

Инструкция по интеграции в iOS приложение

Аналитическая программа с функциями маркетинга и выделения сигналов

SDK для iOS

MERLIN-GL-RU-01.01.00.dINI.01.00

Версия документа: 1.2

СОДЕРЖАНИЕ

1	ВВЕДЕНИЕ	1
1.1	Список изменений	1
2	НЕОБХОДИМЫЕ УСЛОВИЯ	2
2.1	Требования к квалификации специалиста.....	2
2.2	Комплект поставки	2
3	ПОДКЛЮЧЕНИЕ БИБЛИОТЕКИ	3
4	ОПИСАНИЕ ФУНКЦИЙ БИБЛИОТЕКИ	4
4.1	Установка типа уведомлений Firebase Cloud Messaging token или APNS token	4
4.2	Установка токена.....	4
4.3	Интервал времени для фонового обновления (backgroundFetchInterval)	4
4.4	Принудительное выполнение (повторного) сбора данных	4
4.5	Методы инициализации соц.сетей.....	4
4.6	Метод для работы с URL-событиями	5
4.7	Сбор ресурса: месторасположение	5
4.8	Для управления событиями Reachability	5
4.9	Установка данных по серверу, бренду и клиенту.....	5
4.10	Объект типа Permissions	5
5	ИНИЦИАЛИЗАЦИЯ БИБЛИОТЕКИ	8
6	РАБОТА В ФОНОВОМ РЕЖИМЕ	15
6.1	Работа модуля Мерлин в фоновом режиме	15
6.2	Механизмы для работы приложения iOS в фоновом режиме.....	15
6.2.1	<i>BackgroundFetchUIApplication</i>	15
6.2.2	<i>Сервис Google Firebase</i>	15
6.2.3	<i>NotificationCenter</i>	15
6.3	Ограниченные типы приложений которым разрешается работать в фоновом режим в iOS 16	
7	ПОДКЛЮЧЕНИЕ К СОЦИАЛЬНЫМ СЕТЯМ	17
7.1	Пример: Аутентификация и запрос на получение данных для каждой соц. сети (если нужна аутентификация)	22
7.2	Пример: Только запрос на получение данных для каждой соц. сети (быстрая аутентификация).....	24
8	ДОПОЛНИТЕЛЬНЫЕ РАЗРЕШЕНИЯ, КОТОРЫЕ НАДО ДОБАВИТЬ В РАМКАХ ПРОЕКТА	27

Инструкция по интеграции в iOS приложение.

Аналитическая программа с функциями маркетинга и выделения сигналов. MERLIN SDK для iOS.

1 ВВЕДЕНИЕ

Библиотека для iOS поставляется в виде предварительно собранного iOS dynamic framework и предполагает подключение посредством менеджера зависимостей Cocoa Pods (<https://cocoapods.org>). Поддерживается Apple iOS 11.0 и выше.

Настоящая инструкция предназначена для специалиста, выполняющего изложенные в ней процедуры. Требования к уровню его подготовки изложены в разделе 2.1 настоящей инструкции.

1.1 Список изменений

Версия документа	Дата	Изменение
------------------	------	-----------

2 НЕОБХОДИМЫЕ УСЛОВИЯ

2.1 Требования к квалификации специалиста

Все операции, описанные в данной инструкции, должны выполняться специалистом, имеющим соответствующую квалификацию и обладающим правами для осуществления данных операций.

Специалист должен отвечать следующим квалификационным требованиям:

Наличие квалификации Apple iOS разработчика, в том числе:

- владение языком программирования Swift;
- знание iOS API и IDE Apple XCode.

2.2 Комплект поставки

В комплект поставки входят следующие файлы:

1. Merlin-iOS.zip – архив с библиотекой
2. MERLIN-GL-RU-01.01.00.dINI.01.00.pdf – Аналитическая программа с функциями маркетинга и выделения сигналов. SDK для iOS. Инструкция по интеграции в iOS приложение. (Настоящий документ).

