What is The Test

Feat. UniTest, UITest

발표자. 마차챠 팀의 김건형

Test?

개발자가 작성한 코드가 의도한대로 작동하는지 확인하는 과정

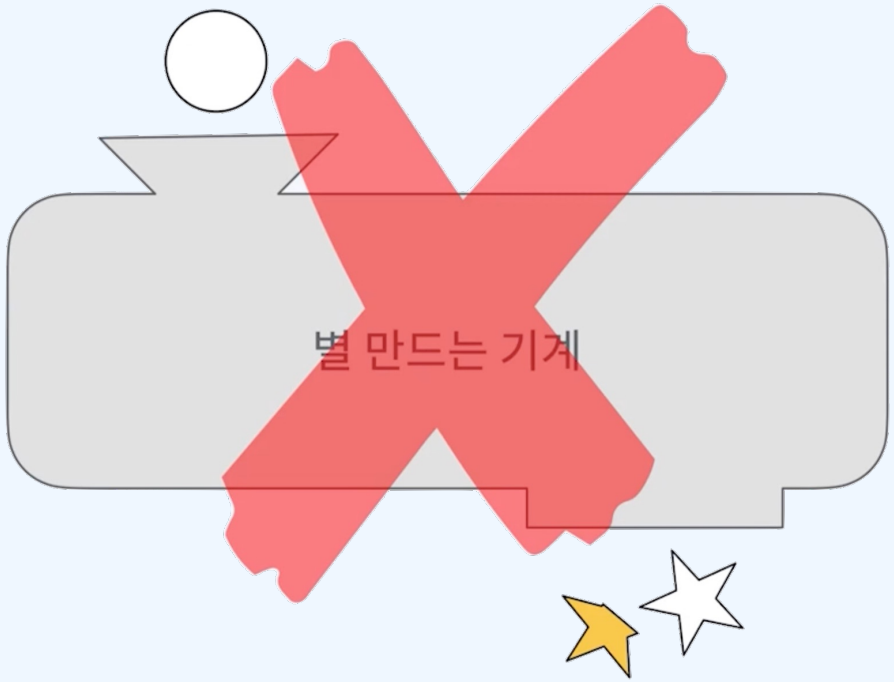
Test?

개발자가 작성한 코드가 의도한대로 작동하는지 확인하는 과정

- UI 표시
- UI Layout 설정
- API request
- UserDefaults 저장하기/ 가져오기
- 등등

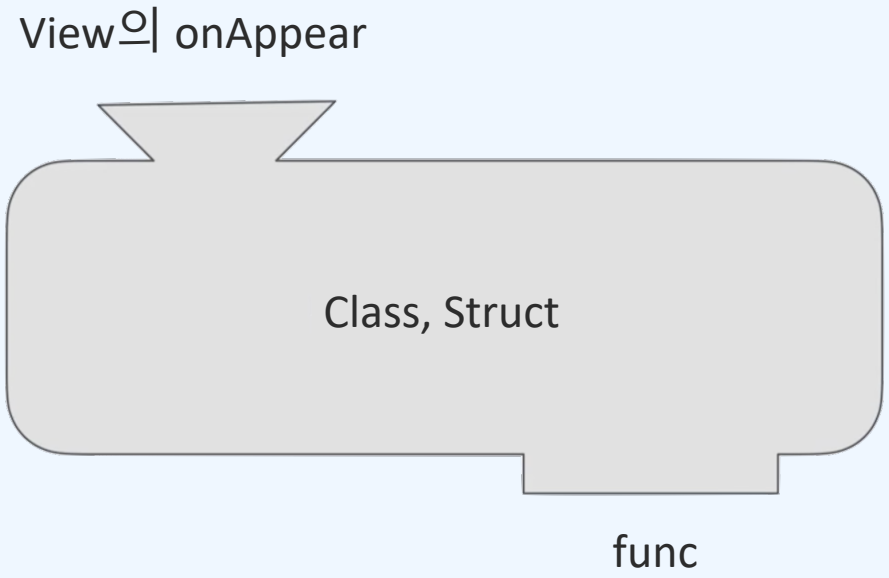
Test?

Test?

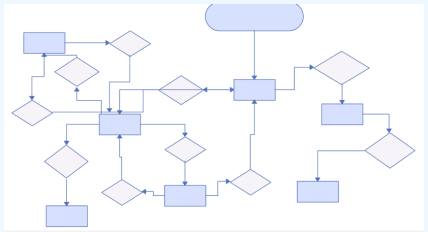


Test?

Test?



Test 종류



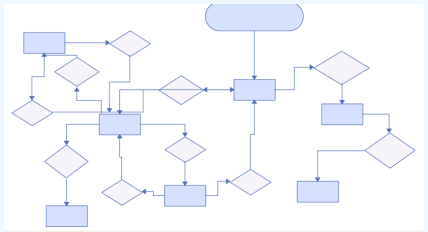
Unit Test

특정 함수, 메소드의 동작에 대한 테스트

UI Test

UI 표시 or UI Action에 대한 테스트

Test 종류



Unit Test

특정 함수, 메소드의 동작에 대한 테스트

UI Test

UI 표시 or UI Action에 대한 테스트

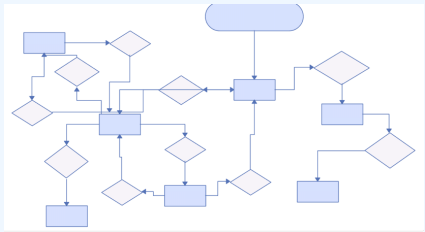
Scene을 테스트 대상으로 실행

Class ProfileViewModel: XCTest {}

하나의 앱을 테스트 대상으로 실행

XCUIApplication().launch()

