

[Authentication Services](#) / [WebAuthenticationSession](#)

Structure

WebAuthenticationSession

A SwiftUI environment value that views use to authenticate someone using a web service.

◆ [AuthenticationServices](#) | ◆ [SwiftUI](#) | iOS 16.4+ | iPadOS 16.4+ | Mac Catalyst 16.4+ | macOS 13.3+ | tvOS 16.4+ | visionOS 1.0+ | watchOS 9.4+

```
@MainActor
struct WebAuthenticationSession
```

Overview

You access an instance of this type by using the SwiftUI [Environment](#) property wrapper and specifying [webAuthenticationSession](#) as the environment value.

To begin an authentication session and display the browser, call [authenticate\(using:callbackURLScheme:preferredBrowserSession:\)](#). For example, when someone taps a button, the web service authenticates that person and then the authentication provider redirects the browser to a URL it constructs using the app's custom callback scheme. The browser detects that redirect, dismisses itself, and returns the complete URL to the awaiting caller.

The following example shows how to use a SwiftUI [Button](#) to invoke a session:

```
struct WebAuthenticationSessionExample: View {
    // Get an instance of WebAuthenticationSession using SwiftUI's
    // @Environment property wrapper.
    @Environment(\.webAuthenticationSession) private var webAuthenticationSession

    var body: some View {
        Button("Sign in") {
```

```

Task {
    do {
        // Perform the authentication and await the result.
        let urlWithToken = try await webAuthenticationSession.authenticate(
            using: URL("https://www.example.com")!,
            callbackURLScheme: "x-example-app"
        )
        // Call the method that completes the authentication using the
        // returned URL.
        try await signIn(using: urlWithToken)
    } catch {
        // Respond to any authorization errors.
    }
}
}
}

```

After receiving the URL, inspect it to determine the authentication request's outcome. For example, you might search the URL's query parameters for a token of some form:

```

let queryItems = URLComponents(string: urlWithToken.absoluteString)?.queryItems
let token = queryItems?.filter({ $0.name == "token" }).first?.value

```

Note

Refer to your authentication provider's documentation for information on the structure of the returned URL and, if necessary, how you should parse it.

Topics

Authenticating a session

```

func authenticate(using: URL, callback: ASWebAuthenticationSession.Callback,
preferredBrowserSession: WebAuthenticationSession.BrowserSession?,
additionalHeaderFields: [String : String]) async throws -> URL

```

Begins a web authentication session.

```
struct BrowserSession
```

Describes the preferred browser session behavior.

class Callback

An object for evaluating navigation events in an authentication session.

Deprecated methods

~~func authenticate(using: URL, callbackURLScheme: String, preferredBrowserSession: WebAuthenticationSession.BrowserSession?) async throws → URL~~

Begins a web authentication session.

Deprecated

Relationships

Conforms To

Sendable, SendableMetatype

See Also

Web authentication sessions

Authenticating a User Through a Web Service

Use a web authentication session to authenticate a user in your app.

Securing Logins with iCloud Keychain Verification Codes

Use time-based codes generated on-device for a secure authentication experience.

class ASWebAuthenticationSession

A session that an app uses to authenticate a user through a web service.

Supporting Single Sign-On in a Web Browser App

Extend your web browser app to handle web authentication requests from other apps.

class ASWebAuthenticationSessionWebBrowserSessionManager

A session manager that mediates sharing data between an app and a web browser.

ASWebAuthenticationSessionWebBrowserSupportCapabilities

A collection of keys that a browser app uses to declare its ability to handle authentication requests from other apps.