3 ПОДКЛЮЧЕНИЕ БИБЛИОТЕКИ

1. Распакуйте архив с библиотекой из комплекта поставки в рабочий каталог.
2. Откройте командную строку и перейдите в каталог `MerlinForIOSArchive` из распакованного архива.
3. Выполните команду для запуска простого http сервера:

```
python -m SimpleHTTPServer 22222
```

Далее с этого сервера будет скачиваться библиотека ('MerlinForIOS'). В случае необходимости вы можете изменить способ хостинга и поменять адрес и порт сервера в файле `{path}/MerlinForIOSArchive/MerlinForIOSArchive.podspec`.

4. В приложение, к которому необходимо подключить библиотеку, добавьте поддержку менеджера зависимостей Cocoa Pods (<https://cocoapods.org>).
5. В файл проекта `Podfile` добавьте путь к `podspec` файлу библиотеки. Путь может быть как абсолютным, так и относительным:

```
pod 'MerlinForIOS', :podspec => '{path to
MerlinForIOSArchive}/MerlinForIOSArchive.podspec'
pod 'Firebase/Core'
pod 'Firebase/Messaging'
```

Библиотеки Google Firebase подключаются для поддержки push уведомлений Firebase.

6. Выполните следующую команду:

```
pod install
```

В результате zip архив из `MerlinForIOSArchive` должен быть скачан с сервера и `pod merlin` подключен к проекту.

7. Для того что бы удалить старый модуль `MerlinForIOS`, выполните следующую команду:

```
find ~/Library/Caches/CocoaPods/ -name "Merlin*" | xargs rm -rf
```

4 ОПИСАНИЕ ФУНКЦИЙ БИБЛИОТЕКИ

Модуль MerlinManager имеет протокол взаимодействия MerlinManagerProtocol, в нем есть следующие методы:

4.1 Установка типа уведомлений Firebase Cloud Messaging token или APNS token

```
func setPushMode(withMode mode: PushJson.PushMode, andDelegate delegate:
UIApplicationDelegate)
```

4.2 Установка токена

Задается токен в виде строки.

```
func setPushToken(withToken token: String)
```

4.3 Интервал времени для фонового обновления (backgroundFetchInterval)

Время измеряется в секундах с плавающей точкой. Данный интервал необходим для запросов на push-уведомления, а так же для работы в фоновом режиме.

```
var backgroundFetchInterval: TimeInterval { get }
```

4.4 Принудительное выполнение (повторного) сбора данных

Функцию стоит вызывать (желательно) с интервалом не чаще чем указано в п.2.2. Функцию надо вызывать для инициирования сбора данных (принудительно), в том числе и в фоне.

Так же раз в сутки присылается silent push для проверки на установленность приложения и для инициирования сбора данных в фоне.

```
func performBackgroundFetch()
```

4.5 Методы инициализации соц.сетей

Здесь надо ввести данные для получение информации через соц.сети.

```
func initOkSdk(okAppId: String, okAppKey: String)
func initVkSdk(vkAppId: String)
func initFbSdk(_ application: UIApplication,
```

Инструкция по интеграции в iOS приложение.

Аналитическая программа с функциями маркетинга и выделения сигналов. MERLIN SDK для iOS.

```
didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey: Any]? ) -> Bool
```

4.6 Метод для работы с URL-событиями

Метод для корректной обработки ответов для SDK соц.сетей.

Его необходимо вызывать из делегата приложения (Application delegate) из метода application(_ :options:)

```
func application(_ app: UIApplication, open url: URL, options:
[UIApplicationOpenURLOptionsKey : Any]) -> Bool
```

4.7 Сбор ресурса: месторасположение

Можно включить/отключить сбор информации о местоположении.

```
func setLocationDataUsed(enable: Bool)

func isLocationDataUsed() -> Bool
```

4.8 Для управления событиями Reachability

```
func startNotificationReachability()

func stopNotificationReachability()
```

4.9 Установка данных по серверу, бренду и клиенту.

Инициализацию клиента надо проводить после того как пользователь согласится с условиями (AgreementPolicy And PrivacyPolicy)

```
func initializeClient(baseUrl: String, brand: String, clientId: String
, completed: @escaping (_ perm: Permissions) -> Void)

func getBaseUrl() -> String

func getClientId() -> String

func getBrand() -> String
```

completed - вернет информацию о статусе/разрешении получения определенных данных на текущий момент.