Test 종류



Unit Test

특정 함수, 메소드의 동작에 대한 테스트

UI Test

UI 표시 or UI Action에 대한 테스트

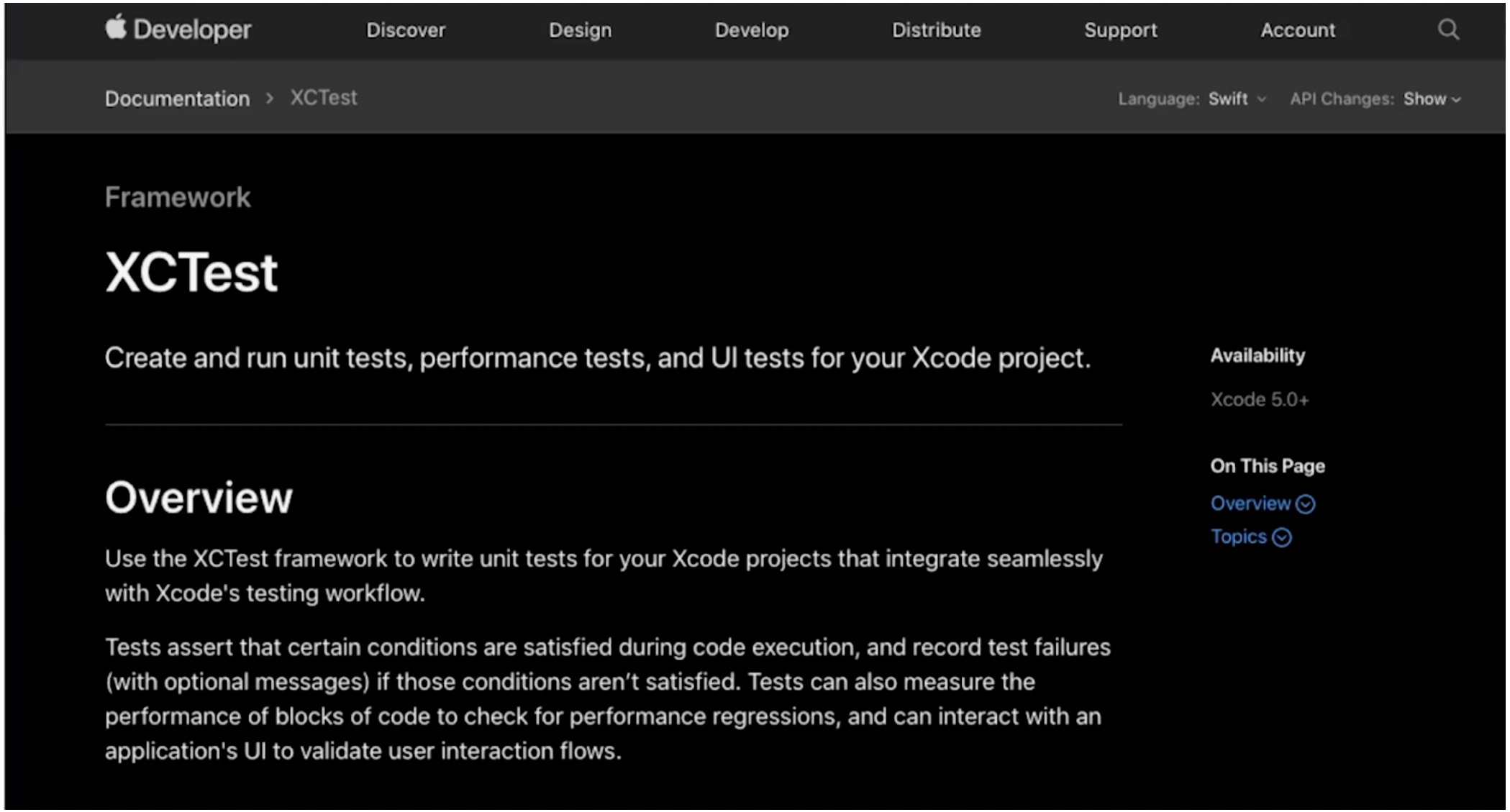
Scene을 테스트 대상으로 실행

Class ProfileViewModel: XCTest {}

하나의 앱을 테스트 대상으로 실행

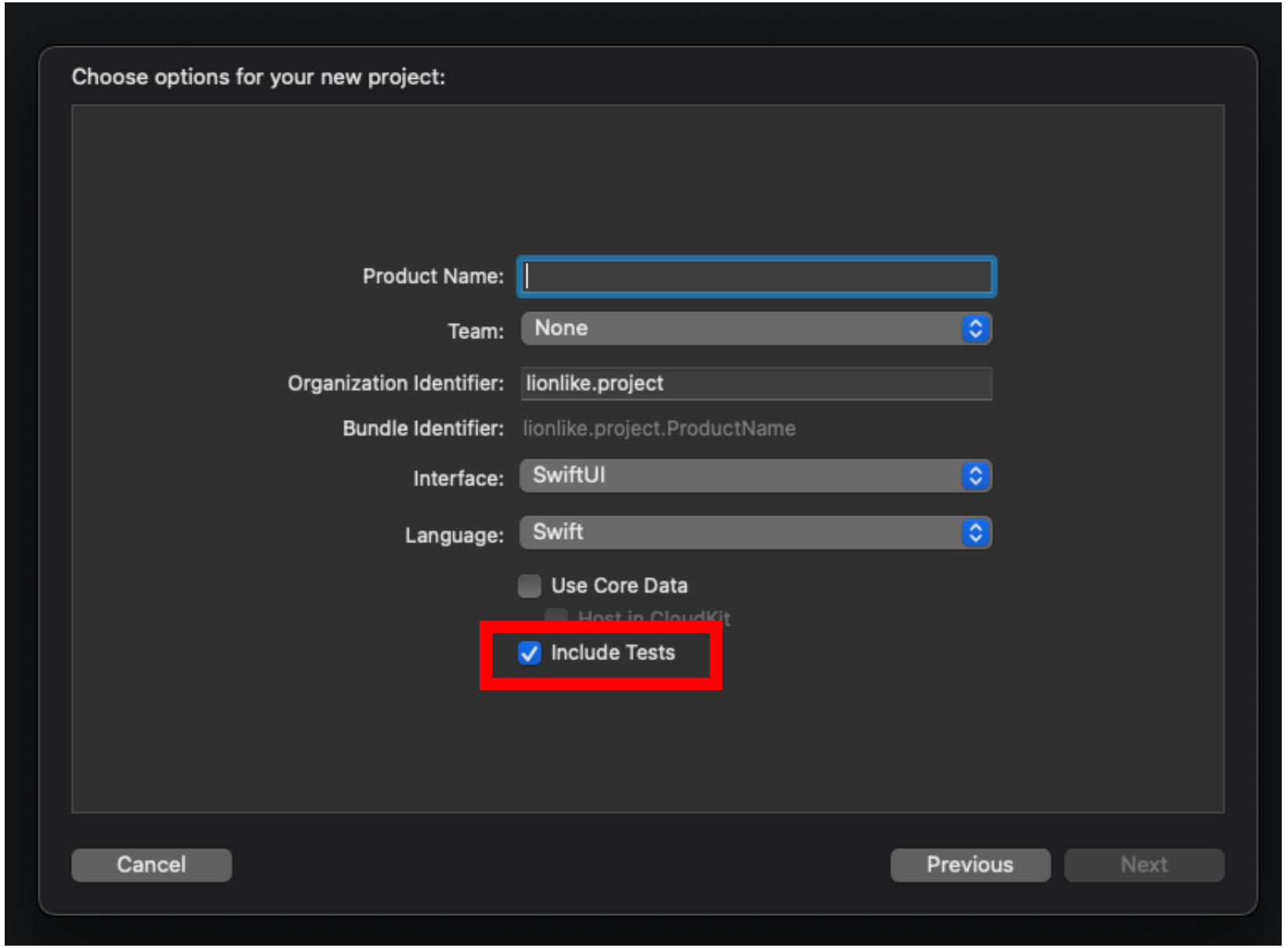
XCUIApplication().launch()

XCTest > UniTest

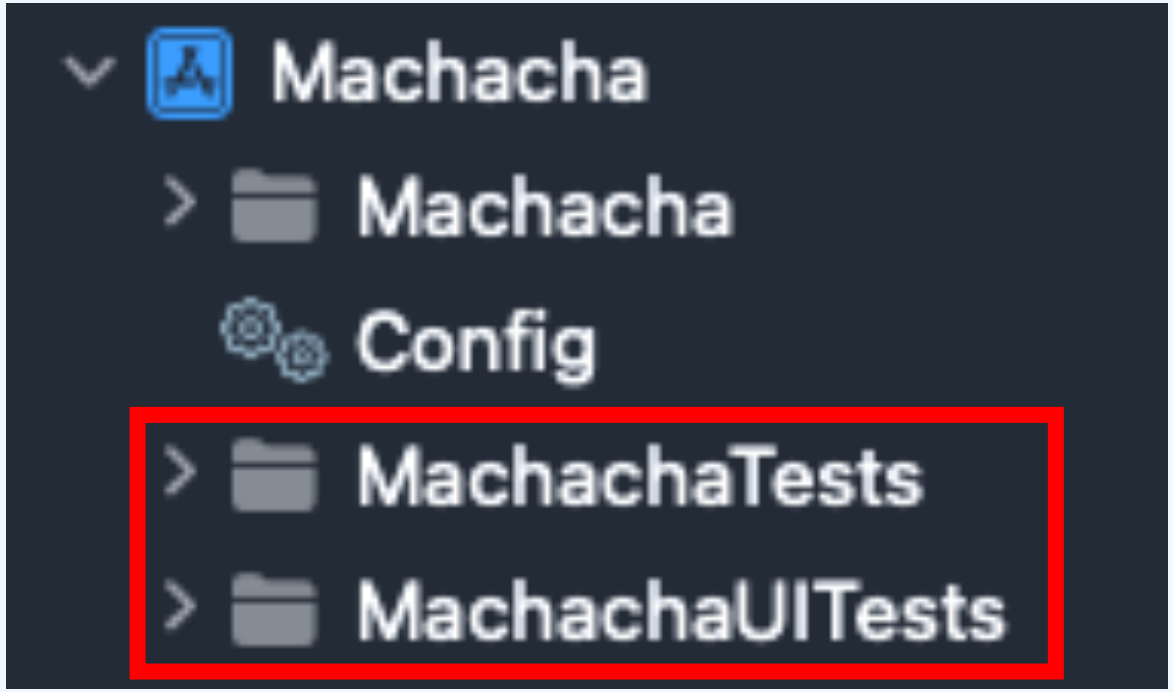


https://developer.apple.com/documentation/xctest/defining_test_cases_and_test_methods

