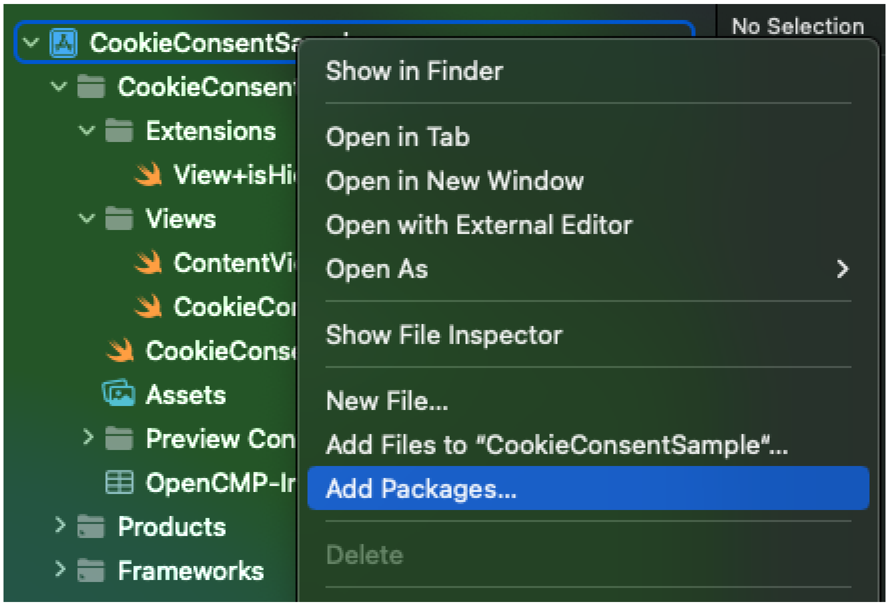
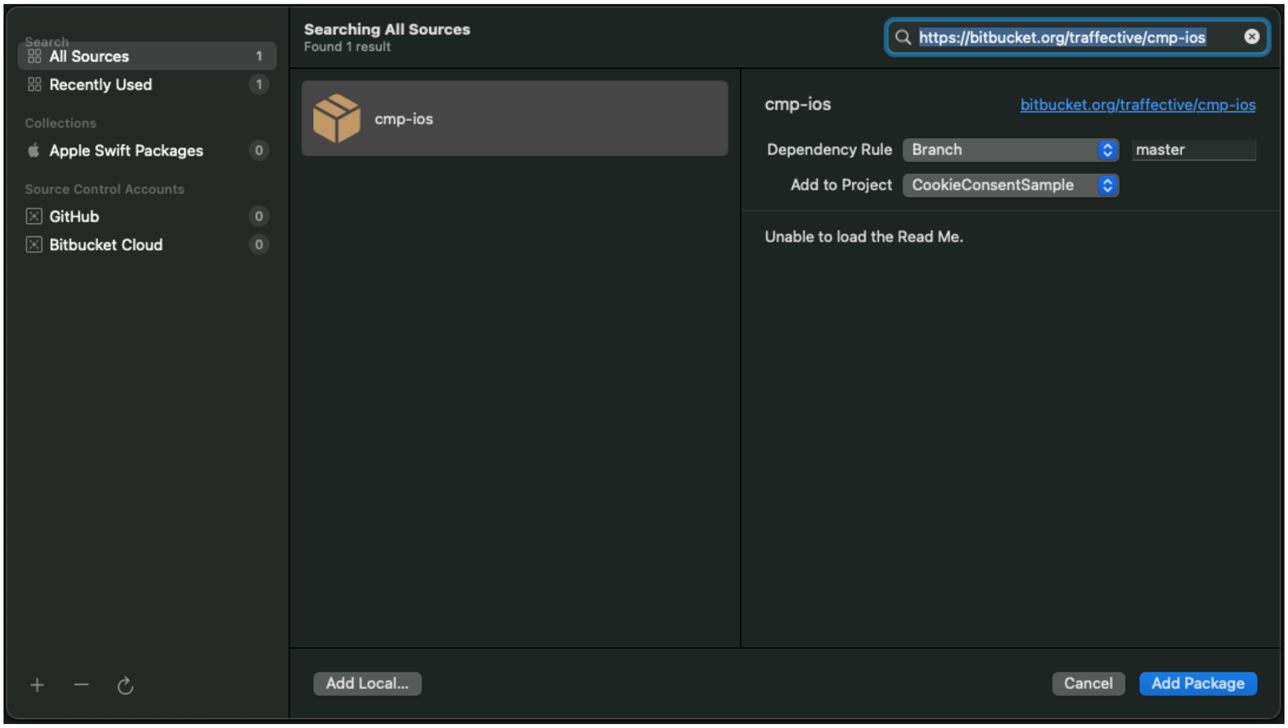
Open CMP iOS

