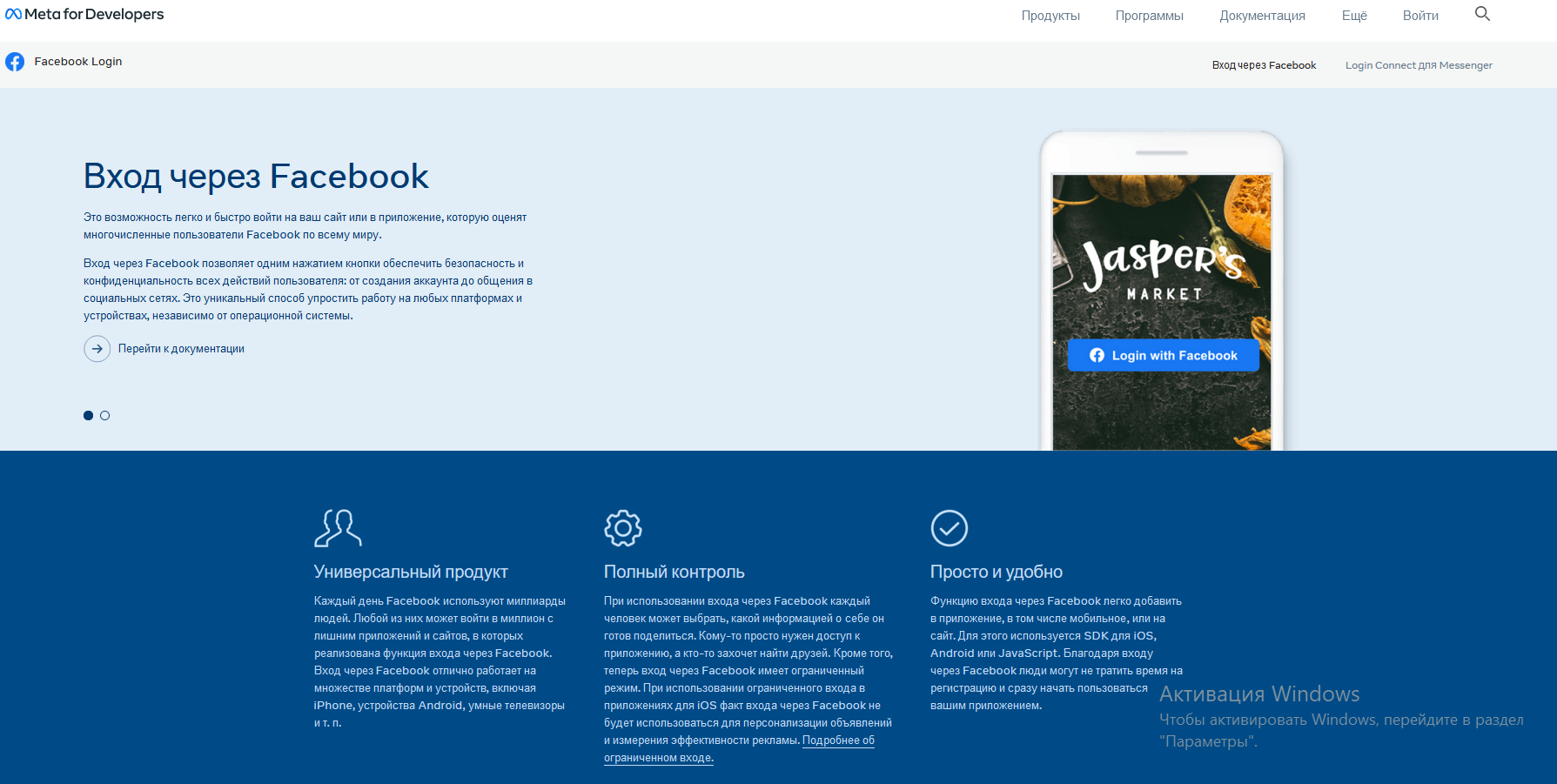
**ЛАБОРАТОРНАЯ РАБОТА №9**АВТОРИЗАЦИЯ ПО ПРОТОКОЛУ OAUTH  
(SPRING BOOT)

