

Alchemer Mobile iOS SDK Quick Start Guide

This documentation will help you quickly integrate the Alchemer Mobile (ApptentiveKit) SDK in your iOS app.

This NEW iOS SDK should be implemented for all customers who are looking to update the Alchemer Mobile platform to include new features and functionality that the previous SDK does not support.

There are several options for integrating. This doc will walk through the Swift Package Manager flow.

Add Alchemer Mobile

In Xcode go to File > Add Packages... and enter <https://github.com/apptentive/apptentive-kit-ios> in the search field. Click Add Package, and then confirm adding the Alchemer Mobile (ApptentiveKit) SDK.

This SDK requires an iOS deployment target of 11.0 and above, so please ensure your project is configured for that.

Initialize Alchemer Mobile

The Alchemer Mobile (ApptentiveKit) SDK is largely operational before it connects with the Alchemer Mobile API, but shortly after each cold app launch, you'll want your app to call the register method.

You'll create an Apptentive.AppCredentials instance to pass in the Alchemer Mobile (Apptentive) App Key and Alchemer Mobile (Apptentive) App Signature values from the API & Development section of the Settings tab in your Alchemer Mobile dashboard.

```
import ApptentiveKit

func application(_ application: UIApplication, didFinishLaunchingWithOptions
    launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
    // ...

    Apptentive.shared.register(with: .init(key: "<#Your Apptentive App Key#>", signature: "<#Your Apptentive App Signature#>"))
    // ...

    return true
}
```

Engage Events

Events record user interaction. You can use them to determine if and when an Interaction will be shown to your customer. At a minimum, you should include 20-50 Events in your app to start taking advantage of Alchemer Mobile, but for now, let's just create one. To trigger an Event, call the `engage()` method. This will record the Event, and then check to see if any Interactions targeted to that Event are allowed to be displayed, based on the logic you set up in the Alchemer Mobile Dashboard.

In this example, trigger an Event when your main view controller appears:

```
import UIKit
import Apptentive

class MainViewController: UIViewController {

    // ...

    override func viewDidLoad(animated: Bool) {
        super.viewDidLoad(animated: animated)

        Apptentive.shared.engage(event: "main_view_appeared", from: self)
    }
}
```

Add Customer ID

You can send Custom Data associated with a person's profile that is using the app, or the device. In particular, this is useful for sending a Customer ID and other information that helps you understand and support your users better. Custom Data can also be used for configuring when Interactions will run. You can add custom data of type `String`, `Number`, and `Boolean`.

Below is an example of a Customer ID value being passed in, along with whether that customer is on a premium account of the app or not:

```
Apptentive.shared.customPersonData["CustomerID"] = "1234321"
Apptentive.shared.customPersonData["is_premium"] = true
```

After setting your Customer ID and other custom data, you can choose which field is your Customer ID in the Alchemer Mobile Platform.

Related Articles