4.10 Объект типа Permissions

Данный объект хранит информацию о статусе/разрешении получения определенных данных

Инструкция по интеграции в iOS приложение.

Аналитическая программа с функциями маркетинга и выделения сигналов. MERLIN SDK для iOS.

```
public enum MerlinPermissions: String, Codable, CaseIterable {
    case camera = "camera"
    case microphone = "microphone"
    case photo = "photo"
    case healthKit = "health_kit"
    case calendar = "calendar"
    case reminder = "reminder"
    case music = "music"
    case speechRecognition = "speech_recognition"
    case contacts = "contacts"
    case location = "location"
}
```

Установка разрешений:

```
MerlinManager.default.setPermissions(permissions: MerlinPermissions.allCases)
```

Название разрешения	Название поля в JSON	Название переменной	Соответствующие разрешения ОС IOS
Контакты	contacts	Contacts	Наличие InfoPlist.NSContactsUsageDescription, а так же CNContactStore.authorizationStatus равное authorized
Местоположение	location	location	CLLocationManager.authorizationStatus равное authorizedAlways или authorizedWhenInUse
Камера	camera	camera	AVCaptureDevice.authorizationStatus(for: .video) равное authorized
Календарь	calendar	calendar	EKEventStore.authorizationStatus(for: EKEntityType.event) равное authorized
Представление напоминания	reminders	reminders	EKEventStore.authorizationStatus(for: EKEntityType.reminder) равное authorized
Микрофон	microphone	microphone	AVAudioSession.sharedInstance().recordPermission() равное AVAudioSessionRecordPermission.granted
Распознавание речи	speech_recognition	speech_recognition	SFSpeechRecognizer.authorizationStatus() равное authorized

Инструкция по интеграции в iOS приложение.

Аналитическая программа с функциями маркетинга и выделения сигналов. MERLIN SDK для iOS.

Название разрешения	Название поля в JSON	Название переменной	Соответствующие разрешения ОС IOS
Медиа	media_access	media_access	SKCloudServiceController.authorizationStatus() равное authorized
Данные о Bluetooth подключении	bluetooth	bluetooth	CBCentralManager.state равное poweredOn
Программа "Дом"	home_kit	home_kit	HMHomeManager.value(forKey: "_didUpdateHomes") равное TRUE
Фитнес	fitness	fitness	HKHealthStore.authorizationStatus(for: HKSampleType.workoutType()) равное sharingAuthorized
Фото	photos	photos	PHPhotoLibrary.authorizationStatus() равное authorized

5 ИНИЦИАЛИЗАЦИЯ БИБЛИОТЕКИ

1. Открыть/переоткрыть проект с подключенной к нему библиотекой Merlin (файл с расширением xcworkspace).
2. Открыть `ViewController` в котором будет производиться инициализация библиотеки. Отредактировать перегруженную функцию `viewDidAppear`

```
import MerlinForIOS

class ViewController: UIViewController {

    override func viewDidAppear(_ animated: Bool) {

        super.viewDidAppear(animated)

        //...

        let serverUrl = "http://merlin-server:8081/api/v1/"

        let brandName = "TestBrandID"

        let clientID = "TestClientID"

        // Отключение работы с GPS

        // MerlinManager.default.setLocationDataUsed(enable: false)

        // Инициализацию клиента

        // после того как пользователь согласится с условиями (AgreementPolicy And
        PrivacyPolicy)

        MerlinManager.default.initializeClient(baseUrl: serverUrl, brand: brandName,
        clientId: clientID) { permissions in

            print(permissions)

        }

    }

}
```

! **Внимание!** В текущей версии для URL сервера надо использовать https, иначе библиотека не инициализируется и сбор данных не будет работать.

В данном коде производится последовательно: задание URL сервера, задание идентификатора бренда и клиента.

Обратите внимание, что в текущей версии идентификатор клиента (`clientId`) должен быть уникальным (на каждую регистрацию) в рамках всех клиентов сервера (`serverUrl`).

