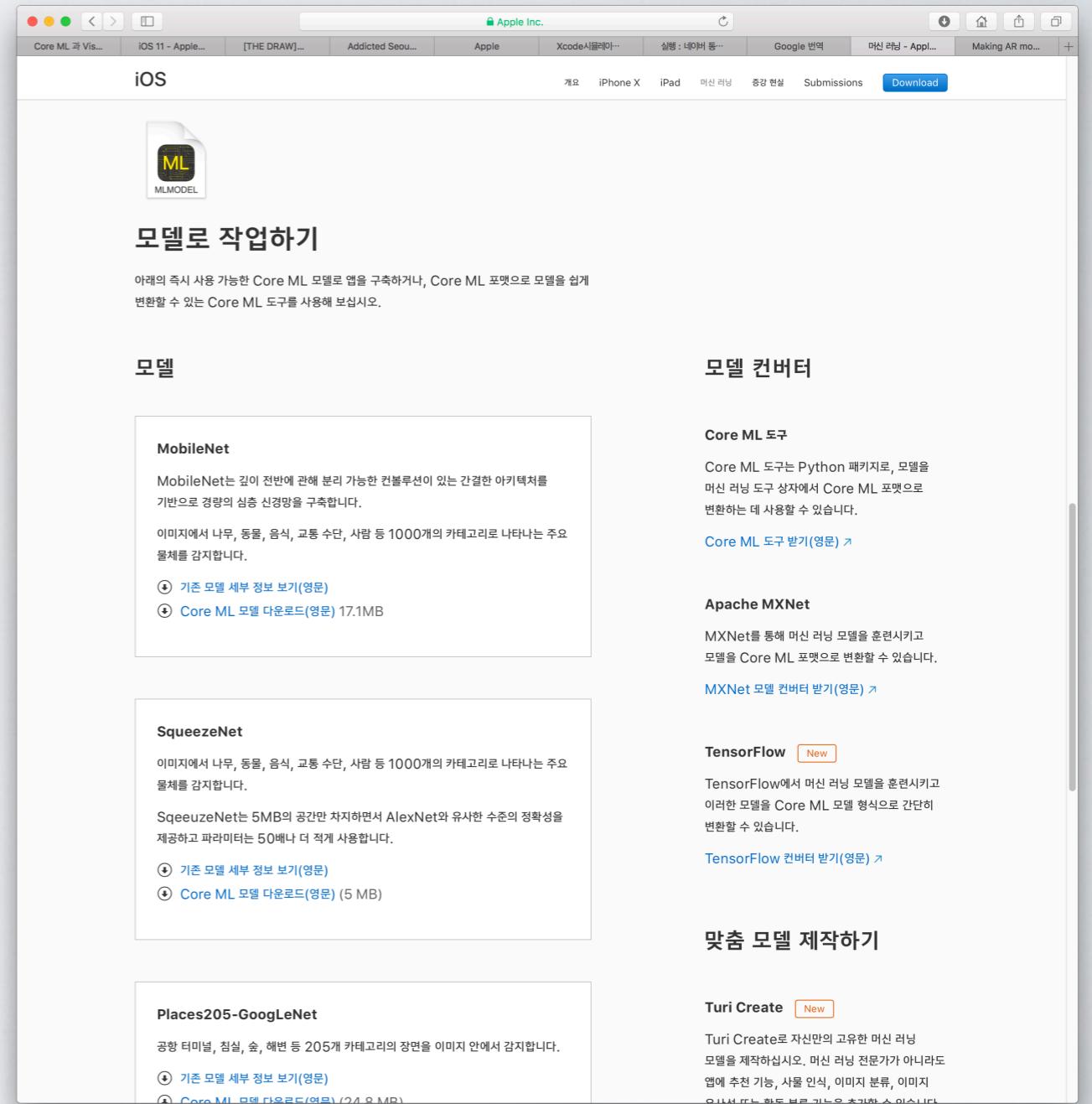
Now you can run your application on the iPad !