XCTest > UniTest



XCTest > UniTest



XCTest > UniTest

```swift
//
//  MachachaTests.swift
//  MachachaTests
//
//  Created by geonhyeong on 2023/01/17.
//

import XCTest
@testable import Machacha

final class MachachaTests: XCTestCase {

    override func setUpWithError() throws {
        // Put setup code here. This method is called before the invocation of each test method in the class.
    }

    override func tearDownWithError() throws {
        // Put teardown code here. This method is called after the invocation of each test method in the class.
    }

    func testExample() throws {
        // This is an example of a functional test case.
        // Use XCTAssert and related functions to verify your tests produce the correct results.
        // Any test you write for XCTest can be annotated as throws and async.
        // Mark your test throws to produce an unexpected failure when your test encounters an uncaught error.
        // Mark your test async to allow awaiting for asynchronous code to complete. Check the results with assertions afterwards.
    }

    func testPerformanceExample() throws {
        // This is an example of a performance test case.
        self.measure {
            // Put the code you want to measure the time of here.
        }
    }

}
```

XCTest > UniTest

```swift
//
//  MachachaTests.swift
//  MachachaTests
//
//  Created by geonhyeong on 2023/01/17.
//

import XCTest
@testable import Machacha

final class MachachaTests: XCTestCase {

    override func setUpWithError() throws {
        // Put setup code here. This method is called before the invocation of each test method in the class.
    }

    override func tearDownWithError() throws {
        // Put teardown code here. This method is called after the invocation of each test method in the class.
    }

    func testExample() throws {
        // This is an example of a functional test case.
        // Use XCTAssert and related functions to verify your tests produce the correct results.
        // Any test you write for XCTest can be annotated as throws and async.
        // Mark your test throws to produce an unexpected failure when your test encounters an uncaught error.
        // Mark your test async to allow awaiting for asynchronous code to complete. Check the results with assertions afterwards.
    }

    func testPerformanceExample() throws {
        // This is an example of a performance test case.
        self.measure {
            // Put the code you want to measure the time of here.
        }
    }

}
```


```swift
import XCTest
@testable import Machacha
```

XCTest > UniTest

```swift
//
//  MachachaTests.swift
//  MachachaTests
//
//  Created by geonhyeong on 2023/01/17.
//

import XCTest
@testable import Machacha

final class MachachaTests: XCTestCase {

    override func setUpWithError() throws {
        // Put setup code here. This method is called before the invocation of each test method in the class.
    }

    override func tearDownWithErr
        // Put teardown code here
    }

    func testExample() throws {
        // This is an example of a functional test case.
        // Use XCTAssert and related functions to verify your tests produce the correct results.
        // Any test you write for XCTest can be annotated as throws and async.
        // Mark your test throws to produce an unexpected failure when your test encounters an uncaught error.
        // Mark your test async to allow awaiting for asynchronous code to complete. Check the results with assertions afterwards.
    }

    func testPerformanceExample() throws {
        // This is an example of a performance test case.
        self.measure {
            // Put the code you want to measure the time of here.
        }
    }

}
```

```swift
final class MachachaTests: XCTestCase {
```

Class ProfileViewModelTest

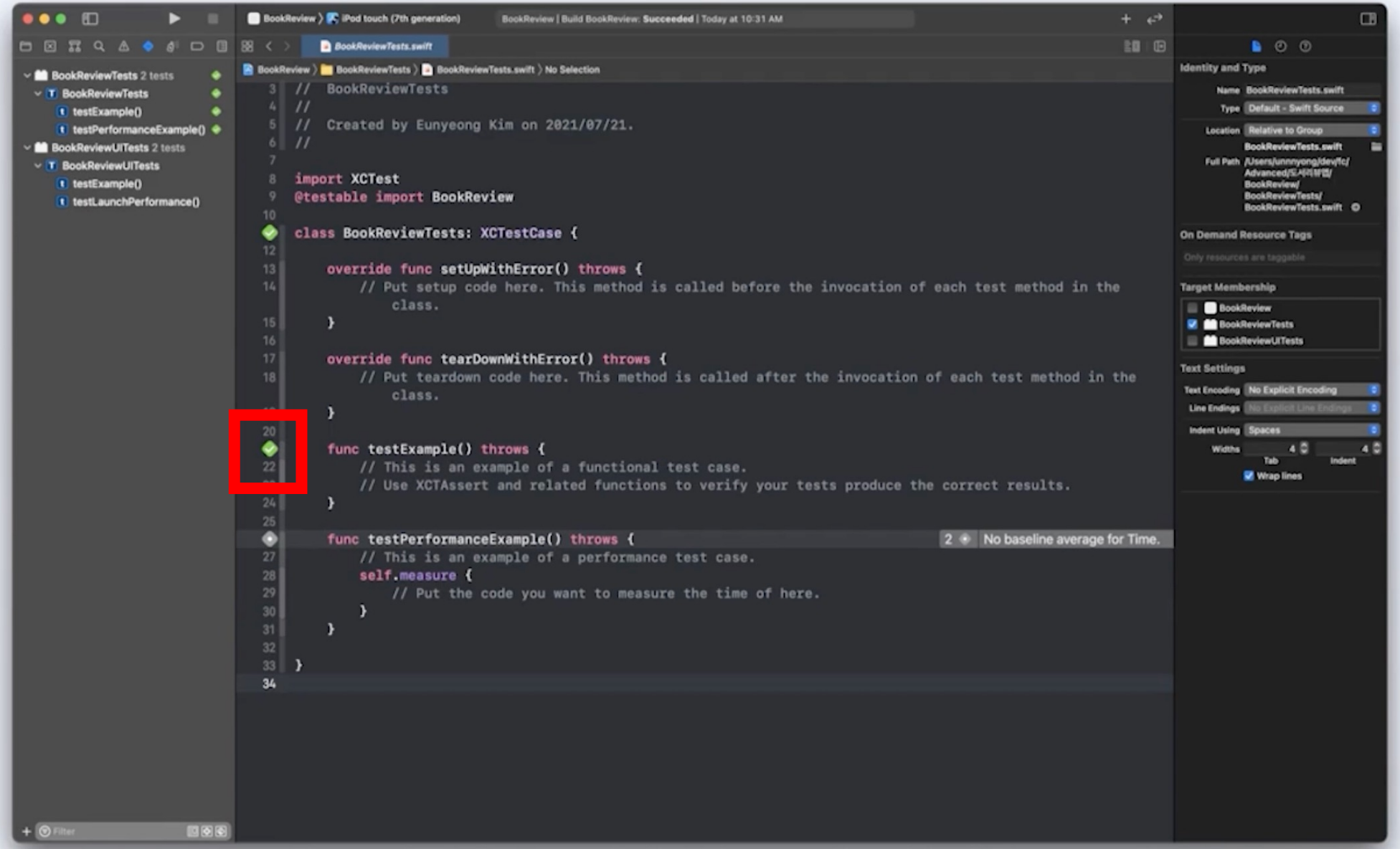
XCTest > UniTest

```swift
//
//  MachachaTests.swift
//  MachachaTests
//
//  Created by geonhyeong on 2023/01/17.
//

import XCTest
@testable import Machacha

final class MachachaTests: XCTestCase {

    override func setUpWithError() throws {
        // Put setup code here. This method is called before the invocation of each test method in the class.
    }

    override func tearDownWithError() throws {
        // Put teardown code here. This method is called after the invocation of each test method in the class.
    }

    func testExample() throws {
        // This is an example of a performance test case.
        self.measure {
            // Put the code you want to measure the time of here.
        }
    }

}
```

