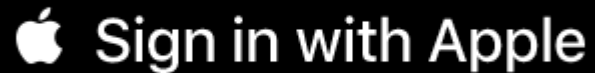


Sign In with Apple using Swift



Priya Talreja [Follow](#)

Jun 10, 2019 · 3 min read



Sign In with Apple allows users to set up an account and sign in to your apps and websites with their Apple ID.













Enabling Sign In with Apple ID

There are two ways using which you can enable Sign In with Apple ID Capability.

Using Apple Developer Account

You can edit your App ID configurations in Certificates, Identifiers & Profiles.

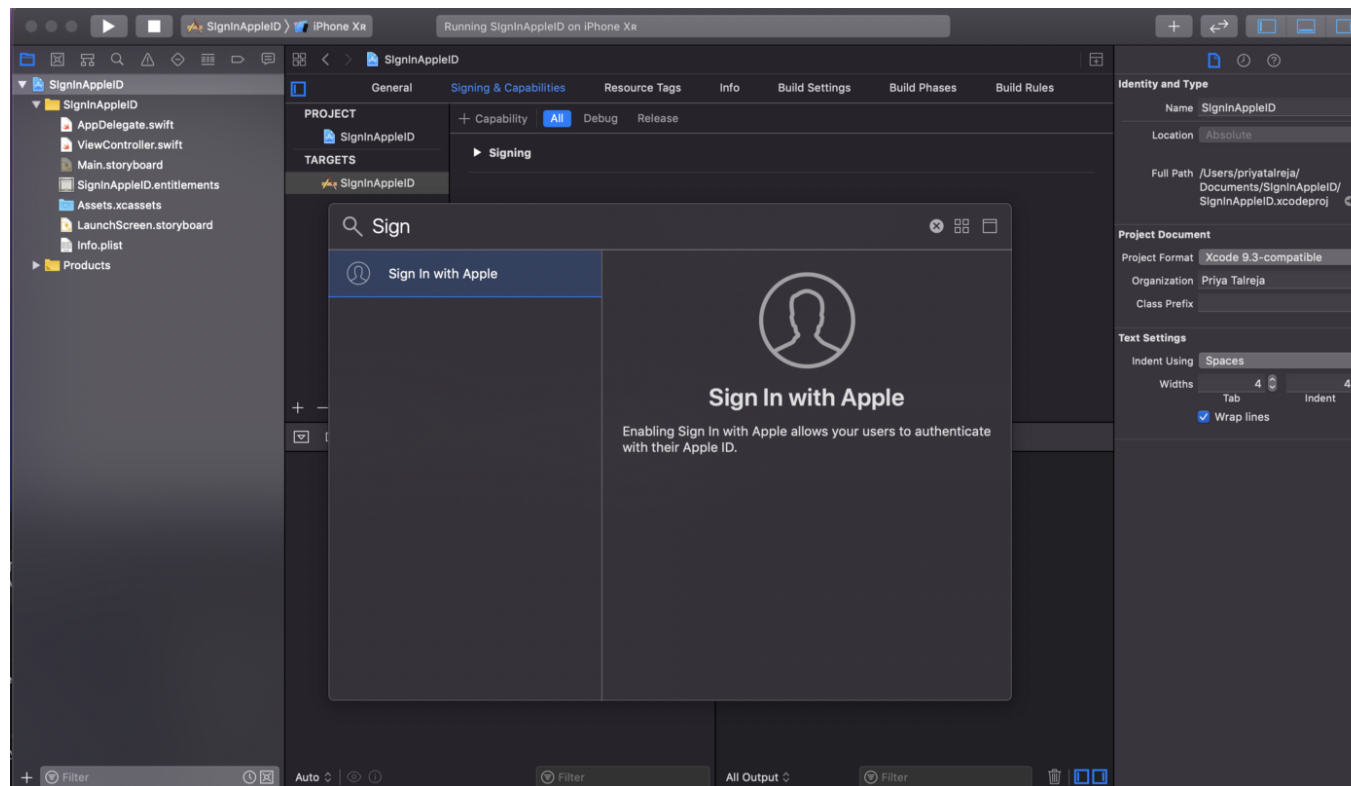
Edit your App ID Configuration Remove Save