Getting Started

1. You must sign up on github.com to download the Open CMP Swift Package
2. Open Xcode and navigate to your iOS app project, as below click “Add Packages…”



3. Our Open CMP package is publicly available. Therefore, with your GitHub account you have access to
4. Hooray! You are ready to download the Swift Package. You should navigate to “All Sources” or “Recently Used” to get to the search field on the top right, as below
5. Copy and paste the url https://github.com/OpenCMP-net/cmp-ios into the search field. You will get the ***cmp-ios*** Swift Package listed in the window



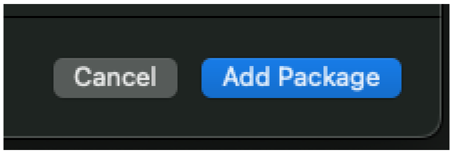
cmp-ios bitbucket.org/traffective/cmp-ios

Dependency Rule: Branch — master

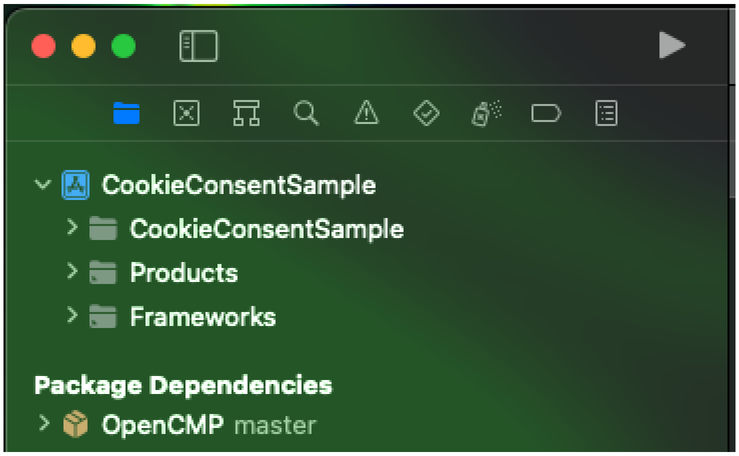
Add to Project: CookieConsentSample

Unable to load the Read Me.