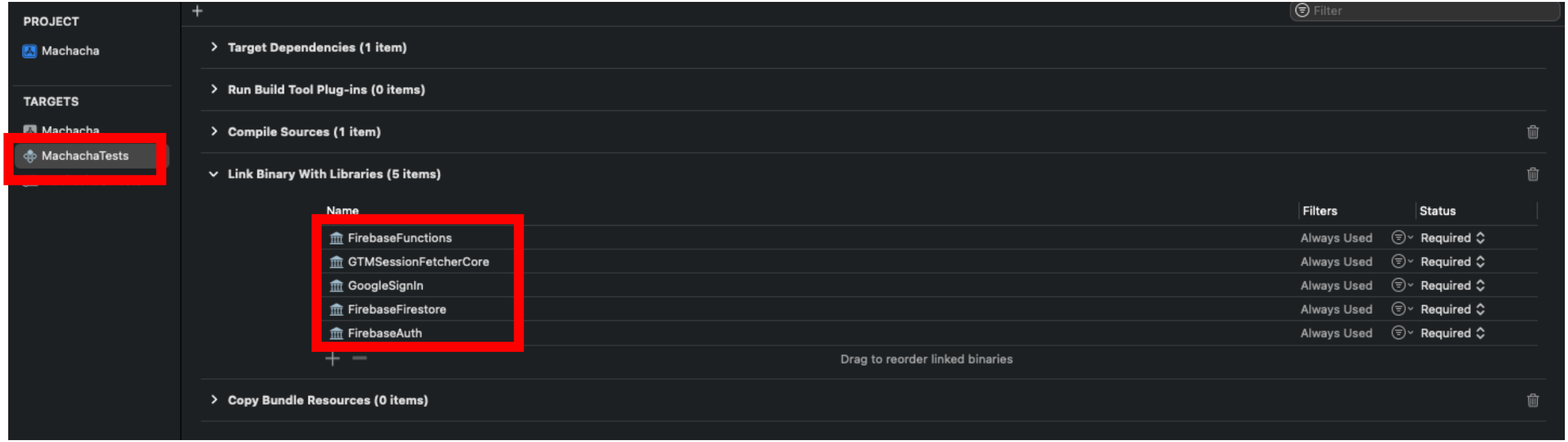
XCTest > UniTest

```swift
//
//  MachachaTests.swift
//  MachachaTests
//
//  Created by geonhyeong on 2023/01/17.
//

func testExample() throws {
    // This is an example of a functional test case.
    // Use XCTAssert and related functions to verify your tests produce the correct results.
    // Any test you write for XCTest can be annotated as throws and async.
    // Mark your test throws to produce an unexpected failure when your test encounters an uncaught error.
    // Mark your test async to allow awaiting for asynchronous code to complete. Check the results with assertions afterwards.
}

        // Put teardown code here. This method is called after the invocation of each test method in the class.
    }

    func testExample() throws {
        // This is an example of a functional test case.
        // Use XCTAssert and related functions to verify your tests produce the correct results.
        // Any test you write for XCTest can be annotated as throws and async.
        // Mark your test throws to produce an unexpected failure when your test encounters an uncaught error.
        // Mark your test async to allow awaiting for asynchronous code to complete. Check the results with assertions afterwards.
    }

    func testPerformanceExample() throws {
        // This is an example of a performance test case.
        self.measure {
            // Put the code you want to measure the time of here.
        }
    }

}
```

