ВСЕ, ЧТО ВЫ ХОТЕЛИ ЗНАТЬ О ФУНКЦИИ “SIGN IN WITH APPLE”
Вчерашний В.Е.

Вчерашний Владислав Евгеньевич - руководитель отдела,
отдел разработки мобильного приложения,
Компания Parler LLC,
преподаватель,
курсы разработки приложений для Apple iOS,
г. Киев, Украина

Аннотация: *в статье анализируется работа технологии Sign In with Apple в операционной системе Apple iOS 13.*
Ключевые слова: *apple, iOS, Swift, macOS.*

Введение

На мой взгляд, анонс сервиса Sign in with Apple, который был представлен на WWDC 2019, является одним из интереснейших, если брать во внимание вопрос приватности и конфиденциальности пользовательских данных.

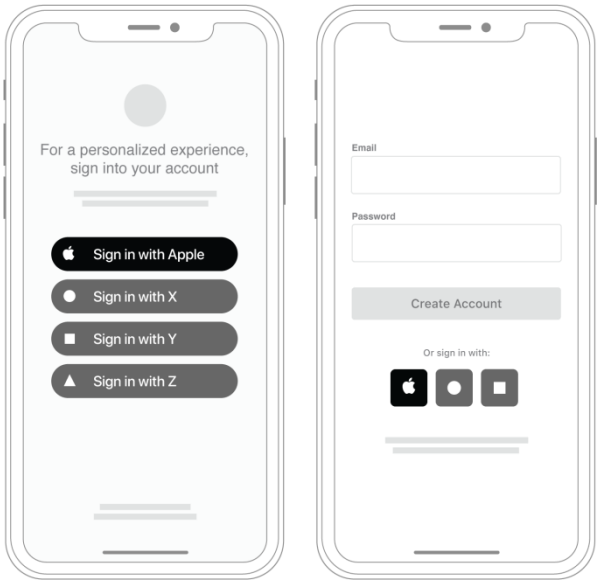

Рис. 1. Дизайн кнопок

Сервис **Sign in with Apple** — это, своего рода альтернатива уже существующей возможности авторизоваться на сайте и/или в приложении через Facebook / Google / Twitter / LinkedIn и другие подобные сервисы. Уже очень давно эти функции стали для нас привычными и удобными, а теперь к этому долгому списку добавляется еще одна, но уже от компании Apple. Так зачем же это нам? – Давайте разбираться, присаживайтесь поудобнее.

Приступая к работе

Важно: *для использования “Sign In with Apple” у вас должен быть куплен аккаунт разработчика.*

Первым делом открываем (создаём) проект и переходим во вкладку Project Navigator > Select your target > Signing and Capabilities tab и добавляем “Sign In with Apple”.

Следующим шагом переходим в ViewController и импортируем библиотеку **AuthenticationServices**:

*// **import** AuthenticationServices*

Далее нам нужно добавить кнопку на экран. Делается это очень просто и всего в несколько строчек кода:

```
// let signInButton = ASAuthorizationAppleIDButton(type: .default, style: .black)
// signInButton.addTarget(self, action: #selector(authorize), for: .touchUpInside)
```

Рис. 2. Строго рекомендуется использовать стандартную кнопку от Apple

Кнопку можно легко кастомизировать, на ваш выбор доступны 3 разных стиля (.black, .white и .whiteOutline) и типов (.signIn, .signUp и .continue). Также у Apple есть подробные и прекрасные гайдлайны здесь.

После добавления кнопки нам нужно добавить обработку действия по нажатию на неё:

```
@objc
  private func authorize() {
    let request = ASAuthorizationAppleIDProvider().createRequest()
    request.requestedScopes = [.fullName, .email]

    let controller = ASAuthorizationController(authorizationRequests: [request])
    controller.delegate = self
    controller.presentationContextProvider = self
    controller.performRequests()
  }
```

Здесь мы:

- Создаём запрос и описываем необходимый скоуп данных (например: имя и почта)
- Создаём ASAuthorizationController и задаем его делегат
- Выполняем созданный запрос

И последнее что нужно сделать — это подключить контроллер к протоколам **ASAuthorizationControllerPresentationContextProviding** и **ASAuthorizationControllerDelegate**, и реализовать их.

```
extension ViewController: ASAuthorizationControllerPresentationContextProviding {
  func presentationAnchor(for controller: ASAuthorizationController) -> ASPresentationAnchor {
    return view.window ?? UIWindow()
  }
}
```