По завершению регистрации модуль возвращает список статусов разрешений.

Если по каким то причинам вы не можете использовать данные о местоположении, то в модуле Merlin надо отключить данный функционал через вызов:

```
MerlinManager.default.setLocationDataUsed(enable: false)
```

Это надо сделать перед вызовом: `MerlinManager.default.initializeClient`.

- Чтобы корректно работали функции фоновой работы, обработки push уведомлений и сбора информации из социальных сетей необходимо также добавить дополнительные обработчики в `app delegate`, которые будут вызывать обработчик из библиотеки Merlin:

```
import UIKit

import Firebase

import FirebaseCore

import FirebaseAnalytics

import FirebaseMessaging

import FirebaseInstanceID

import MerlinForIOS

import UserNotifications

@UIApplicationMain

class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?

    func application(_ application: UIApplication, willFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey : Any]? = nil) -> Bool {

        let permissions = MerlinPermissions.AllCases()
```

```

        MerlinManager.default.setPermissions(permissions: permissions)

        return true
    }

    func application(_ app: UIApplication, open url: URL, options:
[UIApplication.OpenURLOptionsKey : Any] = [:]) -> Bool {

        return MerlinManager.default.application(app, open: url, options: options)
    }

    func application(_ application: UIApplication, didFinishLaunchingWithOptions
launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {

        if #available(iOS 10.0, *) {

            UNUserNotificationCenter.current().delegate = self

            let authOptions: UNAuthorizationOptions = [.alert, .badge, .sound]

            UNUserNotificationCenter.current().requestAuthorization(options:
authOptions, completionHandler: {_, _ in })

            Messaging.messaging().delegate = self
        } else {

            let settings: UIUserNotificationSettings =
UIUserNotificationSettings(types: [.alert, .badge, .sound], categories: nil)

            application.registerUserNotificationSettings(settings)
        }

        application.registerForRemoteNotifications()

        let pushType = Consts.PushType.fcm

        if (pushType == .fcm) {

            FirebaseApp.configure()

            Messaging.messaging().delegate = self as MessagingDelegate

            Messaging.messaging().shouldEstablishDirectChannel = true
        }
    }

```

```

    }

    MerlinManager.default.setPushType(withType: pushType, andDelegate: self)

application.setMinimumBackgroundFetchInterval(MerlinManager.default.backgroundFetchInterval)

    MerlinManager.default.startNotificationReachability()

    return true
}

internal func application(_ application: UIApplication,
didRegisterForRemoteNotificationsWithDeviceToken deviceToken: Data) {

    let tokenString = deviceToken.reduce("", {$0 + String(format: "%02X", $1)})

    Logger.i("deviceToken: \(tokenString)")

    if (MerlinManager.default.getPushType() == .apns) {

        MerlinManager.default.setPushToken(withToken: tokenString)

    } else {

        Messaging.messaging().apnsToken = deviceToken as Data

    }

}

internal func application(_ application: UIApplication,
didReceiveRemoteNotification userInfo: [AnyHashable : Any], fetchCompletionHandler
completionHandler: @escaping (UIBackgroundFetchResult) -> Swift.Void) {

    Logger.i(userInfo)

    MerlinManager.default.performBackgroundFetch(completion: { })

    completionHandler(.newData)

}

internal func application(_ application: UIApplication,
didFailToRegisterForRemoteNotificationsWithError error: Error) {

    Logger.i(error.localizedDescription)
}

```

Инструкция по интеграции в iOS приложение.