8. Now hit the button ***Add Package*** to finally add the new package to your Xcode project

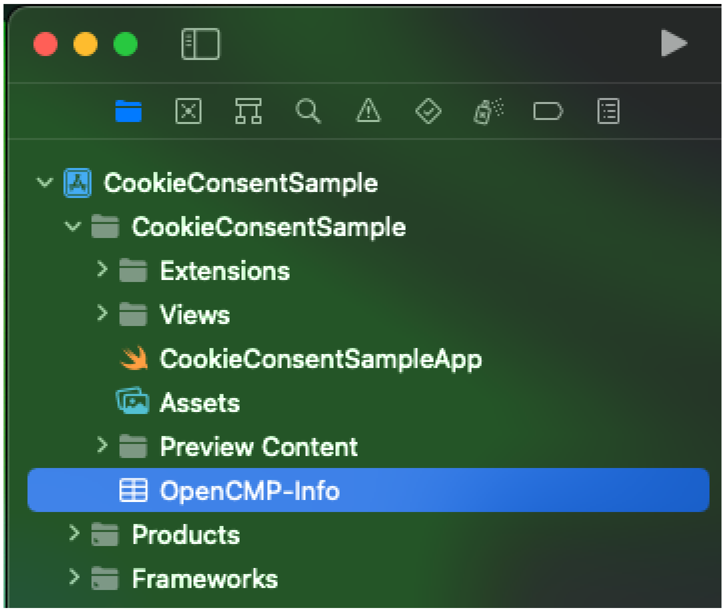


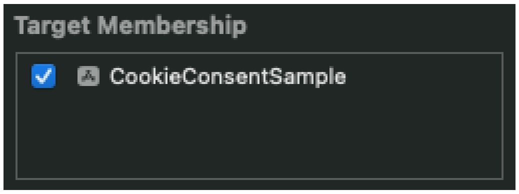
9. The Open CMP Swift Package is listed in the Xcode navigator, as below



Apply configuration

You can start with the default configuration below. However, we provide you with a customized configuration that you can use to apply your styling, etc. You need to copy the configuration file **OpenCMP-Info.plist** to the root directory.




Target Membership

☑ CookieConsentSample

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>domain</key>
        <string>traffective.com</string>
        <key>allowedDomains</key>
        <array>
                <string>cdn.opencmp.net</string>
                <string>www.traffective.com</string>
        </array>
        <key>disableDomainAccessPolicy</key>
        <true/>
        <key>printBrowserLogs</key>
        <false/>
</dict>
</plist>
```

UIKit integration

Our Open CMP framework is developed in UIKit. Therefore it becomes easy to add the view of the Cookie Consent Banner into your app code. You will receive the cookie data via the ***acceptOrReject*** closure. It will dismiss the Open CMP view whenever the user accepted or rejected the consent.

Hint

> *Sometimes it is a challenge to further process data returned in a closure. We suggest to work with the* **Combine** *framework.*

```
let uiView = try? OpenCMPView.shared(acceptOrReject: { cookies in
  // You will receive the stored 'cookies' here.
}, showUi: { }, hideUi: { })
addSubview(uiView)
```

Show the consent ui again if a user clicks a button

```
OpenCMPView.shared?.showUiByUser()
```

Hint

> *You will get the new cookie consent again in the ‘acceptOrReject’ closure.*

Sample cookie data

The *cookie object* is strongly typed. Therefore it suggest you all the attributes of the *cookie object*. We also give you a sample of the possible cookie data:

```
tcf : "CPaXqOpPaXqPbAVACAENCTCsAP_AAH_AAAYgIZtV7T9cbGlDOX59YNt
kWIUH1lAFouQCCBaAE6ABwCOAcKQEw2ASIEzoACACIBgAoDKBIAAEGEEQAEAQQ
IgBADHgIgiEhAAKIAJAABMBAQAAAAsKAAAAEAAIhEAZIACAmCqgQg5mQkAEIIA
QQgABAAgAAKABAoMABAEIABgAAAAAgQAAAAAAAIIZgEiCpcQAFCGGBJIEEECIE
QVhAFAKACCACAAiAAAAAAYKQAUUACAAAIAAAAAAAAoAYBAAAAAEgAAEAAQIAAA
BAAAAAAAAIAAJAABABAAAAAAgCAAAAAAAAAAAAAAAgAAQAAQCQEAAEAAQAgA
BAAAAAIAAAAAAAAAIAAAAAAAAAAAAAAAAIAAAAA.YAAAAAAAAAAA"
google : nil
custom : "CPaXqOpPaXqPbAVACAENCTCgAAAAAAAAAAAAFBwAgAWgKDAAAAA
A.YAAAAAAAAAAA"
meta : ConsentMetaData
  dv : 297
preferences : ConsentPreferences
  cmpSdkId : 21
  cmpSdkVersion : 2
  policyVersion : 2
  gdprApplies : 1
  publisherCc : "DE"
  useNonStandardStacks : 0
  vendorConsents : "110101010111101101001111110101110001101100
01101001010000110011100101111110011111010110000011011011011001
00010110001000010100000111110101100101000000000101101000101110
01000000001000001000000101101000000000010011101000000000000111
00000000100011100000000111000010100100000001001100001101100000
00010010001000000100110011101000000000000010000000000010001000
00000110000000000010100000001100101000000100100000000000000000
01000001100001000001000100000000000001000000000100000100000010
```