XCTest > UniTest

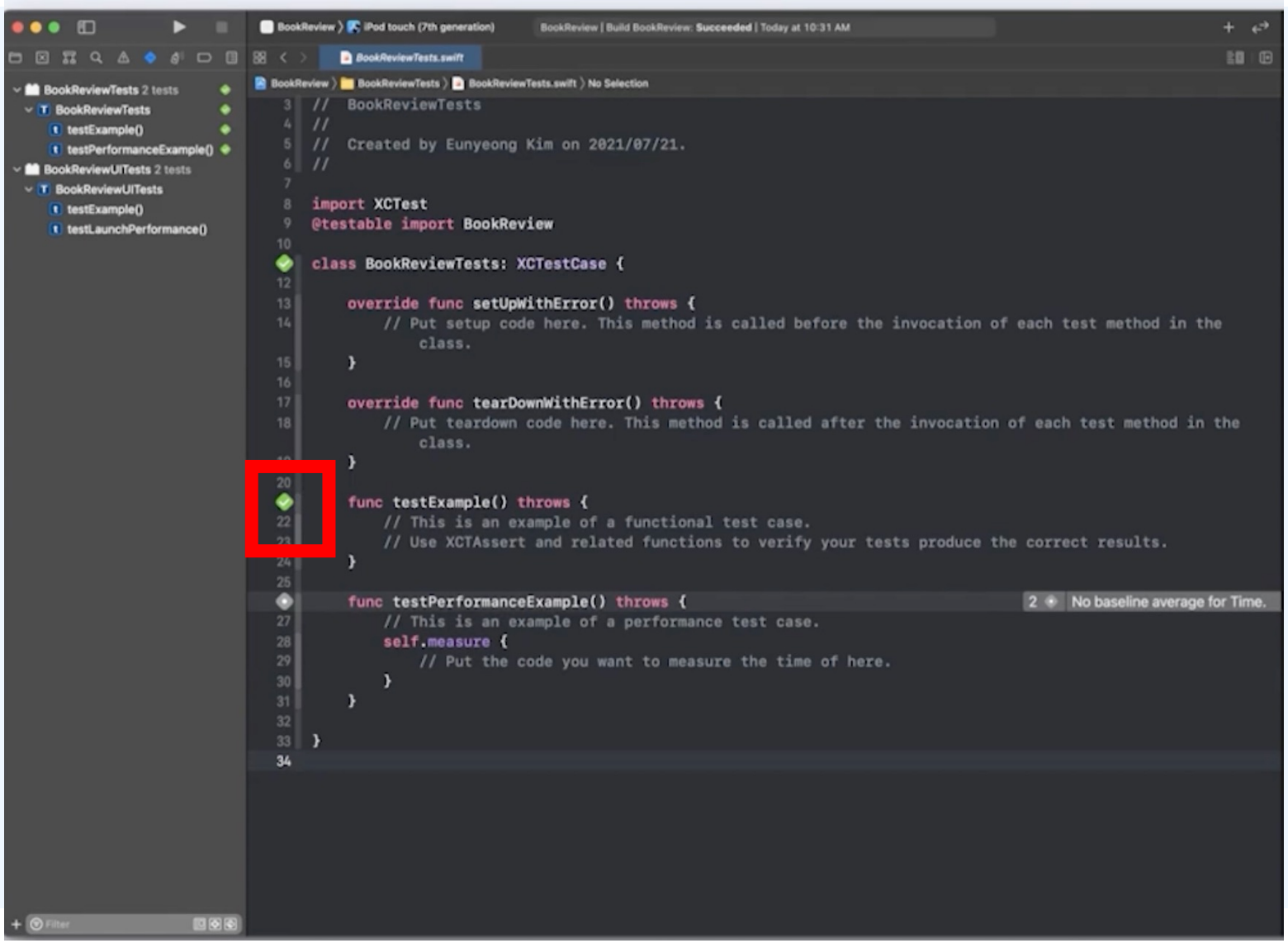


XCTest > UniTest

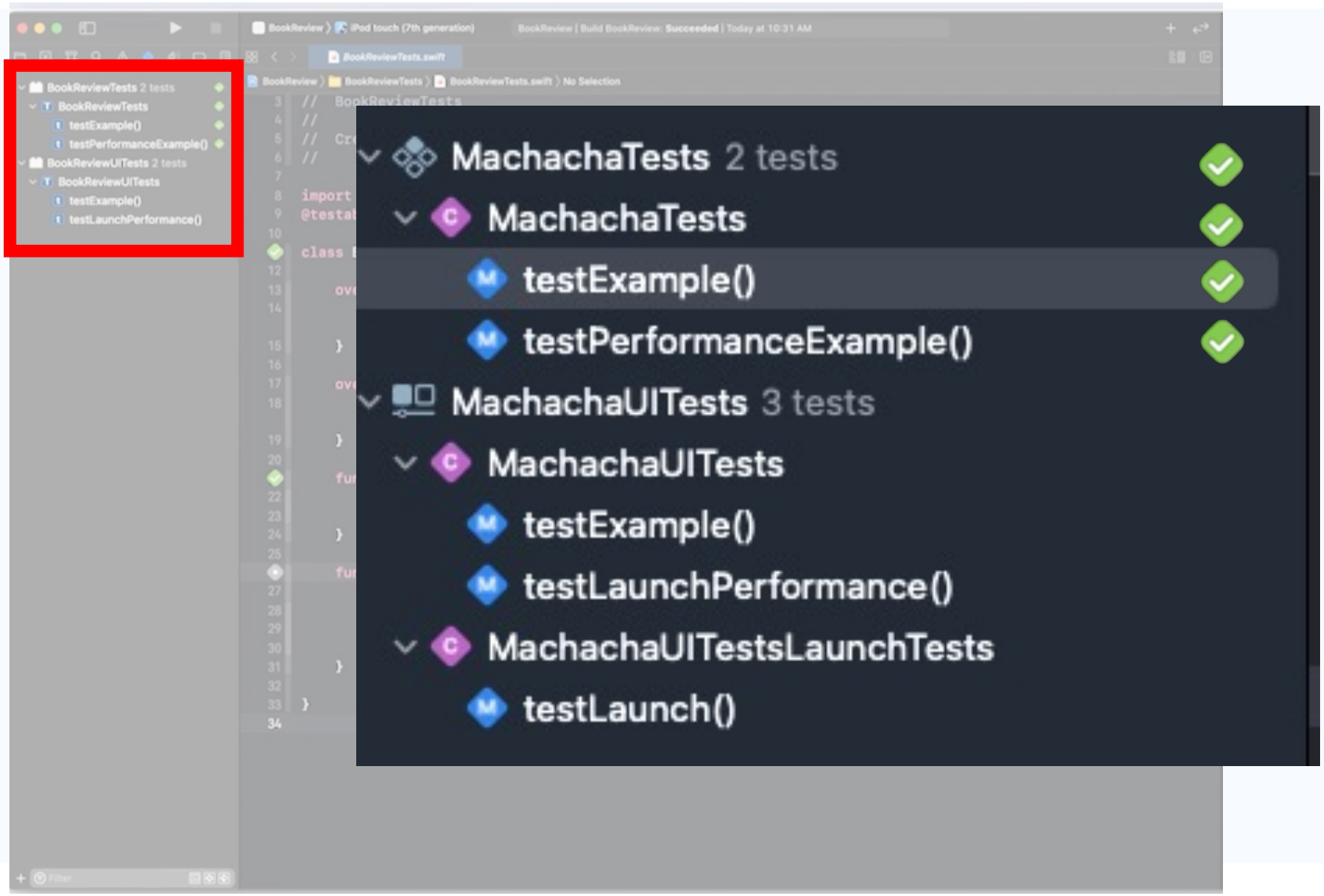


GTMSessionFetcherCore: https://github.com/google/gtm-session-fetcher

XCTest > UniTest



XCTest > UniTest



```swift
import XCTest
@testable import Machacha

final class ProfileViewTests: XCTestCase {
    var sut: ProfileView! // 테스트 대상(일반적으로 sut이라 불린다), 테스트이기 때문에 ?보다는 !로 빠르게 버그를 찾는게 좋다
    var viewModel: ProfileViewModel!

    // 테스트가 실행될때마다 실행됨
    override func setUpWithError() throws {
        sut = ProfileView()
        viewModel = ProfileViewModel()
    }

    // 테스트 메소드들이 끝날때마다 실행
    override func tearDownWithError() throws {
        sut = nil
        viewModel = nil
    }

    // 임의의 test func을 작성
    func testExample() {
        let view = sut.environmentObject(viewModel)
        let body = view.body
        XCTAssertNil(body, "body가 존재하지 않음")
    }

    // 임의의 test func을 작성
    func ProfileViewModel의_fetchImage함수_Test() async {
        do {
            let result = try await viewModel.fetchImage(foodCartId: "InzqNwgl15TytWNOdIZz", imageName: "text.jpg")
            XCTAssertNotNil(result)
        } catch {
            XCTFail("이미지 불러오면서 에러발생, \(error)")
        }
    }
}
```

⚠ Will never be executed

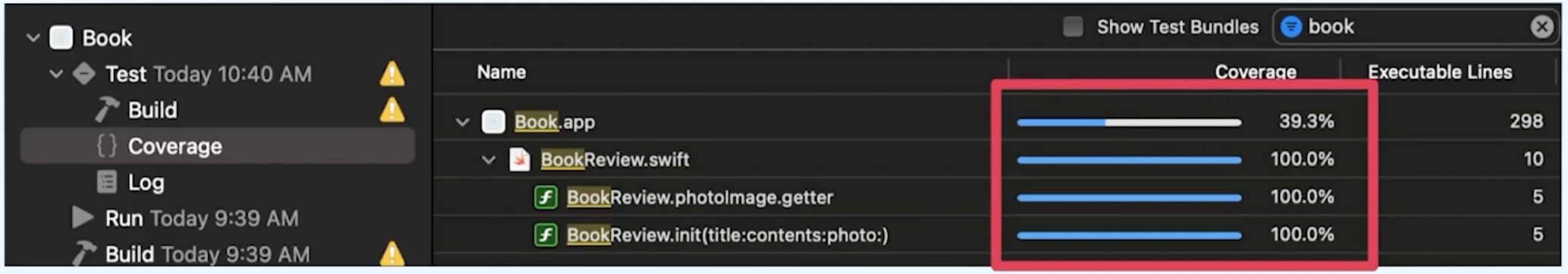
Test Coverage?