**Цель:** описать бизнес-процесс регистрации приложения и реализации протокола OAuth от поставщиков: Facebook.

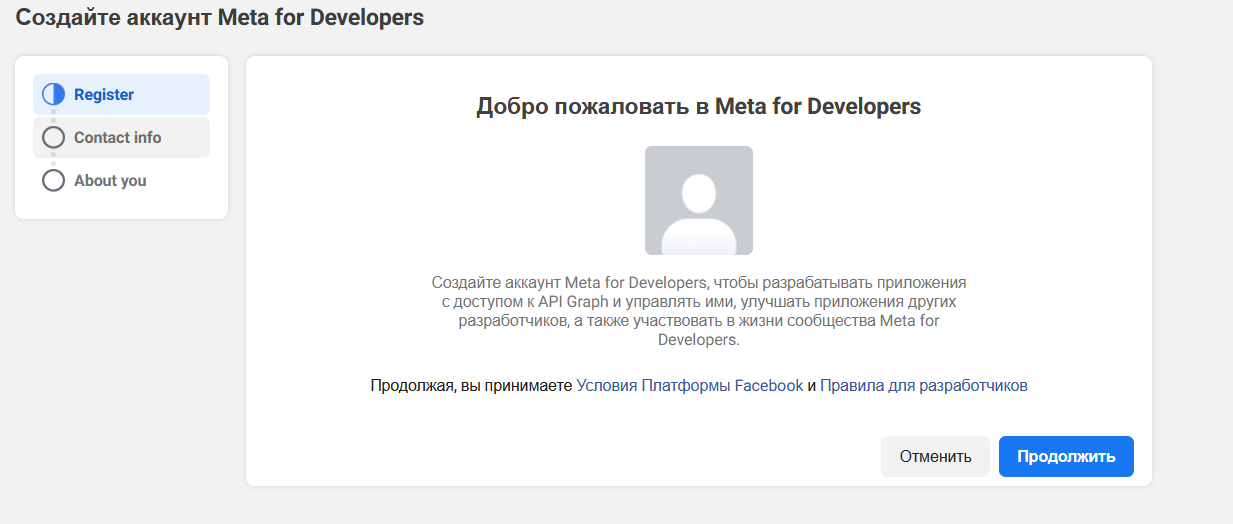
OAuth представляет собой фреймворк для авторизации, позволяющий приложениям осуществлять ограниченный доступ к пользовательским аккаунтам на HTTP сервисах, например, на Facebook, GitHub и DigitalOcean. Он работает по принципу делегирования аутентификации пользователя сервису, на котором находится аккаунт пользователя, позволяя стороннему приложению получать доступ к аккаунту пользователя. OAuth работает в вебе, на десктопных и мобильных приложениях.

