Authentication Services / ASCredentialProviderViewController

Class

ASCredentialProviderViewController

A view controller that a credential manager app uses to extend AutoFill.

iOS 12.0+ | iPadOS 12.0+ | Mac Catalyst 14.0+ | macOS 11.0+ | visionOS 1.0+

```
@MainActor
class ASCredentialProviderViewController
```

Overview

To integrate a password, passkey, or one-time passcode manager app with AutoFill:

1. Add a Credential Provider Extension target to your project that subclasses `ASCredentialProviderViewController`. Add the `AutoFill Credential Provider Entitlement` to both the extension and its containing app.

2. Override the view controller’s `prepareCredentialList(for:)` method to prepare a view with a list of credentials that the person can choose from after opening your extension from the AutoFill suggestions list.

3. Optionally add `ASPasswordCredentialIdentity` and `ASPasskeyCredentialIdentity` instances to the shared `ASCredentialIdentityStore` to make identities available directly in the AutoFill suggestions list. Then override the `provideCredentialWithoutUserInteraction(for:)` method to provide the associated credentials when the person taps a suggestion.

4. Optionally, override the `prepareInterfaceForExtensionConfiguration()` method to specify a configuration interface that you can show when people first enable your credentials manager in Settings.

Receiving credential updates

Apps and websites that allow sign-ins can signal updates to the operating system with the `ASCredentialUpdater` class. The various “report” methods of `ASCredentialUpdater` work like the “signal” methods of `PublicKeyCredential` when using WebAuthn on the web. For example, a website or app can notify credential manager apps that it updated a user name or email for a given account, allowing the manager to stay consistent with the website.

Your credential manager manager receives these updates in the “report” methods of `ASCredentialProviderViewController`. Use these calls to update your manager’s stored credential data or behavior. For example, a call to `reportUnusedPasswordCredential(forDomain:userName:)` can indicate that someone using a passkey will no longer use a password to sign in to a given domain, or that they deleted their account. In this case, the manager should stop showing the user name and password for that domain.

> Note
>
> This class ignores calls from Mac apps built with Mac Catalyst.

Topics

Getting the extension context

`var extensionContext: ASCredentialProviderExtensionContext`

The context your credential provider extension uses to provide information to the system.

`class ASCredentialProviderExtensionContext`

A mechanism that credential provider extensions use to communicate with the system.

Configuring the credential provider extension

`func prepareInterfaceForExtensionConfiguration()`

Prepares the interface to enable the user to configure the extension.

`class ASCredentialIdentityStore`

A container that your extension fills to provide credentials through the QuickType bar.

Selecting a credential

`func prepareCredentialList(for: [ASCredentialServiceIdentifier])`

Prepares the interface to display a list of credentials from which the user can select.

`func prepareCredentialList(for: [ASCredentialServiceIdentifier], requestParameters: ASPasskeyCredentialRequestParameters)`

Prepares the interface to display a list of passkey and password credentials from which the user can select.

`func prepareOneTimeCodeCredentialList(for: [ASCredentialServiceIdentifier])`

Prepares the interface to display a list of one-time passcodes (OTPs) that people can select from.

`func prepareInterface(forPasskeyRegistration: any ASCredentialRequest)`

Prepare the view controller to show user interface for registering a new passkey.

`func prepareInterfaceToProvideCredential(for: any ASCredentialRequest)`

Prepare the view controller to show user interface for providing the requested credential.

`func provideCredentialWithoutUserInteraction(for: any ASCredentialRequest)`

Attempts to provide the user-requested credential with no further user interaction.

`func performWithoutUserInteractionIfPossible(passkeyRegistration: ASPasskeyCredentialRequest)`

Perform a conditional passkey registration, if possible.

`class ASCredentialServiceIdentifier`

An identifier representing a particular service for which the user needs a credential, like a web site.

`protocol ASCredentialRequest`

A protocol that describes a request from the user for your extension to provide a credential.