Важно знать: *реализация протокола* ***ASAuthorizationControllerPresentationContextProviding*** *необязательная. Это нужно для корректного отображения окна Sign In with Apple.*

```
extension ViewController: ASAuthorizationControllerDelegate {
  func authorizationController(controller: ASAuthorizationController, didCompleteWithAuthorization authorization: ASAuthorization) {
    switch authorization.credential {
    case let credential as AS Authorization Apple ID Credential:
      let userId = credential.user
      print("User Identifier: ", userId)

      if let fullname = credential.fullName {
        print(fullname)
      }

      if let email = credential.email {
        print("Email: ", email)
      }
    default:
      break
    }
  }
```

```
    func authorizationController(controller: ASAuthorizationController, didCompleteWithError error: Error)
{
        print("Error: \(error.localizedDescription)")
    }
}
```

Важно знать: *после первой авторизации вам больше не будут доступны данные пользователя. Убедитесь в правильном сохранении данных сразу.*

После регистрации и сохранения пользователя его актуальность и легитимность время от времени нужно сверять с данными Apple:

```
private func verifyUser() {
    let provider = ASAuthorizationAppleIDProvider()
    provider.getCredentialState(forUserID: "Saved user id") { (state, error) in
        switch state {
        case .authorized:
            print("authorized")
        case .notFound:
            print("User not found")
        case .revoked:
            print("Apple has revoked user")
        case .transferred:
            print("Transfered")
        @unknown default:
            break
        }
    }
}
```

Пользовательская сторона

Как проходит авторизация со стороны пользователя? – все очень просто, а главное — удобно:

1. Нажать на кнопку “Sign In with Apple”
2. Выбрать информацию, которую будет видно разработчику в показанном окне
3. Подтвердить регистрацию через Face ID/Touch ID

Все, учетная запись создана! Стоит отметить, что Apple ID защищена двухфакторной аутентификацией, а также учетная запись не имеет пароля, это значит, что ее украсть будет невозможно.

К слово о приватности, если пользователь решит не передавать свой адрес электронной почты Apple создаст некий виртуальный адрес и передаст его разработчику, на который впоследствии будут приходить “письма счастья”.

По словам Apple функция Sign In with Apple будет **обязательной** для всех приложений, которые используют возможность авторизации через социальные сети (Google, Facebook и пр.). Если же разработчик откажется от внедрения этой функции, он обязан также убрать и другие кнопки социальных сетей.

Работать новая функция будет не только на всех платформах компании Apple но и на веб-сайтах и приложениях на Android и Windows с помощью представленной библиотеки на JavaScript.

Вывод

Вы еще не поняли, зачем вам это нужно и в чем разница относительно других сервисов? При входе в приложение через Facebook/Google пользователь передает всю информацию о том, какое приложение и когда пользуется и пользователь не имеет достаточного контроля над приватностью (при входе пользователь разрешает или запрещает доступ к определенным данным), а вот что получит Google или Facebook из этого приложения — простите-извините, вам знать не положено. Как говорит сам Facebook в своей политике конфиденциальности: мы можем получить информацию об устройстве, посещенных сайтах, покупках, увиденной рекламе и использовании служб. Собственно, последние скандалы вокруг Facebook ~~тонко~~ намекают, что нужно дважды подумать стоит ли доверять компании столько личной информации, которую впоследствии могут передать другим компаниям.

Я бы хотел еще отметить важную деталь. Я уверен в компании Apple и в том, что ее целью является не сбор дополнительных персональных данных пользователей, а наоборот — минимизировать возможные утечки данных в других приложения, которые сама Apple контролировать не в состоянии.

Список литературы

1. Документация Apple / Human Interface Guidelines. [Электронный ресурс], 2019. Режим доступа: https://developer.apple.com/design/human-interface-guidelines/sign-in-with-apple/overview/introduction/ (дата обращения: 08.01.2021).
2. Документация Apple / Sign In with Apple. [Электронный ресурс], 2019. Режим доступа: https://developer.apple.com/documentation/authenticationservices/implementing_user_authentication_with_sign_in_with_apple/ (дата обращения: 08.01.2021).