<input checked="" type="checkbox"/>	 In-App Purchase	
<input type="checkbox"/>	 Inter-App Audio	
<input type="checkbox"/>	 Low Latency HLS	
<input type="checkbox"/>	 Mac	Configure
<input type="checkbox"/>	 Multipath	
<input checked="" type="checkbox"/>	 Network Extensions	
<input type="checkbox"/>	 NFC Tag Reading	
<input type="checkbox"/>	 Personal VPN	
<input checked="" type="checkbox"/>	 Push Notifications	Edit Certificates (1)
<input type="checkbox"/>	 Sign In with Apple	Configure
<input type="checkbox"/>	 SiriKit	
<input type="checkbox"/>	 System Extension	

Using Xcode

1. Open your Xcode Project.
2. *Project Navigator* → *Select Project* → *Select Target*.
3. In *Project Editor*, Click *Signing & Capabilities*.

4. Add *Capability* by clicking the + button. Search for **Sign In with Apple** Capability in *Capability Library*.
5. Double-click the capability to add.



Sign In with Apple Entitlement

You need to add entitlement if you enable capability using apple developer account. If you configure capabilities using Xcode, Xcode manages the related entitlements so you don't need to edit the entitlements file or an App ID directly.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>com.apple.developer.applesignin</key>
<array>
<string>Default</string>
</array>
</dict>
</plist>
```

Sign In Set up and Functionality

We need to use **Authentication Services** framework to give users the ability to sign into your services with their Apple ID.

```
import AuthenticationServices
```

Add *Sign In with Apple* Button.

```
func setUpSignInAppleButton() {  
  
    let authorizationButton = ASAuthorizationAppleIDButton()  
  
    authorizationButton.addTarget(self, action:  
#selector(handleAppleIdRequest), for: .touchUpInside)  
  
    authorizationButton.cornerRadius = 10  
  
    //Add button on some view or stack  
    self.signInButtonStack.addArrangedSubview(authorizationButton)  
}
```

Function to create a *request* using **ASAuthorizationAppleIDProvider** and initialize a *controller* **ASAuthorizationController** to perform the *request*.

```
@objc func handleAppleIdRequest() {  
  
    let appleIDProvider = ASAuthorizationAppleIDProvider()  
  
    let request = appleIDProvider.createRequest()  
  
    request.requestedScopes = [.fullName, .email]  
  
    let authorizationController =  
    ASAuthorizationController(authorizationRequests: [request])
```

```
authorizationController.delegate = self

authorizationController.performRequests()

}
```

Conform to **ASAuthorizationControllerDelegate**

Below function is called after successful Sign In.

```
func authorizationController(controller: ASAuthorizationController,
didCompleteWithAuthorization authorization: ASAuthorization) {

    if let appleIDCredential = authorization.credential as?
    ASAuthorizationAppleIDCredential {

        let userIdentifier = appleIDCredential.user

        let fullName = appleIDCredential.fullName

        let email = appleIDCredential.email

        print("User id is \(userIdentifier) \n Full Name is \
        (String(describing: fullName)) \n Email id is \(String(describing:
        email))") }

    }
```

You can handle error in the below function.

```
func authorizationController(controller: ASAuthorizationController,  
didCompleteWithError error: Error) {  
  
    // Handle error.  
  
}
```

Check Credential State

On successful authorization, we get User Info which has **User Identifier**.

We can use that identifier to check the user's credential state by calling the `getCredentialState(forUserID:completion:)` method:

```
let appleIDProvider = ASAuthorizationAppleIDProvider()  
appleIDProvider.getCredentialState(forUserID: userID) {  
    (credentialState, error) in  
        switch credentialState {  
            case .authorized:  
                // The Apple ID credential is valid.  
                break  
            case .revoked:
```

```
        // The Apple ID credential is revoked.
        break
    case .notFound:
        // No credential was found, so show the sign-in UI.
    default:
        break
    }
}
```

Sign In with Apple is Complete!

Thank you. Hope this helps you. Please let me know if you have any queries.

[iOS](#) [Sign In With Apple](#) [Swift](#) [Ios 13](#) [Apple Id](#)

Discover Medium

Welcome to a place where words matter. On Medium, smart voices and original ideas take center stage - with no ads in sight. Watch

Make Medium yours

Follow all the topics you care about, and we'll deliver the best stories for you to your homepage and inbox. Explore

Become a member

Get unlimited access to the best stories on Medium — and support writers while you're at it. Just \$5/month. Upgrade

[About](#) [Help](#) [Legal](#)