```
00100000000001000000000011000111100000001000100000100010000100
10000100000000000000101000100000000010010000000000000000000100
11000000010000000100000000000000000000000000000010110000101000
00000000000000000000000100000000000000001000100001000100000000
01100100100000000000001000000010011000001010101010000001000010
00001110011001100100001001000000000001000010000010000000000100
00010000100000000000000001000000000000100000000000000000001010
00000000000100000010100000110000000000000100000000010000100000
00000000011000000000000000000000000000000000001000000100000000
00000000000000000000000000000000000000001"
  vendorLegitimateInterests : "0000000100100010000010101001011
10001000000000000000101000010000110000110000000100100100000000010
00001000001000000100010000000100010000010101100001000000000001010
00000001010000000000010000010000000000010000000000000100010000
00000000000000000000000000000000001100000101001000000000000001010
00101000000000000100000000000000000000010000000000000000000000
00000000000000000000000000000101000000000011000000001000000000
00000000000000000000000010010000000000000000000010000000000000
00100000010000000000000000000000000010000000000000000000000000
00000000000000000000000000000001000000000000000001001000000000
00000000100000000000100000000000000000000000000000000000010000
00000100000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000100000000000000000010
00000000000000001000000001001000000010000000000000000010000000
00000000100000000001000000000000000010000000000000000000000000
00000001000000000000000000000000000000000000000000000000000000
00000100000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000001"
  purposeConsents : "1111111111"
  purposeLegitimateInterests : "0111111111"
  specialFeaturesOptIns : "11"
  publisherConsent : ""
  publisherLegitimateInterests : ""
  publisherCustomPurposesConsents : ""
  publisherCustomPurposesLegitimateInterests : ""
  customVendorConsents : "000000000000000000000000000000000000
00000000100000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000001"
  customVendorLegitimateInterests : ""
```

SwiftUI: Interfacing with UIKit (optional)

Our Open CMP framework is developed in UIKit. No worries! If you use SwiftUI you can easily develop a view class by using the ***Coordinator*** and ***UIViewControllerRepresentable*** interfaces.

```
import SwiftUI
import OpenCMP
import WebKit

struct CookieConsentView: UIViewControllerRepresentable {
  var acceptOrReject: (_ cookies: ConsentCookies?) -> Void

  var showUi: () -> Void

  var hideUi: () -> Void

  func makeUIViewController(context: Context) -> some UIViewController {
    let uiView = try? OpenCMPView.shared(acceptOrReject: acceptOrReject, showUi: showUi, hideUi: hideUi)
    return (uiView!)
  }

  func updateUIViewController(_ uiViewController: UIViewControllerType, context: Context) {

  }

  func makeCoordinator() -> Coordinator {
    Coordinator(self)
  }

  internal class Coordinator {
    let cookieConsent: CookieConsentView

    init(_ cookieConsent: CookieConsentView) {
      self.cookieConsent = cookieConsent
    }
  }
}
```

SwiftUI integration sample code

```
import SwiftUI
import OpenCMP

struct ContentView: View {
  var body: some View {
    CookieConsentView(
      acceptOrReject: { cookies in print(cookies) },
      showUi: {  },
      hideUi: {  }
    )
  }
}
```

Show the consent ui again if a user clicks a button

```
Button("Show Ui", action: {
  OpenCMPView.shared?.showUiByUser()
})
```

Hint

> *You will get the new cookie consent again in the ‘acceptOrReject’ closure.*