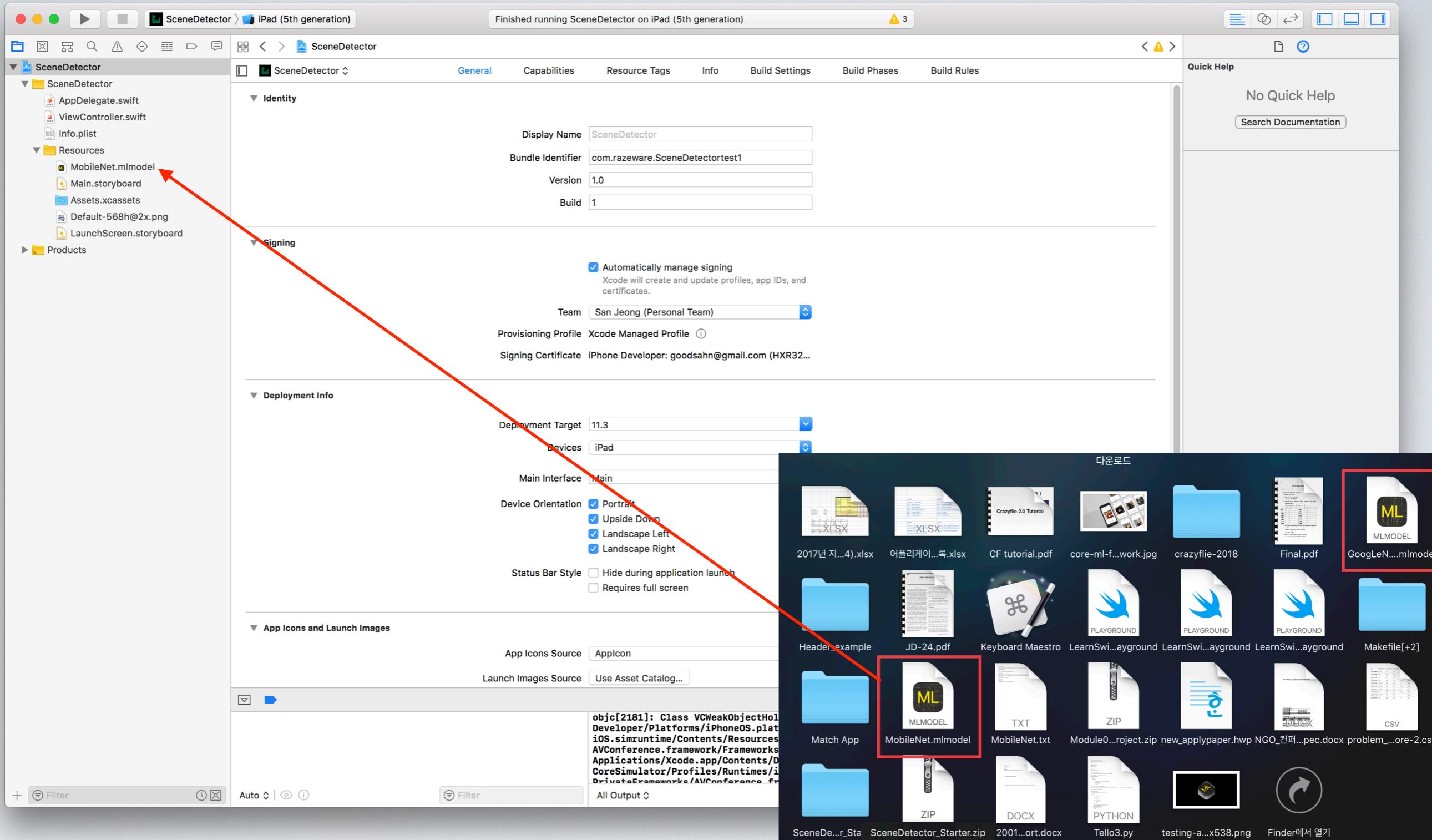


Core ML tutorial

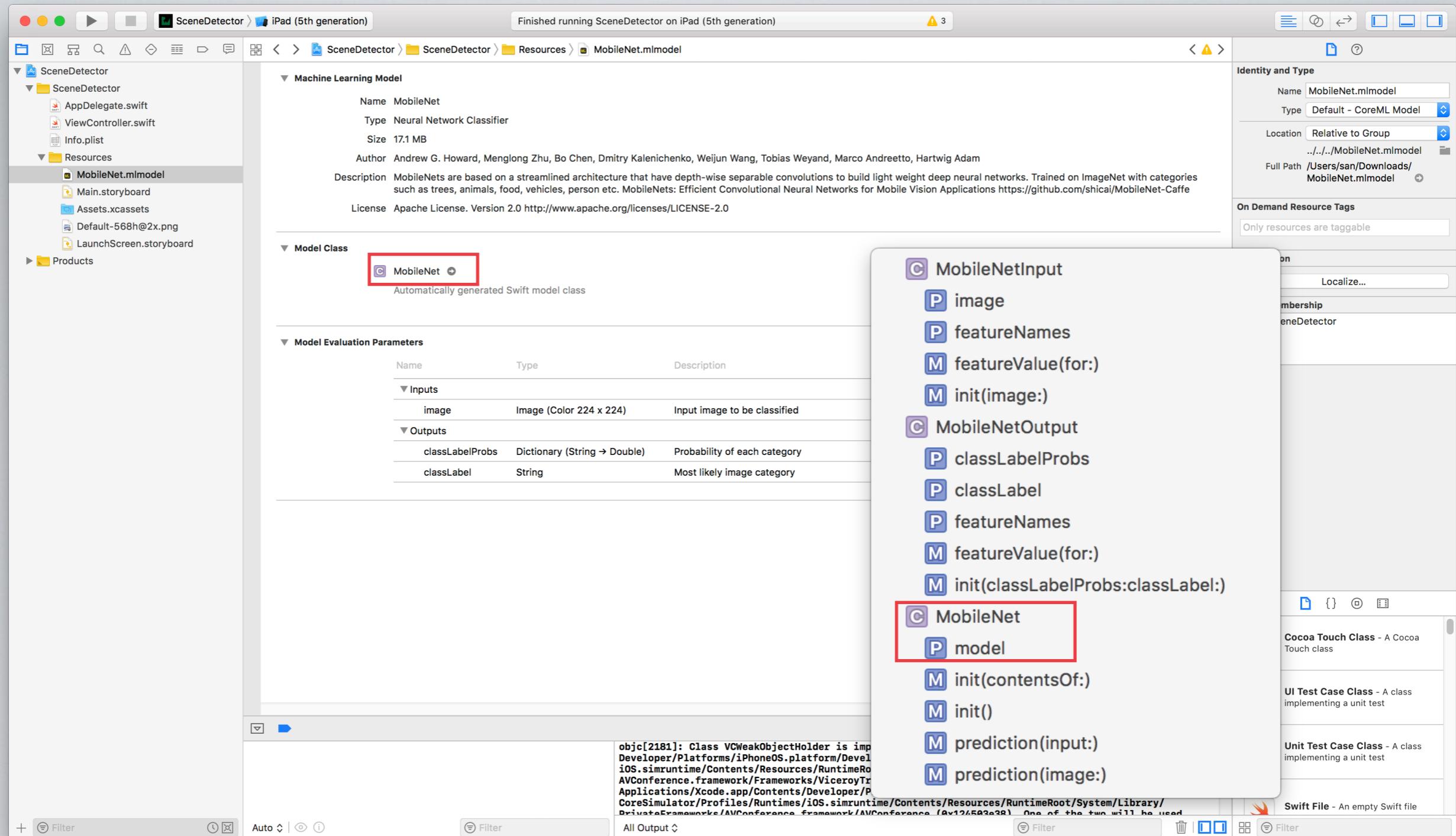
Preparing Models

- Core ML Link
- <https://developer.apple.com/kr/machine-learning/>
- You can download an already created model, or you can convert a previously learned model and insert it into your project,