**Регистрация OAuth**

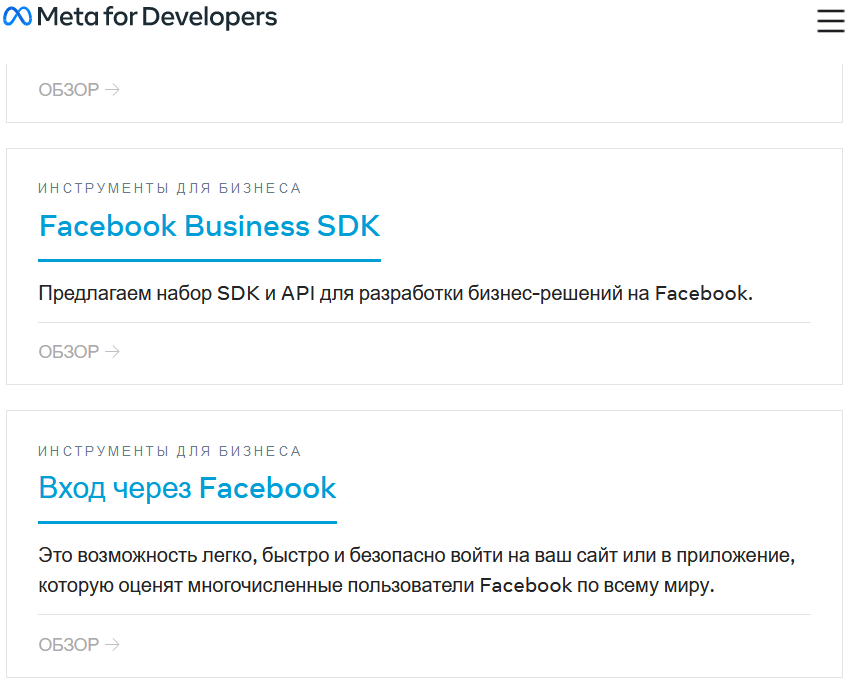
Для того, чтобы приложение могло использовать стороннего поставщика, его необходимо зарегистрировать, в нашем случае поставщиком является Facebook. У Facebook есть специальный сайт Meto for Developers (<https://developer.facebook.com>.) для разработчиков, который представляет приложение для пользователей этих сайтов.



1. Для начала нужно зарегистрировать в качестве разработчика и написать всю необходимую информацию.

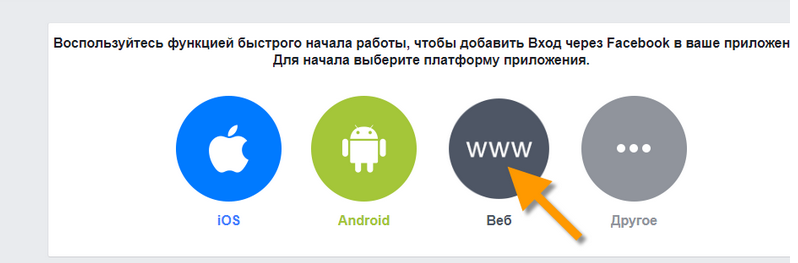


1. Затем во вкладке продукты необходимо выбрать «Вход через Facebook» и нажать кнопку «Начать».





1. Далее можно воспользовать функцией быстрого начала начала работы, чтобы добавить Вход через Facebook в наше приложение. Для начала нужно выбрать платформу приложения.

**Перед началом работы**

Нам понадобятся:

* [аккаунт разработчика Facebook](https://developers.facebook.com/apps/);
* [зарегистрированное приложение Facebook](https://developers.facebook.com/docs/apps#register) с заданными базовыми настройками;
* [Facebook SDK для JavaScript](https://developers.facebook.com/docs/javascript).

## **1. Активация SDK для JavaScript для входа через Facebook.**

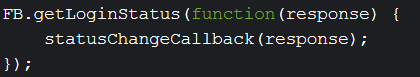
Выберите свое приложение на [панели приложений](https://developers.facebook.com/apps), прокрутите страницу до раздела **Добавить продукт** и нажмите **Настроить** в карточке **Вход через Facebook**. На панели навигации слева выберите **Настройки** и в разделе **Клиентские настройки OAuth** введите URL перенаправления в поле **Действительные URI для перенаправления OAuth** для получения разрешения.

Укажите, что используете SDK для JavaScript для входа, активировав переключатель **Вход с помощью SDK для JavaScript**. Затем в списке **Разрешенные домены для SDK для JavaScript** укажите домен страницы, на которой установлен SDK. Так маркеры доступа будут возвращаться только обратным вызовам из авторизованных доменов. Для аутентификации с помощью SDK для JavaScript поддерживаются только страницы, адрес которых начинается с https.

