Microsoft

2024

Clarity Mobile SDK for Native iOS Applications

PRIVATE PREVIEW CUSTOMER DOCUMENTATION

CLARITY APPS

Microsoft

Contents

Microsoft

Overview

The Clarity SDK for iOS allows you to capture essential information about your user interactions with all parts of your app. This information can be used to replay user sessions, view heatmaps, and monitor key application signals via the metrics dashboard. Integrating Clarity SDK into your mobile application requires minimum development effort. The SDK can be retrieved through an xcframework file hosted on a private storage account.

How does Clarity Native iOS SDK work?

1. Once the SDK is integrated into the mobile application and is initialized, data is automatically captured and sent to Clarity's servers. The SDK captures:
    A. User screens / visuals
    B. User interactions (taps, double taps and touch motions)
2. Captured data will be periodically uploaded while the application is running, and the device is connected to the internet.
3. The data takes about 30 mins up to 2 hours to show up on the dashboard.

SDK Integration

1. Create a Clarity project with the "Mobile app" type from Microsoft Clarity. Click on "Get started" button and take note of the *project ID*.
2. Prepare a local Clarity package folder using either A. or B.:
    A. Clone from the private GitHub repo amralaa-MSFT/Clarity-iO to a local folder.
    B. The Clarity Beta iOS SDK is published on a private storage account. We'll provide you with a URL to the xcframework file as well as the checksum of the framework file. Then you'd need to create a new folder anywhere on your Mac and create a file inside it called **Package.swift** that should look something like this:

```
// swift-tools-version:5.8
import PackageDescription

let package = Package(
    name: "Clarity",
    platforms: [
        .iOS(.v13)
    ],
    products: [
        .library(
            name: "Clarity",
            targets: ["Clarity"]
        ),
    ],
    targets: [
        .binaryTarget(
```

```
            name: "Clarity",
            url: "https://clarityappsresources.blob.core.windows.net/ios/Clarity-
0.1.0.xcframework.zip?<token>",
            checksum: "<checksum>"
        ),
    ]
)
```

Note: that you'd need to plug in the <token> & <checksum> placeholders.

3. Add package dependency to your XCode project and select the local folder containing Package.swift file.
4. Initialize Clarity by calling the **ClaritySDK.initialize()** function once on the application startup, e.g. in AppDelegate.application(:didFinishLaunchingWithOptions), with the required configuration as shown below:

```
import Clarity
import UIKit

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {
    func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        // Override point for customization after application launch.
        ClaritySDK.initialize(config: ClarityConfig(projectId: "<ProjectID>", logLevel:
LogLevel.verbose))
        return true
    }
}
```

Notes:

- To ensure that Clarity functions properly, you should set the LogLevel to verbose when you are testing.

Microsoft

Configuration Options

```
@objc public class ClarityConfig: NSObject {
    @objc public init(
        projectId: String,
        userId: String? = nil,
        logLevel: LogLevel = .none
    ) {
        // ...
    }
}
```

Project ID

Distinct project ID in Clarity's database.

User ID

A unique ID for the current user. The same user ID persists across sessions on the same device. If no user ID is given, a random one will be saved in the app's UserDefaults database and used for later sessions.

User ID must:

1. Not be blank.
2. Should be base36 and smaller than "1Z141Z4".

Log Level

The level of logging to show in the device's Console or XCode's console while debugging. The SDK captures no logging by default.

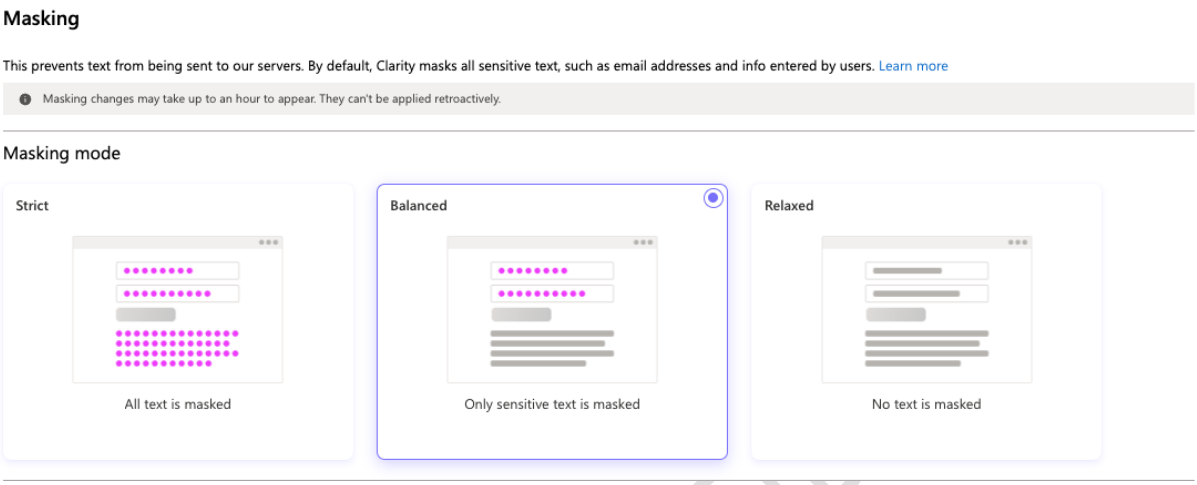
Clarity APIs

```
/// Masks a certain view instance.
@objc public static func maskView(_: UIView)

/// Unmasks a certain view instance.
@objc public static func unmaskView(_: UIView)

/// Sets a custom user ID value.
@objc public static func setCustomUserId(_: String)

/// Sets a custom session ID value.
@objc public static func setCustomSessionId(_: String)

/// Sets a custom tag, a key-value pair that can be used to filter sessions in Clarity dashboard.
@objc public static func setCustomTag(key: String, value: String)
```