Drag & Drop .mlmodel file to resources directory

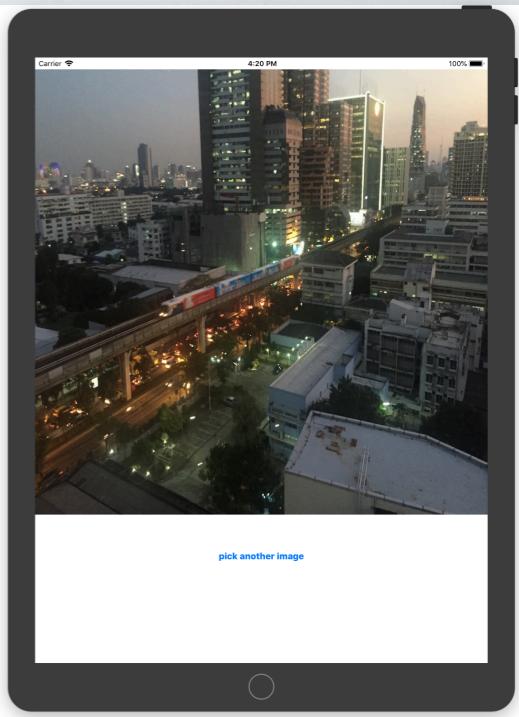


Click the arrow to see the model class
 You can see the classes and methods created by Xcode

Example project has the ability to import images from the photobook and display them in the interface.

There is already a Privacy - Photo Library Usage Description item in info.plist so you can access the photo album without problems.

```
31 import UIKit
32
33 class ViewController: UIViewController {
34
35     // MARK: - IBOutlets
36     @IBOutlet weak var scene: UIImageView!
37     @IBOutlet weak var answerLabel: UILabel!
38
39     // MARK: - View Life Cycle
40     override func viewDidLoad() {
41         super.viewDidLoad()
42
43         guard let image = UIImage(named: "train_night") else {
44             fatalError("no starting image")
45         }
46
47         scene.image = image
48     }
49 }
50
51 // MARK: - IBActions
52 extension ViewController {
53
54     @IBAction func pickImage(_ sender: Any) {
55         let pickerController = UIImagePickerController()
56         pickerController.delegate = self
57         pickerController.sourceType = .savedPhotosAlbum
58         present(pickerController, animated: true)
59     }
60 }
61
62 // MARK: - UIImagePickerControllerDelegate
63 extension ViewController: UIImagePickerControllerDelegate {
64
65     func imagePickerController(_ picker: UIImagePickerController, didFinishPickingMediaWithInfo info: [String : Any]) {
66         dismiss(animated: true)
67
68         guard let image = info[UIImagePickerControllerOriginalImage] as? UIImage else {
69             fatalError("couldn't load image from Photos")
70         }
71
72         scene.image = image
73     }
74 }
75
76 // MARK: - UINavigationControllerDelegate
77 extension ViewController: UINavigationControllerDelegate {
78 }
79
```



Open “ViewController.swift” file and import CoreML and Vision

```
import CoreML  
import Vision
```

- Core ML makes it easy to use learned models for your applications
- Vision, you can use Apple's model to detect faces, landmarks, text, squares, barcodes, and various objects.

```
30
31 import UIKit
32 import CoreML
33 import Vision
34
35
36
37 class ViewController: UIViewController {
38
39     // MARK: - IBOutlets
40     @IBOutlet weak var scene: UIImageView!
41     @IBOutlet weak var answerLabel: UILabel!
42
43     // MARK: - Properties
44     let vowels: [Character] = ["a", "e", "i", "o", "u"]
45
46     // MARK: - View Life Cycle
47     override func viewDidLoad() {
48         super.viewDidLoad()
49
50         guard let image = UIImage(named: "train_night") else {
51             fatalError("no starting image")
52         }
53
54         scene.image = image
55     }
56 }
57
58 // MARK: - IBActions
59 extension ViewController {
60
61     @IBAction func pickImage(_ sender: Any) {
62         let pickerController = UIImagePickerController()
63         pickerController.delegate = self
64         pickerController.sourceType = .savedPhotosAlbum
65         present(pickerController, animated: true)
66     }
67 }
68
69 // MARK: - UIImagePickerControllerDelegate
70 extension ViewController: UIImagePickerControllerDelegate {
71
72     func imagePickerController(_ picker: UIImagePickerController, didFinishPickingMediaWithInfo info:
```

Then add the extensions after the IBActions extension

```
// MARK: - Methods
extension ViewController {

    func detectScene(image: CIImage) {
        answerLabel.text = "detecting scene..."

        // Load the ML model through its generated class
        guard let model = try? VNCoreMLModel(for: MobileNet().model) else {
            fatalError("can't load Places ML model")
        }
    }
}
```