Xcode Project에서 몇 %의 코드에 대해서 테스트가 작성되어 있는지 나타낸 숫자

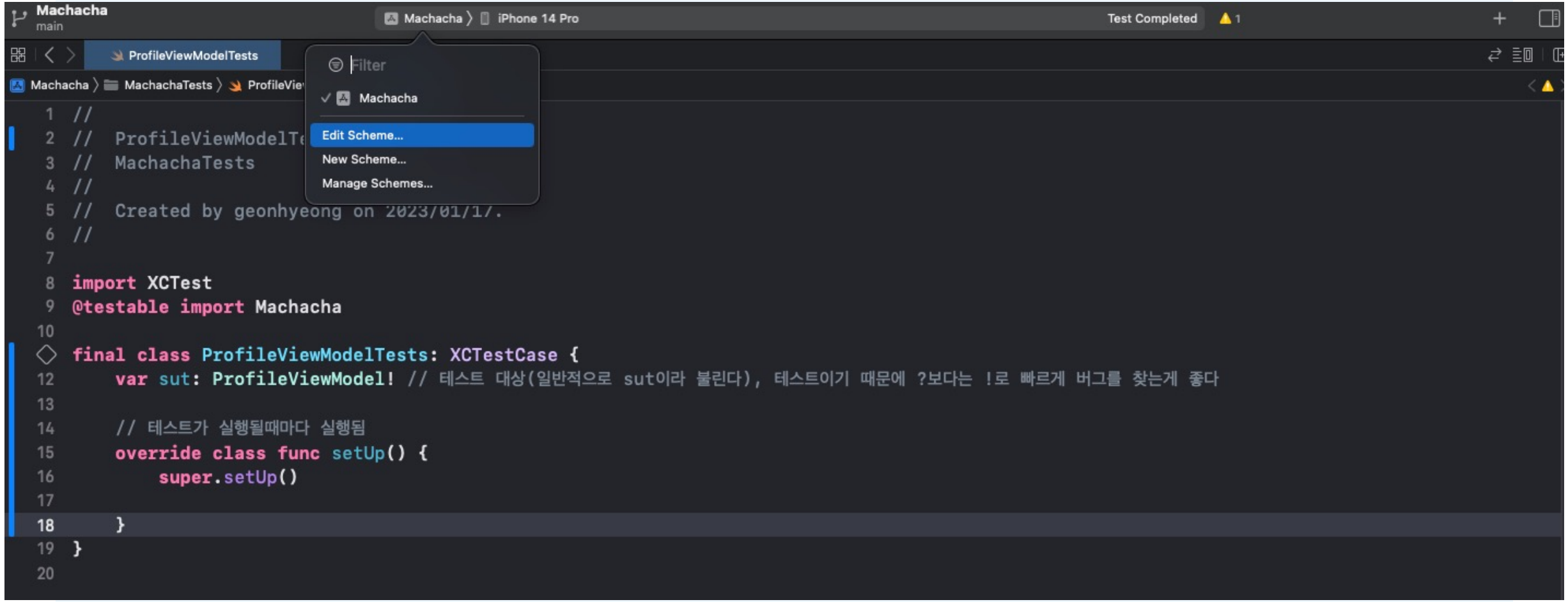
Test Coverage의 사용

App의 안정성을 확인하는 기준

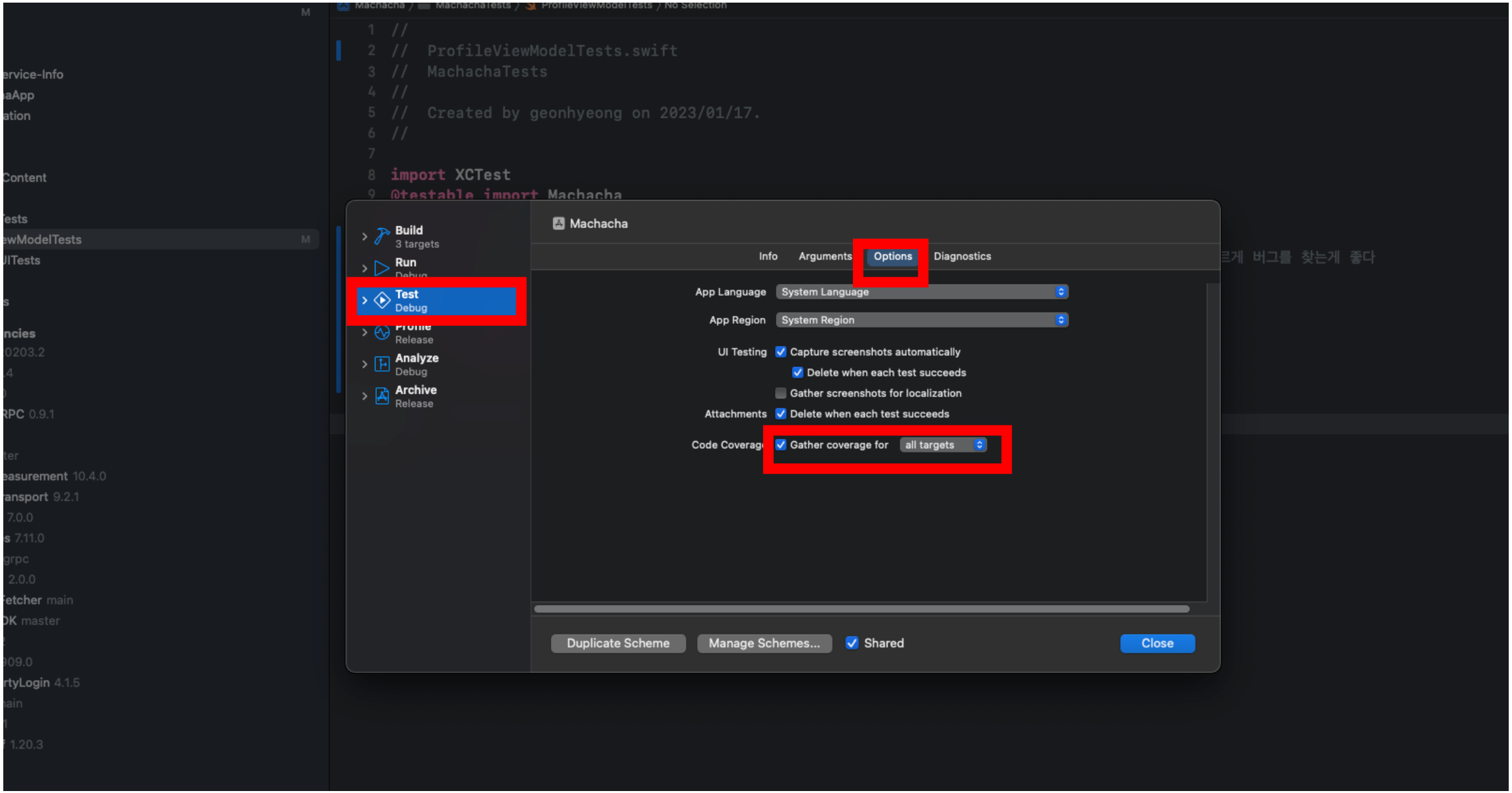
XCTest > UniTest



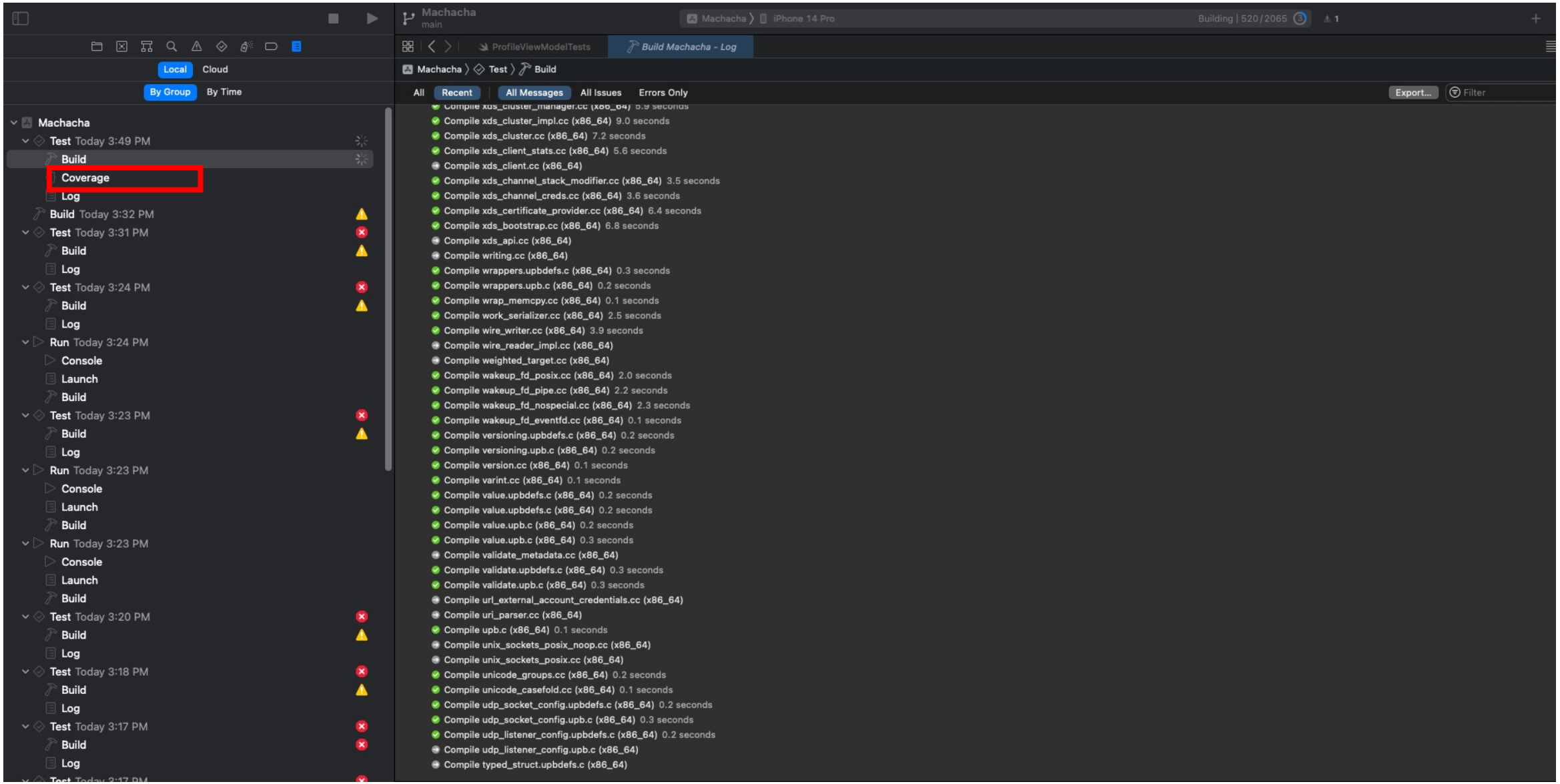
XCTest > UniTest



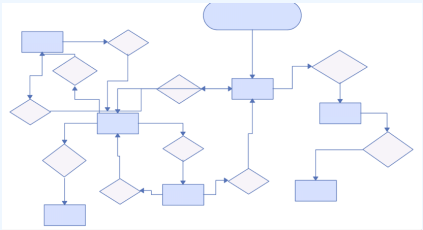
XCTest > UniTest



XCTest > UniTest



XCTest > UI Test



Unit Test

특정 함수, 메소드의 동작에 대한 테스트

UI Test

UI 표시 or UI Action에 대한 테스트

Class ProfileViewModel: XCTest {}

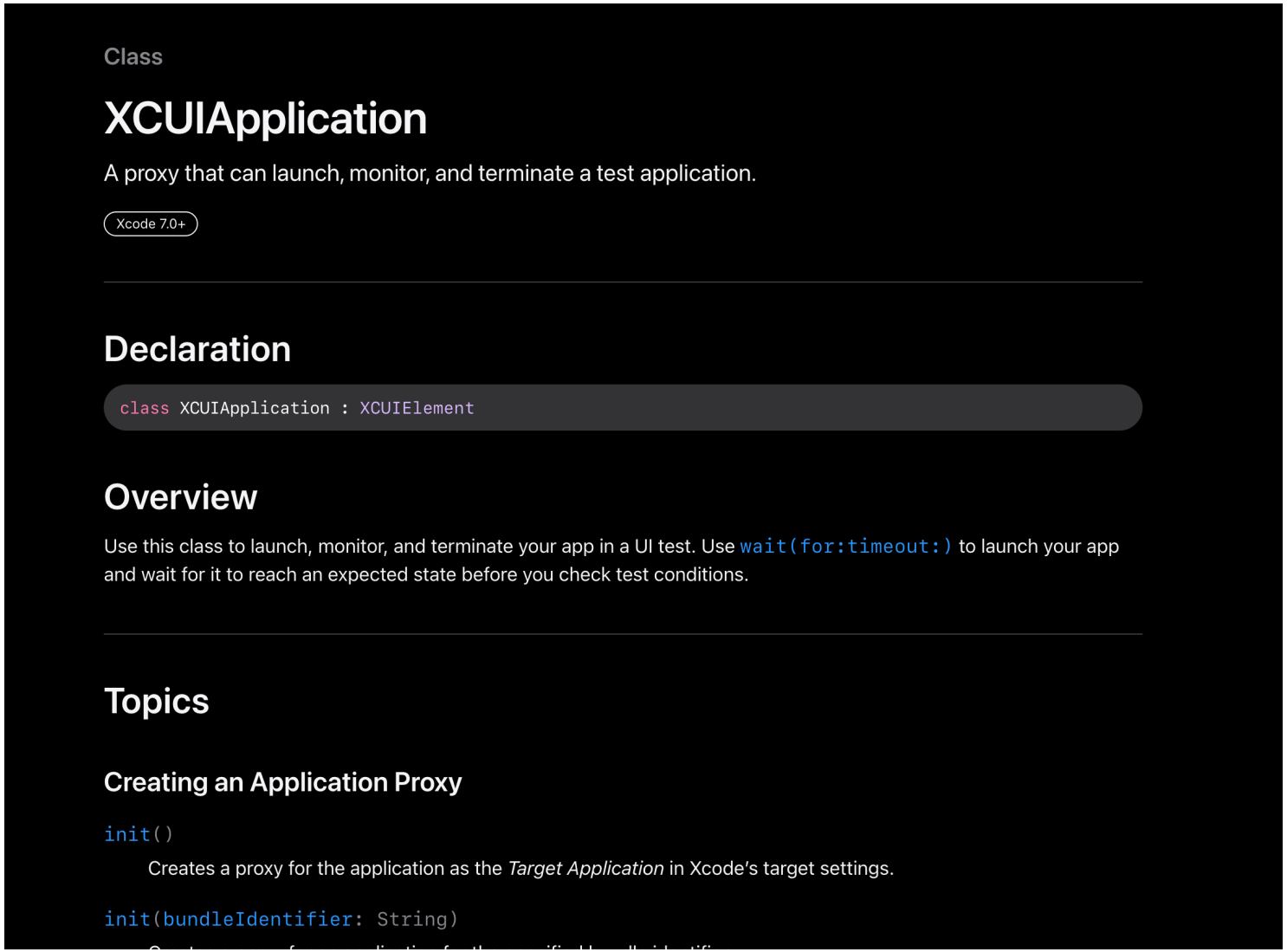
XCUIApplication().launch()

UI Test

UI Component의 표시와 동작이
의도한 대로 잘 작동되는지 확인하는 Test