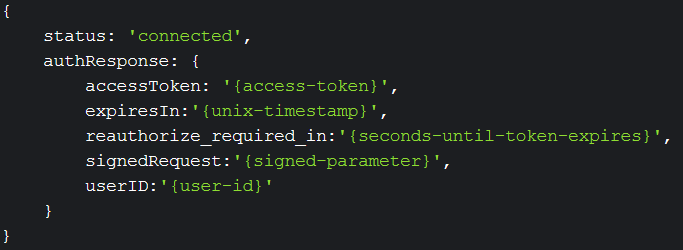
## **2. Проверка статуса входа человека**

При загрузке веб-страницы прежде всего нужно проверить, не вошел ли уже человек на нее с помощью входа через Facebook. Вызов [FB.getLoginStatus](https://developers.facebook.com/docs/reference/javascript/FB.getLoginStatus) инициирует вызов к Facebook для получения статуса входа. Затем Facebook отправляет полученные данные вашей функции обратного вызова.

Пример вызова:



Пример ответа:



Параметр status определяет статус входа этого человека на веб-странице. Возможные значения параметра status:

| **Тип Status** | **Описание** |
| --- | --- |
| connected | Человек вошел в Facebook и на вашу веб-страницу. |
| not\_authorized | Человек вошел в Facebook, но не вошел на веб-страницу. |
| unknown | Человек не вошел в Facebook, поэтому нельзя узнать, вошел ли он на веб-страницу. Другая возможная причина такого статуса: вызов [FB.logout()](https://developers.facebook.com/docs/facebook-login/web#logout) уже был сделан раньше, поэтому подключиться к Facebook не удается. |

Если статус имеет значение connected, в ответ входят следующие параметры authResponse:

| **Параметры authResponse** | **Значение** |
| --- | --- |
| accessToken | Маркер доступа для пользователя веб-страницы. |
| expiresIn | Метка времени (в формате UNIX) для истечения срока действия маркера. Когда срок действия маркера истечет, пользователю придется заново войти. |
| reauthorize\_required\_in | Время до окончания сессии входа в секундах. После этого пользователю потребуется заново войти. |
| signedRequest | Параметр подписи, содержащий сведения о пользователе страницы. |
| userID | ID пользователя страницы. |

SDK для JavaScript автоматически отслеживает статус входа, поэтому делать это вручную не потребуется.

## **3. Вход пользователя**

Если человек открыл вашу страницу, но не вошел на нее либо не вошел в Facebook, с помощью [диалога входа](https://developers.facebook.com/docs/facebook-login/overview/#logindialog) вы можете предложить оба варианта входа. Пользователю сначала будет предложено войти в Facebook, если он этого ещё не сделал, а затем — войти на страницу.

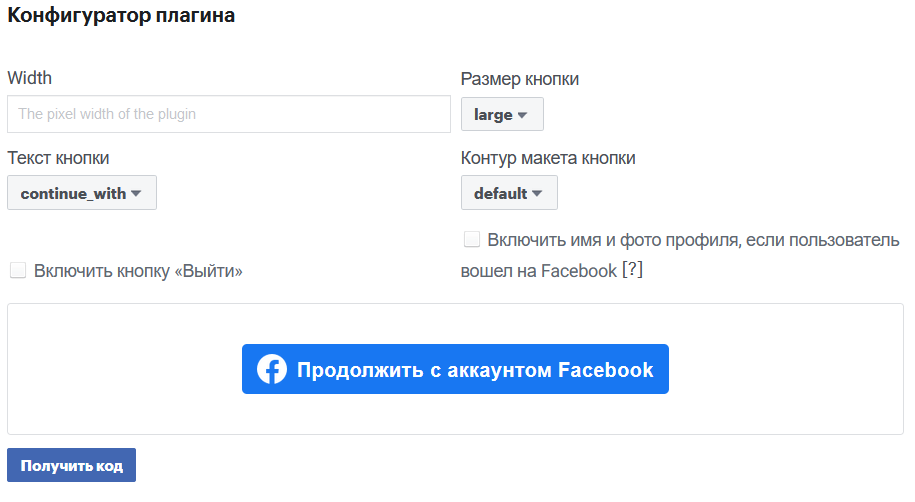
Выполнить вход можно двумя способами:

* [с помощью кнопки входа через Facebook](https://developers.facebook.com/docs/facebook-login/web#loginbutton);
* [с помощью диалога входа](https://developers.facebook.com/docs/facebook-login/web#logindialog) из SDK для JavaScript.