- Before the picture is analyzed, inform the user it's under detecting
- Create a model to use (Use try for unexpected error)
- VNCoreMLModel is a container that allows CoreML to be used forVision requests

```
class ViewController: UIViewController {

    // MARK: - IBOutlets
    @IBOutlet weak var scene: UIImageView!
    @IBOutlet weak var answerLabel: UILabel!

    // MARK: - Properties
    let vowels: [Character] = ["a", "e", "i", "o", "u"]

    // MARK: - View Life Cycle
    override func viewDidLoad() {
        super.viewDidLoad()

        guard let image = UIImage(named: "train_night") else {
            fatalError("no starting image")
        }

        scene.image = image
    }
}

// MARK: - IBActions
extension ViewController {

    @IBAction func pickImage(_ sender: Any) {
        let pickerController = UIImagePickerController()
        pickerController.delegate = self
        pickerController.sourceType = .savedPhotosAlbum
        present(pickerController, animated: true)
    }

    func detectScene(image: CIImage) {
        answerLabel.text = "detecting scene..."

        // Load the ML model through its generated class
        guard let model = try? VNCoreMLModel(for: MobileNet().model) else { ⚠️ Value 'model' was defined bu...
            fatalError("can't load Places ML model")
        }
    }
}

// MARK: - UIImagePickerControllerDelegate
extension ViewController: UIImagePickerControllerDelegate {
```

Add the following line to the last line of detectScene(image :)

```
// Create a Vision request with completion handler
let request = VNCoreMLRequest(model: model) { [weak self] request, error in
    guard let results = request.results as? [VNClassificationObservation],
          let topResult = results.first else {
        fatalError("unexpected result type from VNCoreMLRequest")
    }

    // Update UI on main queue
    let article = (self?.vowels.contains(topResult.identifier.first!)!) ? "an" : "a"
    DispatchQueue.main.async { [weak self] in
        self?.answerLabel.text = "\((Int(topResult.confidence * 100))% it's \(article) \
        \n\((topResult.identifier))"
    }
}
```

- VNCoreMLRequest is an image analysis request that performs work using Core ML model.
- Complete Handler receives request and error objects.
- “Results” determines if an array of VNClassificationObsevation objects.
- If it is set to “else”, it causes an error because the analysis failed.

Add the following line to the last line of detectScene(image :)

```
// Create a Vision request with completion handler
let request = VNCoreMLRequest(model: model) { [weak self] request, error in
    guard let results = request.results as? [VNClassificationObservation],
          let topResult = results.first else {
        fatalError("unexpected result type from VNCoreMLRequest")
    }

    // Update UI on main queue
    let article = (self?.vowels.contains(topResult.identifier.first!)!) ? "an" : "a"
    DispatchQueue.main.async { [weak self] in
        self?.answerLabel.text = "\((Int(topResult.confidence * 100))% it's \(article) \
        \n\((topResult.identifier))"
    }
}
```

- **VNClassificationObservation** has two attributes: identifier (String type) and confidence (a number between 0 and 1)
- Take the first result with the highest confidence
- It will pass the value to the main queue to update the label

```
// MARK: - IBActions
extension ViewController {

    @IBAction func pickImage(_ sender: Any) {
        let pickerController = UIImagePickerController()
        pickerController.delegate = self
        pickerController.sourceType = .savedPhotosAlbum
        present(pickerController, animated: true)
    }

    func detectScene(image: CIImage) {
        answerLabel.text = "detecting scene..."

        // Load the ML model through its generated class
        guard let model = try? VNCoreMLModel(for: MobileNet().model) else {
            fatalError("can't load Places ML model")
        }

        let request = VNCoreMLRequest(model: model) { [weak self] request, error in ⚠ Initialization of im...
            guard let results = request.results as? [VNClassificationObservation],
                  let topResult = results.first else {
                fatalError("unexpected result type from VNCoreMLRequest")
            }

            // Update UI on main queue
            let article = (self?.vowels.contains(topResult.identifier.first!)!) ? "an" : "a"
            DispatchQueue.main.async { [weak self] in
                self?.answerLabel.text = "\((Int(topResult.confidence * 100))% it's \(article) \
                    \(topResult.identifier)"
            }
        }
    }

    // MARK: - UIImagePickerControllerDelegate
    extension ViewController: UIImagePickerControllerDelegate {

        func imagePickerController(_ picker: UIImagePickerController, didFinishPickingMediaWithInfo info: [String : Any]) {
```