Аналитическая программа с функциями маркетинга и выделения сигналов. MERLIN SDK для iOS.

```
}

```

```
func application(_ application: UIApplication, performFetchWithCompletionHandler
completionHandler: @escaping (UIBackgroundFetchResult) -> Void) {

    MerlinManager.default.performBackgroundFetch(completion: {})

    completionHandler(.newData)

}

func applicationWillResignActive(_ application: UIApplication) {

}

func applicationDidEnterBackground(_ application: UIApplication) {

    MerlinManager.default.stopNotificationReachability()

}

func applicationWillEnterForeground(_ application: UIApplication) {

    MerlinManager.default.startNotificationReachability()

}

func applicationDidBecomeActive(_ application: UIApplication) {

    Messaging.messaging().shouldEstablishDirectChannel = true

}

func applicationWillTerminate(_ application: UIApplication) {

    MerlinManager.default.stopNotificationReachability()

}

var applicationStateString: String {

    if UIApplication.shared.applicationState == .active {

        return "active"

    }

}

```

```

    } else if UIApplication.shared.applicationState == .background {
        return "background"
    } else {
        return "inactive"
    }
}
}

extension AppDelegate: UNUserNotificationCenterDelegate {

    func userNotificationCenter(_ center: UNUserNotificationCenter, willPresent
notification: UNNotification, withCompletionHandler completionHandler: @escaping
(UNNotificationPresentationOptions) -> Void) {

        let userInfo = notification.request.content.userInfo

        Logger.i("[UserNotificationCenter] applicationState: \(applicationStateString)
willPresentNotification: \(userInfo)")

        completionHandler([.alert])
    }

    func userNotificationCenter(_ center: UNUserNotificationCenter, didReceive
response: UNNotificationResponse, withCompletionHandler completionHandler: @escaping ()
-> Void) {

        let userInfo = response.notification.request.content.userInfo

        Logger.i("[UserNotificationCenter] applicationState: \(applicationStateString)
didReceiveResponse: \(userInfo)")

        completionHandler()
    }
}

extension AppDelegate: MessagingDelegate {

    func messaging(_ messaging: Messaging, didReceiveRegistrationToken fcmToken:
String) {

        if (MerlinManager.default.getPushType() == .fcm) {

```

Инструкция по интеграции в iOS приложение.

Аналитическая программа с функциями маркетинга и выделения сигналов. MERLIN SDK для iOS.

```

        MerlinManager.default.setPushToken(withToken: fcmToken)
    }
}
}

```

При этом для корректного подключения к Google Firebase необходимо зарегистрировать приложение в Firebase: <https://console.firebase.google.com/?hl=ru&pli=1>

Пока приложение не зарегистрировано в Firebase необходимо закомментировать вызов `FirebaseApp.configure()` чтобы избежать падений.

6 РАБОТА В ФОНОВОМ РЕЖИМЕ

Из-за особенностей платформы iOS приложение в фоновом режиме очень сильно ограничено.

6.1 Работа модуля Мерлин в фоновом режиме

В модуле предусмотрена функция для вызова в моменты пробуждения в фоновом режиме:

```
MerlinManager.default.perfomBackgroundFetch()
```

В п.3 описан пример использования данной функции. Модуль сам определяет временной интервал необходимости сбора информации. Поэтому вызывать эту функцию можно часто, не беспокоясь о частоте сбора информации.

6.2 Механизмы для работы приложения iOS в фоновом режиме

6.2.1 BackgroundFetchUIApplication

Предоставляет возможность вызывать программу в фоновом режиме с определенным интервалом (по умолчанию вызовы не происходят).

Для того что бы установить интервал обновлений в фоновом режиме, нужно написать следующий код:

```
application.setMinimumBackgroundFetchInterval(MerlinManager.default.backgroundFetchInterval)
```

Точнее мы определяем минимальный порог интервала обновления, но iOS не гарантирует что через это время у приложения будет возможность поработать в фоновом режиме.

6.2.2 Сервис Google Firebase

Сервер модуля Мерлин работает с push-уведомлениями. Сервис Google Firebase предлагает возможность отправки push-уведомлений. Сервер Google Firebase отправляет push-уведомления на сервис Apple, а тот в свою очередь на конечное устройство iOS, которое вызывает приложение. Таким образом программа работает в фоновом режиме, получая push-уведомления в функции UIApplication:

```
func application(_ application: UIApplication, didReceiveRemoteNotification userInfo: [AnyHashable : Any], fetchCompletionHandler completionHandler: @escaping (UIBackgroundFetchResult) -> Swift.Void)
```

6.2.3 NotificationCenter

NotificationCenter представляет возможность подписываться на обновления по определенной тематике.