Microsoft

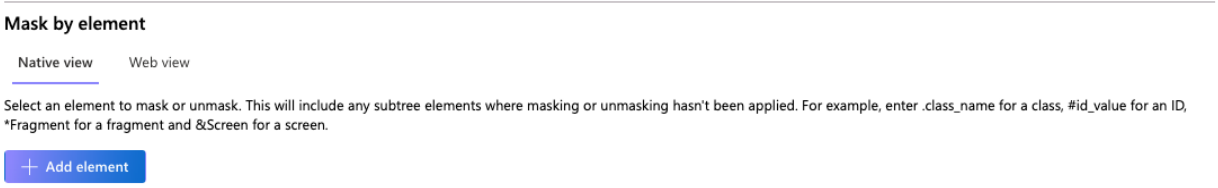
Masking Modes (Dashboard configuration)

The mode of masking to be applied to your content while Clarity collects data. Clarity filters masked information on the client and doesn't send it to the Clarity servers.



- **Balanced**: PII (digits & emails) are masked by default.
- **Strict**: all text and images will be masked.
- **Relaxed**: nothing is masked by default except input fields.

Mask by element:



You can add masking rules on the dashboard settings page, which will reflect on your application and (un)mask the target elements. We have few support selectors:

- **View type**: ".view_type", for example, masking .CustomUILabel would mask all occurrences of the CustomUILabel view in your application.
- **Screen title**: "&screen_title", for example, masking &HomeVC would mask all content inside the view controller whose title is set to HomeVC. (Note that for this to take effect, the specified view controller must be a top-level one).
    - You can unmask view controllers/views inside a masked screen one and vice versa.
- **View controller title**: "*view_controller_title", for example, masking *SettingsVC would mask all content inside the view controller whose title is set to SettingsVC.
    - You can unmask view controllers/views inside a masked one and vice versa.

Notes:

5. You can use the **maskView API** to mask a specific view.
6. Child views of masked views will also be masked.
7. Input fields are always force masked and cannot be unmasked.

Performance

The Clarity Mobile SDK is designed to be lightweight with minimal performance impact on your app.

Main Thread Usage

The Main (UI) thread usage by Clarity highly depends on the nature of your application. The below numbers were obtained via testing on a representative set of applications and device configurations.

Here are some key results:

Percentage of UI thread time used by Clarity	2.8%

Bandwidth Usage

The SDK will stream the recording data directly to Clarity's servers while the app is running. However, the session recording will be available in Clarity dashboard 30 minutes after the session ends.

You can expect to see around **5 KB (per second of recording)** going to Clarity excluding images**.** If your app is image/graphics-heavy, it may experience higher outgoing traffic at first due to asset uploading to all Clarity servers.

Disk Usage

Clarity buffers some of the data on disk before sending it. The data is written to a temporary storage in the dedicated app's documents directory. Moreover, the data will be periodically deleted by the SDK.

Size

The Clarity iOS SDK will increase your app's file's size by around **4.5 MB.**

Microsoft

Limitations/Known Issues

- Our iOS SDK supports iOS versions starting 15.0.0 – 17.2.* inclusive. If the iOS version is outside the supported range, the application would still build and run successfully but Clarity would not collect or upload any data.
- Only apps that are built using UIKit are supported.
- Currently, web views aren't supported.
- System dialogs might not be captured as expected.
- Notifications, navigation bars and the keyboard are external components to the app and thus are not going to be captured.
- Multiple touches are not currently supported.
- Offline sessions are not currently getting captured.

Microsoft

FAQs

When does the session start and end?

The session starts the moment the application is started by the user. A single session's maximum duration is 30 minutes. After 30 minutes a new session is started. If the user paused the app and got back to it within the 30 minutes window, the previous session will be continued until its total length is beyond 30 minutes. One session translates to one recording.

Does Clarity SDK support offline mode?

No, currently Clarity iOS SDK does not support offline sessions.

Is it possible for Clarity to slowdown my app?

The Clarity SDK is designed to be efficient and cognizant of critical applications contexts to avoid causing a perceptible performance impact on the host application. If you notice a significant performance drop, please report back to us by sending an email to *clarity-apps-support@microsoft.com*

I have installed Clarity SDK and I cannot view the user's recordings yet.

Uploaded recordings could take up to 2 hours to be available for playback.

How can I update the SDK?

With every new update, we'll share with you a new link and a new checksum that corresponds to the new version. Then you should update the Package.swift file with these values.