Add the next line to the last line of detectScene(image :)

```
// Run the Core ML MobileNet classifier on global dispatch queue
let handler = VNImageRequestHandler(ciImage: image)
DispatchQueue.global(qos: .userInteractive).async {
    do {
        try handler.perform([request])
    } catch {
        print(error)
    }
}
```

- `VNImageRequestHandler` is the standard request handler for the Vision framework.
- You give it the image that came into `detectScene(image:)` as an argument
- Then you run the handler by calling its `perform` method, passing an array of requests.
- The `perform` method throws an error, so you wrap it in a try-catch.

```
// MARK: - IBActions
extension ViewController {

    @IBAction func pickImage(_ sender: Any) {
        let pickerController = UIImagePickerController()
        pickerController.delegate = self
        pickerController.sourceType = .savedPhotosAlbum
        present(pickerController, animated: true)
    }

    func detectScene(image: CIImage) {
        answerLabel.text = "detecting scene..."

        // Load the ML model through its generated class
        guard let model = try? VNCoreMLModel(for: MobileNet().model) else {
            fatalError("can't load Places ML model")
        }

        let request = VNCoreMLRequest(model: model) { [weak self] request, error in
            guard let results = request.results as? [VNClassificationObservation],
                  let topResult = results.first else {
                fatalError("unexpected result type from VNCoreMLRequest")
            }

            // Update UI on main queue
            let article = (self?.vowels.contains(topResult.identifier.first!)!)! ? "an" : "a"
            DispatchQueue.main.async { [weak self] in
                self?.answerLabel.text = "\((Int(topResult.confidence * 100))% it's \(article) \
                    \(topResult.identifier)"
            }
        }
    }

    // Run the Core ML GoogLeNetPlaces classifier on global dispatch queue
    let handler = VNImageRequestHandler(ciImage: image)
    DispatchQueue.global(qos: .userInteractive).async {
        do {
            try handler.perform([request])
        } catch {
            print(error)
        }
    }
}
}
```

Add the following lines at the end of viewDidLoad() and at the end of imagePickerController(_:didFinishPickingMediaWithInfo:):

```
guard let ciImage = CIImage(image: image) else {  
    fatalError("couldn't convert UIImage to CIImage")  
}  
  
detectScene(image: ciImage)
```

- Using the model for image analysis
- The model is executed after the initial image is loaded, or after a new image has been uploaded

```
39 // MARK: - IBOutlets
40 @IBOutlet weak var scene: UIImageView!
41 @IBOutlet weak var answerLabel: UILabel!
42
43 // MARK: - Properties
44 let vowels: [Character] = ["a", "e", "i", "o", "u"]
45
46 // MARK: - View Life Cycle
47 override func viewDidLoad() {
48     super.viewDidLoad()
49
50     guard let image = UIImage(named: "train_night") else {
51         fatalError("no starting image")
52     }
53
54     scene.image = image
55
56     guard let ciImage = CIImage(image: image) else {
57         fatalError("couldn't convert UIImage to CIImage")
58     }
59
60     detectScene(image: ciImage)
61 }
62 }
```