Данный механизм не позволяет запустить в фоне приложение по возникновению события. Механизм просто накапливает события и передает их в приложение после того как приложение выходит из фонового режима.

Инструкция по интеграции в iOS приложение.

Аналитическая программа с функциями маркетинга и выделения сигналов. MERLIN SDK для iOS.

6.3 Ограниченные типы приложений которым разрешается работать в фоновом режим в iOS

Есть ограниченные типы приложений которым разрешается работать в фоновом режим в iOS. По возможности стоит определить свое приложение как одно из предложенных ниже:

- Приложения для проигрывания музыки
- Приложения для записи аудио
- Приложения-навигаторы
- VoIP приложения
- Приложения, получающие информацию с внешних аксессуаров

Если приложение соответствует одному из этих типов то события NotificationCenter будут вызываться сразу даже в фоновом режиме.

Инструкция по интеграции в iOS приложение.

Аналитическая программа с функциями маркетинга и выделения сигналов. MERLIN SDK для iOS.

7 ПОДКЛЮЧЕНИЕ К СОЦИАЛЬНЫМ СЕТЯМ

1. Для подключения к соц. сетям необходимо зарегистрировать приложение на сайте каждой из соц. сети

- facebook: <https://developers.facebook.com/apps/>
- vkontakte: <https://vk.com/apps?act=manage>
- odnoklassniki: <https://ok.ru/vitrine/myuploaded>

2. После регистрации надо прописать некоторые параметры в info.plist

Facebook (<https://www.facebook.com/>)

```
<key>LSApplicationQueriesSchemes</key>
<array>
  <string>fbapi</string>
  <string>fb-messenger-share-api</string>
  <string>fbauth2</string>
  <string>fbshareextension</string>
</array>

<key>CFBundleURLTypes</key>
<array>
  <dict>
    <key>CFBundleURLSchemes</key>
    <array>
      <string>fb{your-app-id}</string>
    </array>
  </dict>
</array>

<key>FacebookAppID</key>
<string>{your-app-id}</string>

<key>FacebookDisplayName</key>
<string>{your-app-name}</string>
```

Инструкция по интеграции в iOS приложение.

Аналитическая программа с функциями маркетинга и выделения сигналов. MERLIN SDK для iOS.

Vkontakte (<https://vk.com>)

```
<key>LSApplicationQueriesSchemes</key>
<array>
  <string>vk</string>
  <string>vk-share</string>
  <string>vkauthorize</string>
  <string>vk{your-app-id}</string>
</array>

<key>CFBundleURLTypes</key>
<array>
  <dict>
    <key>CFBundleTypeRole</key>
    <string>Editor</string>
    <key>CFBundleURLName</key>
    <string>vk{your-app-id}</string>
    <key>CFBundleURLSchemes</key>
    <array>
      <string>vk{your-app-id}</string>
    </array>
  </dict>
</array>
```

Odnoklassniki (<https://ok.ru/>)

```
<key>LSApplicationQueriesSchemes</key>
<array>
  <string>okauth</string>
  <string>ok{your-app-id}</string>
</array>
```

Инструкция по интеграции в iOS приложение.

Аналитическая программа с функциями маркетинга и выделения сигналов. MERLIN SDK для iOS.

```
<key>CFBundleURLTypes</key>
<array>
  <dict>
    <key>CFBundleTypeRole</key>
    <string>Viewer</string>
    <key>CFBundleURLName</key>
    <string>ru.ok</string>
    <key>CFBundleURLSchemes</key>
    <array>
      <string>ok{your-app-id}</string>
    </array>
  </dict>
</array>

<key>NSAppTransportSecurity</key>
<dict>
  <key>NSAllowsArbitraryLoads</key>
  <true/>
</dict>
```

3. Необходимо добавить инициализацию в app delegate

```
class AppDelegate: UIResponder, UIApplicationDelegate
{
  //...

  func application(_ application: UIApplication, didFinishLaunchingWithOptions
launchOptions: [UIApplicationLaunchOptionsKey: Any]?) -> Bool {
  //...

  MerlinManager.default.initVkSdk(vkAppId: "vkAppId")
```

Инструкция по интеграции в iOS приложение.