- XCUIApplication().launch()

XCTest > UI Test



Class

XCUIApplication

A proxy that can launch, monitor, and terminate a test application.

Xcode 7.0+

Declaration

```
class XCUIApplication : XCUIElement
```

Overview

Use this class to launch, monitor, and terminate your app in a UI test. Use `wait(for:timeout:)` to launch your app and wait for it to reach an expected state before you check test conditions.

Topics

Creating an Application Proxy

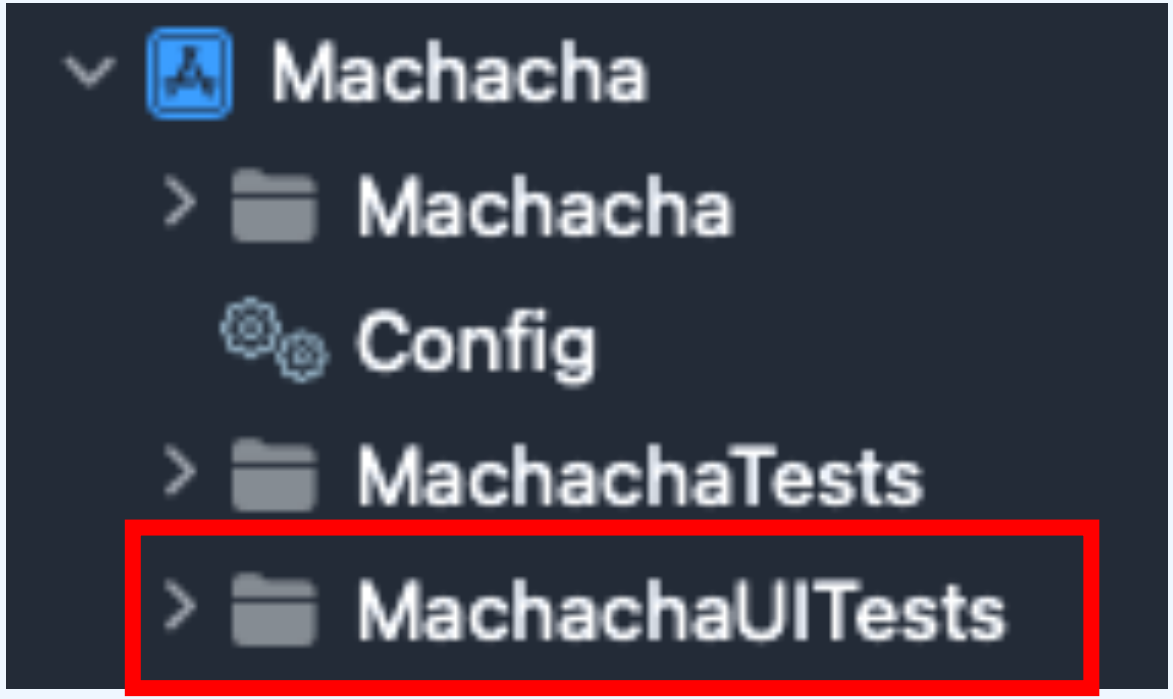
`init()`

Creates a proxy for the application as the *Target Application* in Xcode's target settings.