### A. Вход с помощью кнопки "Вход"

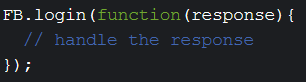
Чтобы использовать кнопку входа через Facebook, [настройте ее](https://developers.facebook.com/docs/facebook-login/web/login-button) с помощью [конфигуратора плагина](https://developers.facebook.com/docs/facebook-login/web/login-button) и получите код.

### **Конфигуратор плагина**



### B. Вход с помощью диалога входа из SDK для JavaScript

Чтобы использовать собственную кнопку, вызывайте диалог входа через [FB.login()](https://developers.facebook.com/docs/reference/javascript/FB.login).

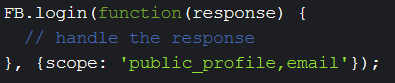


### **Запрос дополнительных разрешений**

Когда пользователь нажимает HTML-кнопку, появляется всплывающее окно диалога входа. Диалог позволяет [запросить разрешение](https://developers.facebook.com/docs/facebook-login/web/permissions) на доступ к данным пользователя. Параметр scope можно передать вместе с вызовом функции FB.login(). Этот необязательный параметр содержит разделенный запятыми список [разрешений](https://developers.facebook.com/docs/facebook-login/permissions), которые пользователь должен подтвердить, чтобы предоставить странице доступ к своим данным. Чтобы вход через Facebook был доступен внешним пользователям, необходима расширенная версия разрешения public\_profile.

#### **Пример вызова**

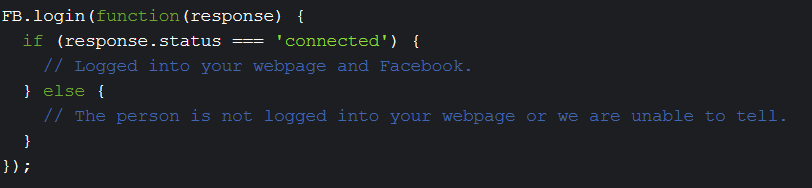
В этом примере у пользователя, выполняющего вход, запрашивается разрешение на доступ страницы к его общедоступному профилю и электронному адресу.



### **Обработка ответа из диалога входа**

Ответ о подключении или отмене возвращает объект authResponse в обратный вызов, который вы указали при совершении вызова FB.login(). Этот ответ можно получить и обработать с помощью FB.login().

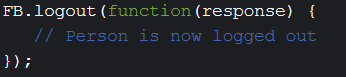
#### **Пример вызова**



## **4. Выход пользователя**

Чтобы обеспечить выход пользователя со страницы, прикрепите к кнопке или ссылке функцию FB.logout() из SDK для JavaScript.

#### **Пример вызова**



**Примечание. Вызов этой функции может также повлечь за собой выход человека из Facebook.**

#### **Возможные сценарии**

1. Человек входит в Facebook, а затем — на вашу страницу. После выхода из приложения он остается в Facebook.
2. Человек входит на страницу и в Facebook, используя процесс входа, реализованный в вашем приложении. При выходе из приложения он также выходит из Facebook.
3. Человек входит на другую (не вашу) страницу и в Facebook, используя процесс входа, реализованный на другой странице, а затем входит на вашу страницу. При выходе с любой из страниц он также выходит из Facebook.

Кроме того, выход с вашей страницы не отменяет разрешения, предоставленные ей пользователем при входе. [Отмену разрешений](https://developers.facebook.com/docs/facebook-login/permissions#revokelogin) нужно выполнить отдельно. Проектируйте страницу таким образом, чтобы вышедший пользователь при повторном входе не перенаправлялся на диалог входа.

## **Пример полного кода**

Этот код загружает SDK для JavaScript на вашу HTML-страницу и инициализирует его. Замените {app-id} своим [ID приложения](https://developers.facebook.com/docs/apps), а {api-version} — требуемой версией API Graph. Если использовать более раннюю версию нет необходимости, укажите последнюю: v15.0.