Аналитическая программа с функциями маркетинга и выделения сигналов. MERLIN SDK для iOS.

```

        MerlinManager.default.initOkSdk(okAppId: "okAppId", okAppKey: "okAppKey")

        MerlinManager.default.initFbSdk(application, didFinishLaunchingWithOptions:
launchOptions)

        MerlinManager.default.startNotificationReachability()

        return true
    }
}

```

4. Подключение к соц сетям.

Модуль поддерживает быструю аутентификацию. То есть если пользователь зашел уже на свою страницу в соц сети в приложении или в браузере, то в модуле авторизация пройдет по этим данным и у пользователя не будет запрошены данные о его странице в соц сети.

Управление соц. сетями происходит по отдельным протоколам.

`SocialControllerDelegateProtocol` – протокол для аутентификации в соц сетях.

```

public protocol SocialControllerDelegateProtocol {

    // Функция для авторизации соц-сетей

    func doAuth(_ data: TypeAuthSocialServer)

    // Функция для проверки - авторизована ли соц-сеть

    func isAuthenticated(type typeSocialServer: TypeSocialServer) -> Bool

    // Функция для деавторизации из соц-сети (по одной соц-сети)

    func doLogout(type typeSocialServer: TypeSocialServer)

    // Функция для деавторизации из соц-сетей (по списку)

    func doLogout(types typeSocialServerList: [TypeSocialServer])

}

```

`TypeSocialServer` – перечень соц. сетей

```

public enum TypeSocialServer {

    // Odnoklassniki (https://ok.ru/)

    case ok

    // Vkontakte (https://vk.com)

    case vk
}

```

Инструкция по интеграции в iOS приложение.

Аналитическая программа с функциями маркетинга и выделения сигналов. MERLIN SDK для iOS.

```
// Facebook (https://www.facebook.com/)

case fb

}
```

TypeAuthSocialServer – перечень параметров для авторизации соц-сети в соответствии с соц-сетью (на данный момент параметры одинаковые)

```
public enum TypeAuthSocialServer {

    // Odnoklassniki (https://ok.ru/)

    case ok(controller: UIViewController, showSuccess: (Any?)->Void, showError:
(Error?) -> Void)

    // Vkontakte (https://vk.com)

    case vk(controller: UIViewController, showSuccess: (Any?)->Void, showError:
(Error?) -> Void)

    // Facebook (https://www.facebook.com/)

    case fb(controller: UIViewController, showSuccess: (Any?)->Void, showError:
(Error?) -> Void)

}
```

SocialNetworkProfileDelegateProtocol – протокол для получение данных из соц. сетей.

```
public protocol SocialNetworkProfileDelegateProtocol {

    // Функция для запуска сбора информации из определенной соц-сети

    func sendProfile(for socialToSend: SocialSend)

}
```

SocialSend – перечень параметров для запроса по сбору данных в рамках одной соц. сети

```
public enum SocialSend {

    // Odnoklassniki (https://ok.ru/)

    case ok((Any?) -> Void, (Error?) -> Void)

    // Vkontakte (https://vk.com)

    case vk((Any?) -> Void, (Error?) -> Void)

    // Facebook (https://www.facebook.com/)

    case fb((Any?) -> Void, (Error?) -> Void)

}
```

```
}
```