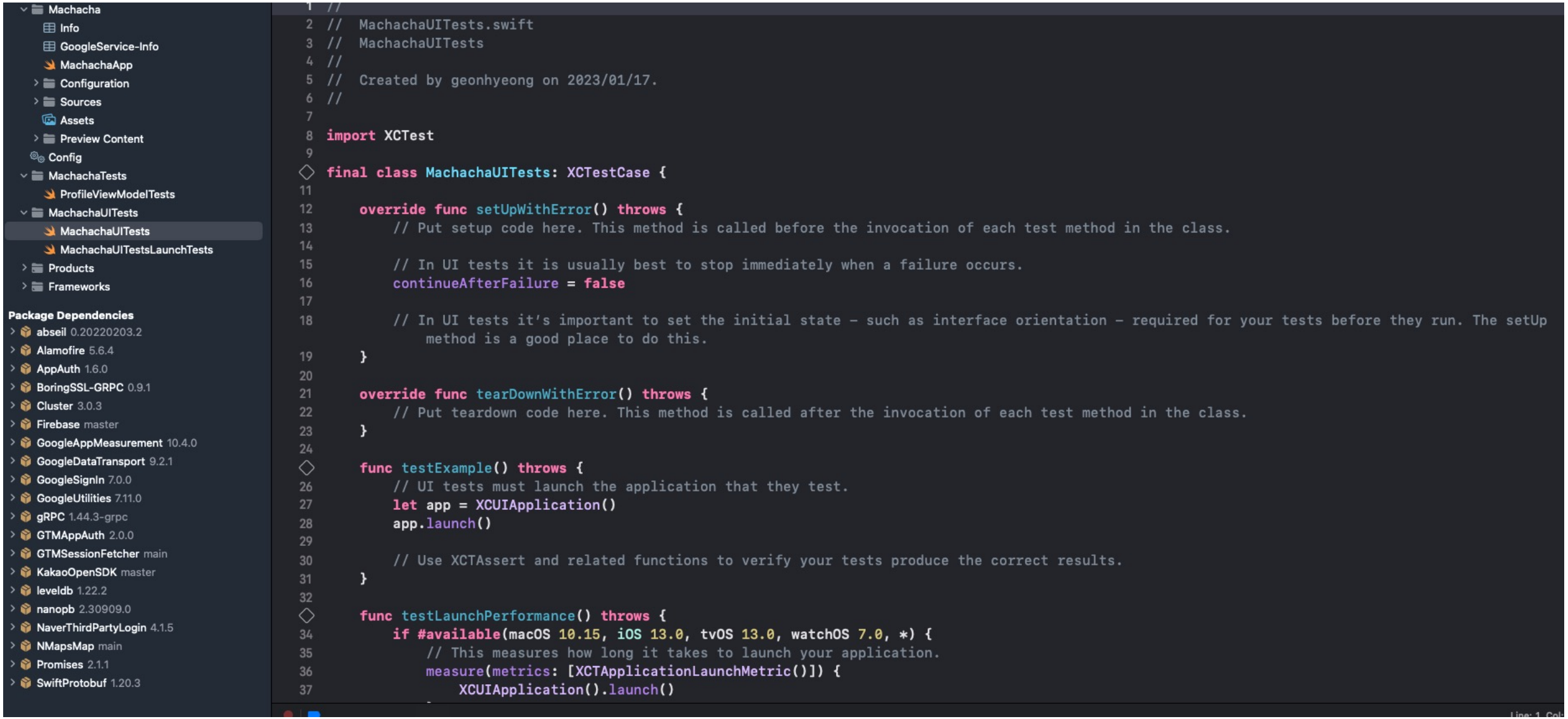
`init(bundleIdentifier: String)`

https://developer.apple.com/documentation/xctest/xcuiapplication

XCTest > UI Test



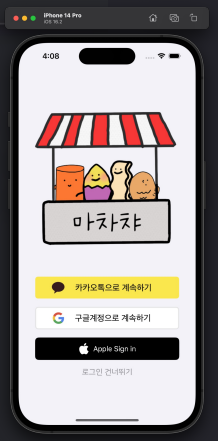
XCTest > UI Test



```swift
import XCTest

final class MachachaUITests: XCTestCase {
    var app: XCUIApplication!            // 1. UniTest에서 sut으로 표시되었던 테스트 대상

    // 테스트가 실행될때마다 실행됨
    override func setUp() {
        super.setUp()

        continueAfterFailure = false     // 2. 실패해도 계속 동작할 것인지? false(하나라도 실패가 생기면 종료)

        app = XCUIApplication()          // 3. XCUIApplication을 대입후,
        app.launch()                     // 4. 앱을 런치 시켜준다
    }

    override func tearDown() {
        super.tearDown()

        app = nil                        // 5. 앱을 nil로 초기화
    }

    // 6. test할 화면을 함수로 만들기
    func test_navigation의_Title이_즐겨찾기로_설정되어있다() {
        let existNavigationBar = app.navigationBars["즐겨찾기"].exists // 7. NavigationBar의 제목이 즐겨찾기가 존재하는지 확인
        XCTAssertTrue(existNavigationBar)
    }
}

```

```swift
final class MachachaUITests: XCTestCase {
    override func setUp() {

        app = XCUIApplication()          // 3. XCUIApplication을 대입후,
        app.launch()                     // 4. 앱을 런치 시켜준다
    }

    override func tearDown() {
        super.tearDown()

        app = nil                        // 5. 앱을 nil로 초기화
    }

    // 6. test할 화면을 함수로 만들기
    func test_navigation의_Title이_즐겨찾기로_설정되어있다() {
        let existNavigationBar = app.navigationBars["즐겨찾기"].exists // 7. NavigationBar의 제목이 즐겨찾기가 존재하는지 확인
        XCTAssertTrue(existNavigationBar)
    }
}
```

test_navigation의_Title이_즐겨찾기로_설정되어있다(): XCTAssertTrue failed



4 characters

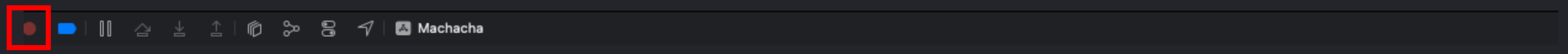
```
2023-02-02 16:58:55.095040+0900 MachachaUITests-Runner[44693:29165669] [SceneConfiguration] Info.plist contained no UIScene configuration dictionary (looking
    for configuration named "(no name)")
2023-02-02 16:58:55.103740+0900 MachachaUITests-Runner[44693:29165669] Running tests...
Test Suite 'MachachaUITests' started at 2023-02-02 16:59:01.459
Test Case '-[MachachaUITests.MachachaUITests test_navigation의_Title이_즐겨찾기로_설정되어있다]' started.
    t =     0.00s Start Test at 2023-02-02 16:59:01.463
    t =     0.18s Set Up
    t =     0.19s     Open lionlike.project.Machacha
    t =     0.26s         Launch lionlike.project.Machacha
    t =    10.61s             Setting up automation session
    t =    26.68s             Wait for lionlike.project.Machacha to idle
    t =    28.02s Checking existence of `"즐겨찾기" NavigationBar`
/Users/geonhyeong/Desktop/finalproject-machacha/src/frontend/user/MachachaUITests/MachachaUITests.swift:32: error: -[MachachaUITests.MachachaUITests
    test_navigation의_Title이_즐겨찾기로_설정되어있다] : XCTAssertTrue failed
    t =    28.76s Tear Down
Test Case '-[MachachaUITests.MachachaUITests test_navigation의_Title이_즐겨찾기로_설정되어있다]' failed (28.971 seconds).
Test Suite 'MachachaUITests' failed at 2023-02-02 16:59:30.433.
     Executed 1 test, with 1 failure (0 unexpected) in 28.971 (28.974) seconds
```

Machacha 〉 MachachaUITests 〉 MachachaUITests 〉 test_test()

```
final class MachachaUITests: XCTestCase {
    override func tearDown() {
        app = nil                        // 5. 앱을 nil로 초기화
    }

    // 6. test할 화면을 함수로 만들기
//  func test_navigation의_Title이_즐겨찾기로_설정되어있다() {
//      let existNavigationBar = app.navigationBars["즐겨찾기"].exists // 7. NavigationBar의 제목이 즐겨찾기가 존재하는지 확인
//      XCTAssertTrue(existNavigationBar)
//  }
//
    // 7. 손쉽게 작성
    func test_test() {

    }
}

```

Line: 38 Col: 3

```
final class MachachaUITests: XCTestCase {
    override func tearDown() {
        app = nil                           // 5. 앱을 nil로 초기화
    }

    // 6. test할 화면을 함수로 만들기
//  func test_navigation의_Title이_즐겨찾기로_설정되어있다() {
//      let existNavigationBar = app.navigationBars["즐겨찾기"].exists // 7. NavigationBar의 제목이 즐겨찾기가 존재하는지 확인
//      XCTAssertTrue(existNavigationBar)
//  }
//

    // 7. 녹화버튼을 눌러 손쉽게 작성
    func test_test() {
        let app = XCUIApplication()
        app.buttons["로그인 건너뛰기"].tap()
        app.buttons["내정보"].tap()
        app.scrollViews.otherElements.buttons["1, 즐겨찾기"].tap()
    }
}

```



마무리

BDD(Behavior Driven Develop, 시나리오를 기반으로 테스트 케이스를 작성하는 방법)
를 따르는 UniTest/UITest 작성해보기

FREE PRESENTATION TEMPLATE

THANK YOU