Documentation Language: Swift

StoreKit / DisplayMessageAction

Structure

DisplayMessageAction

An instance that asks StoreKit to display an App Store message, if appropriate.

StoreKit | SwiftUI | iOS 16.0+ | iPadOS 16.0+ | Mac Catalyst 16.0+ | visionOS 1.0+

```
@MainActor
struct DisplayMessageAction
```

Overview

A StoreKit message represents a sheet that appears over your app to display important information from the App Store to the customer. Messages have a reason, indicated by the `reason` value. StoreKit retrieves any messages from the App Store each time your app launches, and presents them by default. Your app can optionally delay or suppress App Store messages by listening for the messages and determining the appropriate time to ask the system to display them.

To use this API, read the `displayStoreKitMessage` environment value to get an instance of the structure for a given `Environment`. Call the instance to ask StoreKit to display the App Store message. StoreKit displays a message only if it’s still pending. It doesn’t display expired messages. You call the instance directly because it defines a `callAsFunction(_:)` method that Swift calls when you call the instance.

> Note
>
> If your app uses `UIWindowScene` and not SwiftUI views, use `display(in:)` instead.

The following code example listens for App Store messages and decides whether to defer them by saving them to an array, display them immediately, or suppress them. A private function that the app calls according to its logic asks the system to display all the deferred messages.

```
import SwiftUI
import StoreKit

struct MessageExampleView: View {
    @Environment(\.displayStoreKitMessage) private var displayStoreMessage
    @State private var deferredMessages = [Message]()

    enum MessageBehavior {
        // Display the message at a later time.
        case displayLater
        // Display the message immediately.
        case displayNow
        // Do not display the message.
        case ignore
    }

    var body: some View {
        Text("Hello World")
        .task {
            for await message in StoreKit.Message.messages {
                let behavior = processMessage(message)
                switch behavior {
                    case .displayNow:
                        try? displayStoreMessage(message)
                    case .displayLater:
                        // Save the message to display it later.
                        deferredMessages.append(message)
                    case .ignore:
                        // Suppresses the message.
                        break
                }
            }
        }
    }

    private func processMessage(_ message: Message) -> MessageBehavior {
        var messageBehavior: MessageBehavior

        // Add your logic here to determine how your app should handle the message.

        return messageBehavior
    }
    // Call this function when the app is ready to display deferred messages.
    @MainActor private func displayDeferredMessages() {
        for message in deferredMessages {
            try? displayStoreMessage(message)
        }
        deferredMessages.removeAll()
    }
}
```

Topics

Displaying the message

`func callAsFunction(Message) throws`

Tells StoreKit to display the App Store message, if appropriate.

Relationships

Conforms To

`Sendable`, `SendableMetatype`

See Also

Messages

`struct Message`

An instance for receiving and displaying App Store messages in your app.

`struct Reason`

Reasons for the App Store messages.