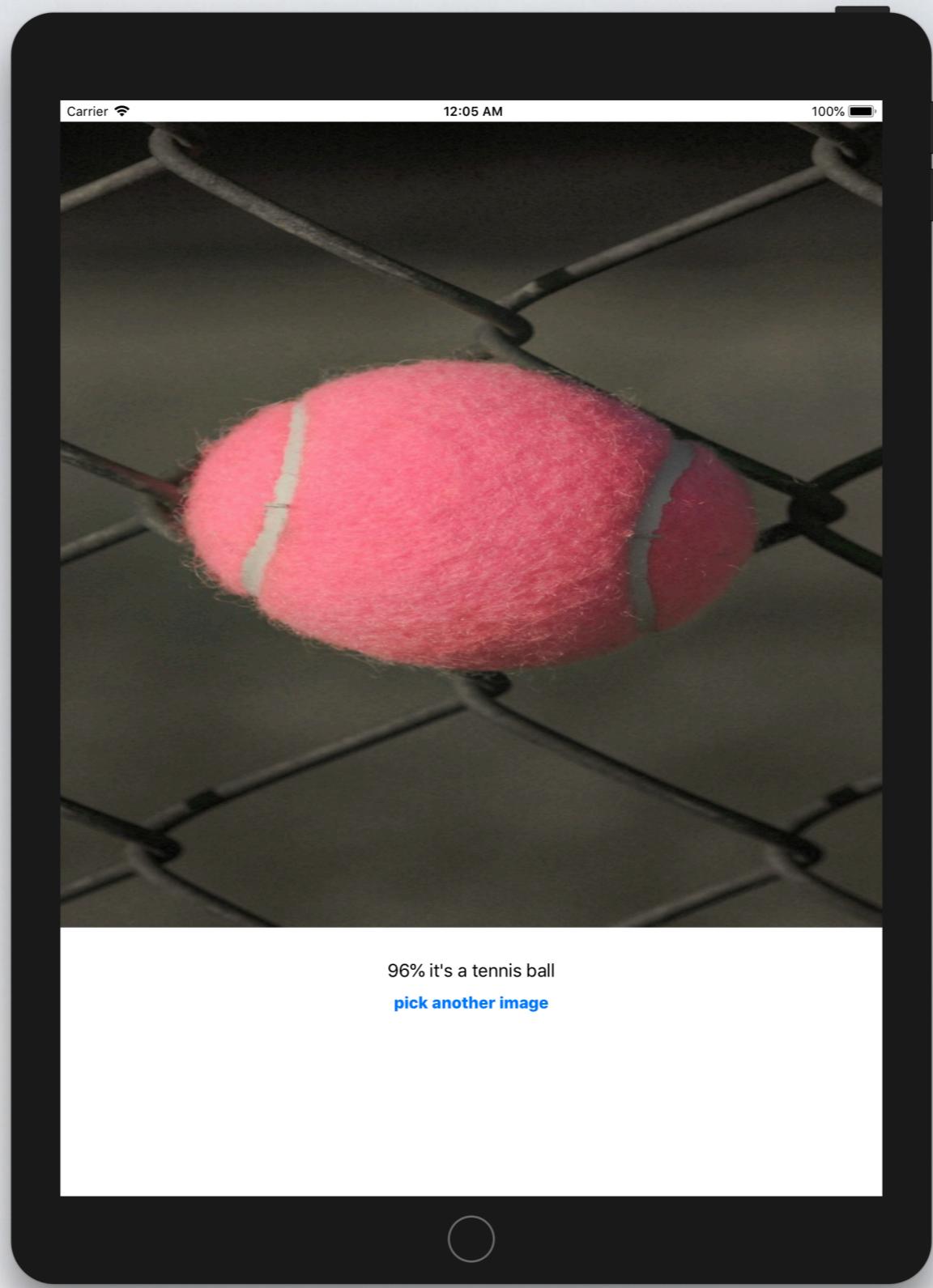
```
// MARK: - UIImagePickerControllerDelegate
extension ViewController: UIImagePickerControllerDelegate {

    func imagePickerController(_ picker: UIImagePickerController, didFinishPickingMediaWithInfo info: [String : Any]) {
        dismiss(animated: true)

        guard let image = info[UIImagePickerControllerOriginalImage] as? UIImage else {
            fatalError("couldn't load image from Photos")
        }

        scene.image = image
        guard let ciImage = CIImage(image: image) else {
            fatalError("couldn't convert UIImage to CIImage")
        }

        detectScene(image: ciImage)
    }
}
```



iPad (5th generation) - 11.3

Now build and run

REFERENCE

- <https://www.raywenderlich.com/164213/coreml-and-vision-machine-learning-in-ios-11-tutorial>
- <https://developer.apple.com/kr/machine-learning/>