7.1 Пример: Аутентификация и запрос на получение данных для каждой соц. сети (если нужна аутентификация)

```
import UIKit
import MerlinForIOS
import SwiftyJSON

class UserController: UIViewController {

    @IBAction func logOutAllButtonPressed(_ sender: Any) {
        delegate.doLogout(types: [.ok, .vk, .fb])
    }

    var delegate: SocialControllerDelegateProtocol!
    var sender: SocialNetworkProfileDelegateProtocol!

    override func viewDidLoad() {
        super.viewDidLoad()

        delegate = SocialControllerDelegate()
        sender = SocialNetworkProfileDelegate()
    }

    func showError(_ errorValue: Error) {
        Alert.showAlert(controller: self, title: "Error", message: String(describing: errorValue))
    }

    func showSucces(_ dataValue: Any) {
        let str = dataValue is Data || dataValue is [String:String] || dataValue is [String:Any] ? JSON(dataValue).description : String(describing: dataValue)

        Alert.showAlert(controller: self, title: "Succes", message: str)
    }
}
```



```

    }

    func showError(_ error: Error?){
        guard let errorValue = error else {
            return
        }
        showError(errorValue)
    }

    func showSucces(_ data: Any?){
        guard let dataValue = data else {
            return
        }
        showSucces(dataValue)
    }

    @IBAction func clickAuthFb(_ sender: Any) {
        delegate.doAuth(.fb(controller: self,
                               showSuccess: { [unowned self] _ in
                                               self.sender.sendProfile(for: .fb(self.showSucces,
                                               self.showError))
                                               },
                               showError: showError))
    }

    @IBAction func clickAuthVk(_ sender: Any) {
        delegate.doAuth(.vk(controller: self,
                               showSuccess: { [unowned self] _ in
                                               self.sender.sendProfile(for: .vk(self.showSucces,
                                               self.showError))
                                               },
                               showError: showError))
    }

```

Инструкция по интеграции в iOS приложение.

Аналитическая программа с функциями маркетинга и выделения сигналов. MERLIN SDK для iOS.

```

        showError: showError))
    }

```

```

@IBAction func clickAuthOk(_ sender: Any) {
    delegate.doAuth(.ok(controller: self,
        showError: { [unowned self] _ in
            self.sender.sendProfile(for: .ok(self.showSuccess,
self.showError))
        },
        showError: showError))
}
}

```

7.2 Пример: Только запрос на получение данных для каждой соц. сети (быстрая аутентификация)

```

import UIKit
import MerlinForIOS
import SwiftyJSON

class UserController: UIViewController {

    var delegate: SocialNetworkProfileDelegateProtocol!

    override func viewDidLoad() {
        super.viewDidLoad()
        delegate = SocialNetworkProfileDelegate()
    }

    func showError(_ error:Error) {
        Alert.showAlert(controller: self, title: "Error", message: String(describing:
error))
    }
}

```

```
}
```

```
func showSucces(_ data:Any) {
    let str = data is Data || data is [String:String] || data is [String:Any] ?
JSON(data).description : String(describing: data)

    Logger.i(str)

    Alert.showAlert(controller: self, title: "Succes", message: str)
}

func showError(_ error:Error?) {
    guard error != nil else { return }
    showError(error!)
}

func showSucces(_ data:Any?) {
    guard data != nil else { return }
    showSucces(data!)
}

@IBAction func sendFbButtonPressed(_ sender: Any) {
    delegate.sendProfile(for: .fb(showSucces, showError))
}

@IBAction func sendVkButtonPressed(_ sender: Any) {
    delegate.sendProfile(for: .vk(showSucces, showError))
}

@IBAction func sendOkButtonPressed(_ sender: Any) {
    delegate.sendProfile(for: .ok(showSucces, showError))
}
```

Инструкция по интеграции в iOS приложение.

Аналитическая программа с функциями маркетинга и выделения сигналов. MERLIN SDK для iOS.

```
}
```

Инструкция по интеграции в iOS приложение.

Аналитическая программа с функциями маркетинга и выделения сигналов. MERLIN SDK для iOS.

8 ДОПОЛНИТЕЛЬНЫЕ РАЗРЕШЕНИЯ, КОТОРЫЕ НАДО ДОБАВИТЬ В РАМКАХ ПРОЕКТА

1. В среде IDE Apple XCode открыть страницу с настройками проекта и переключиться на вкладку Capabilities.
2. Для корректной обработки завершения при удалении программы пользователем нужно добавить:

```
[Background Modes] -> [Remote notifications]
```

3. Для получения и обработки событий когда приложение находится в фоновом режиме нужно добавить:

```
[Push Notifications]

[Background Modes] -> [Remote notifications]

[Background Modes] -> [Background fetch]

[Background Modes] -> [Location updates]
```

4. Через протокол модуля можно определить наличие следующих разрешений:

```
[HealthKit]

[HomeKit]
```