`class ASPasswordCredentialRequest`

A class that represents a request to supply a password credential.

`class ASOneTimeCodeCredentialRequest`

`protocol ASAuthorizationPublicKeyCredentialRegistrationRequest`

An interface that defines properties for a credential registration request.

`class ASPasskeyCredentialRequest`

A class that represents a request to supply a passkey credential.

`class ASPasskeyCredentialRequestParameters`

A class that represents information about a passkey credential request.

Providing text to AutoFill

```
func prepareInterfaceForUserChoosingTextToInsert()
```

Prepare the view controller to show a list of all insertable text with user selectable fields.

Recognizing errors

```
struct ASExtensionError
```

A credential provider extension error.

```
let ASExtensionErrorDomain: String
```

The domain for a credential provider extension error.

```
enum Code
```

The codes for a credential provider extension error.

Accessing settings

```
class ASSettingsHelper
```

A class that opens Settings and navigates to the settings for configuring credential providers.

Receiving credential updates

```
func reportAllAcceptedPublicKeyCredentials(forRelyingParty: String, userHandle: Data, acceptedCredentialIDs: [Data])
```

Receives a report from the system that a relying party sent a snapshot of all accepted credentials for an account.

```
func reportPublicKeyCredentialUpdate(forRelyingParty: String, userHandle: Data, newName: String)
```

Receives a report from the system that a relying party indicated that a passkey’s user name updated.

```
func reportUnknownPublicKeyCredential(forRelyingParty: String, credentialID: Data)
```

Receives a report from the system that a relying party indicated a passkey credential is invalid.

`func reportUnusedPasswordCredential(forDomain: String, userName: String)`

Receives a report from the system that a relying party indicatd that a password credential isn’t needed anymore for a given user name.

Deprecated methods

~~`func provideCredentialWithoutUserInteraction(for: ASPasswordCredentialIdentity)`~~

Attempts to provide the user-requested credential with no further user interaction.

Deprecated

~~`func prepareInterfaceToProvideCredential(for: ASPasswordCredentialIdentity)`~~

Prepares the interface for a user interaction, like a database login, that enables it to access and return the credential for the given identity.

Deprecated

Instance Methods

`func performWithoutUserInteraction(generatePasswordsRequest: ASGeneratePasswordsRequest)`

Attempt to generate passwords based on developer-specified rules.

Beta

`func performWithoutUserInteractionIfPossible(savePasswordRequest: ASSavePasswordRequest)`

Attempt to save a password credential.

Beta

`func prepareInterface(for: ASSavePasswordRequest)`

Prepares the interface to display a prompt to save a password credential.

Beta

`func prepareInterface(for: ASGeneratePasswordsRequest)`

Prepares the interface to display a prompt to generate passwords based on developer-specified rules.

Beta

Relationships

Inherits From

`NSViewController`, `UIViewController`

Conforms To

`CVarArg`
`CustomDebugStringConvertible`
`CustomStringConvertible`
`Equatable`
`Hashable`
`NSCoding`
`NSEditor`
`NSExtensionRequestHandling`
`NSObjectProtocol`
`NSSeguePerforming`
`NSStandardKeyBindingResponding`
`NSTouchBarProvider`
`NSUserActivityRestoring`
`NSUserInterfaceItemIdentification`
`Sendable`
`SendableMetatype`
`UIActivityItemsConfigurationProviding`
`UIAppearanceContainer`
`UIContentContainer`
`UIFocusEnvironment`
`UIPasteConfigurationSupporting`
`UIResponderStandardEditActions`
`UIStateRestoring`
`UITraitChangeObservable`
`UITraitEnvironment`
`UIUserActivityRestoring`

See Also

AutoFill credentials

Providing one-time passcodes to AutoFill

Help people efficiently perform multifactor authentication.

`AutoFill Credential Provider Entitlement`

A Boolean value that indicates whether the app may, with user permission, provide user names and passwords for AutoFill in Safari and other apps.