Biometric Authentication for iOS in Swift

Many users rely on biometric authentication like Face ID or Touch ID to enable secure, effortless access to their devices. As a fallback option, and for devices without biometry, a passcode or password serves a similar purpose. Use the LocalAuthentication framework to leverage these mechanisms in your app and extend authentication procedures your app already implements.

The Local Authentication framework provides facilities for requesting a passphrase or Touch ID authentication from users. Developers can display and utilize an authentication prompt by utilizing the function evaluatePolicy of the LAContext class.

Local Authentication

The Local Authentication framework is available to iOS and macOS apps that want to use Face ID and Touch ID.

```
import LocalAuthentication
```

LAContext

An **LAContext** is the object that will be used to interact with both Face ID and Touch ID. The **LAContext** initializer does not require any parameters:

```
// Create an LAContext
var context = LAContext()
```

The **biometryType** enum on an **LAContext** provides information on what authentication mechanisms are available on the user's device. There are three modern options:

1. **LABiometryType.none**, meaning no biometric authentication is available
2. **LABiometryType.touchID**, meaning the device supports Touch ID

3 **LABiometryType.faceID**, meaning the device supports Face ID

LAError

LocalAuthentication implements a descriptive error **LAError** that can provide more information about any errors that may occur during authentication.

```
// If error is an instance of LAError
var code = LAError.Code(rawValue: error.code)

switch code {
case LAError.Code.appCancel:
    // The app canceled authentication by
    // invalidating the LAContext
case LAError.Code.authenticationFailed:
    // The user did not provide
    // valid credentials
case LAError.Code.invalidContext:
    // The LAContext was invalid
case LAError.Code.notInteractive:
    // Interaction was not allowed so the
    // authentication failed
case LAError.Code.passcodeNotSet:
    // The user has not set a passcode
    // on this device
case LAError.Code.systemCancel:
    // The system canceled authentication,
    // for example to show another app
case LAError.Code.userCancel:
    // The user canceled the
    // authentication dialog
case LAError.Code.userFallback:
    // The user selected to use a fallback
    // authentication method
case LAError.Code.biometryLockout:
    // Too many failed attempts locked
    // biometric authentication
case LAError.Code.biometryNotAvailable:
    // The user's device does not support
    // biometric authentication
case LAError.Code.biometryNotEnrolled:
```

```
    // The user has not configured
    // biometric authentication
@unknown default:
    // An other error occurred
}
```

Login with Face ID Sample Code

Use `evaluatePolicy(_:, localizedReason:, reply:)` to show the Face ID authentication popup on a device that supports Face ID and where the user has configured Face ID:

```
let reason = "Log in with Face ID"
context.evaluatePolicy(
    // .deviceOwnerAuthentication allows
    // biometric or passcode authentication
    .deviceOwnerAuthentication,
    localizedReason: reason
) { success, error in
    if success {
        // Handle successful authentication
    } else {
        // Handle LAError error
    }
}
```

Login with Touch ID Sample Code

Use `evaluatePolicy(_:, localizedReason:, reply:)` to show the Touch ID authentication popup on a device that supports Touch ID and where the user has configured Touch ID:

```
let reason = "Log in with Touch ID"
context.evaluatePolicy(
    // .deviceOwnerAuthentication allows
    // biometric or passcode authentication
    .deviceOwnerAuthentication,
    localizedReason: reason
) { success, error in
    if success {
        // Handle successful authentication
    } else {
        // Handle LAError error
```

```
        }
    }
```