**ЛАБОРАТОРНАЯ РАБОТА №9**АВТОРИЗАЦИЯ ПО ПРОТОКОЛУ OAUTH

OAuth — протокол авторизации, позволяющий выдать одному сервису (приложению) права на доступ к ресурсам пользователя на другом сервисе. Протокол избавляет от необходимости доверять приложению логин и пароль, а также позволяет выдавать ограниченный набор прав, а не все сразу.

OAuth основан на использовании базовых веб-технологий: HTTP-запросах, редиректах и т. п. Поэтому использование OAuth возможно на любой платформе с доступом к интернету и браузеру: на сайтах, в мобильных и desktop-приложениях, плагинах для браузеров…

Общая схема работы приложения, использующего OAuth, такова:

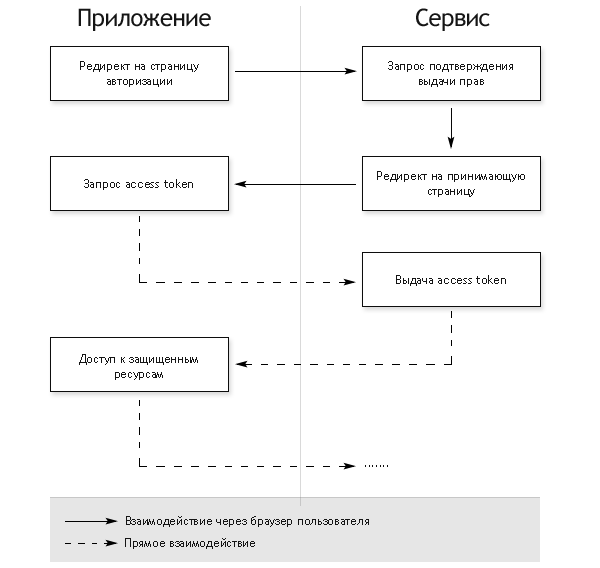
* получение авторизации
* обращение к защищенным ресурсам

Результатом авторизации является access token — некий ключ (обычно просто набор символов), предъявление которого является пропуском к защищенным ресурсам. Обращение к ним в самом простом случае происходит по HTTPS с указанием в заголовках или в качестве одного из параметров полученного access token'а.

В протоколе описано несколько вариантов авторизации, подходящих для различных ситуаций:

* авторизация для приложений, имеющих серверную часть (чаще всего, это сайты и веб-приложения)
* авторизация для полностью клиентских приложений (мобильные и desktop-приложения)
* авторизация по логину и паролю
* восстановление предыдущей авторизации

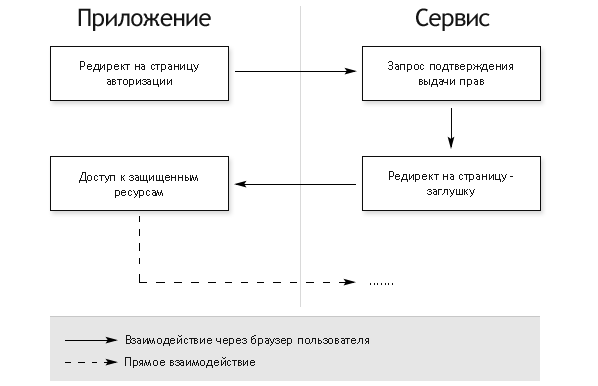
*Авторизация для приложений, имеющих серверную часть*



1. Редирект на страницу авторизации
2. На странице авторизации у пользователя запрашивается подтверждение выдачи прав
3. В случае согласия пользователя, браузер редиректится на URL, указанный при открытии страницы авторизации, с добавлением в GET-параметры специального ключа — authorization code
4. Сервер приложения выполняет POST-запрос с полученным authorization code в качестве параметра. В результате этого запроса возвращается access token

Это самый сложный вариант авторизации, но только он позволяет сервису однозначно установить приложение, обращающееся за авторизацией (это происходит при коммуникации между серверами на последнем шаге). Во всех остальных вариантах авторизация происходит полностью на клиенте и по понятным причинам возможна маскировка одного приложения под другое. Это стоит учитывать при внедрении OAuth-аутентификации в API сервисов.

*Авторизация полностью клиентских приложений*



1. Открытие встроенного браузера со страницей авторизации
2. У пользователя запрашивается подтверждение выдачи прав
3. В случае согласия пользователя, браузер редиректится на страницу-заглушку во фрагменте (после #) URL которой добавляется access token
4. Приложение перехватывает редирект и получает access token из адреса страницы

Этот вариант требует поднятия в приложении окна браузера, но не требует серверной части и дополнительного вызова сервер-сервер для обмена authorization code на access token.

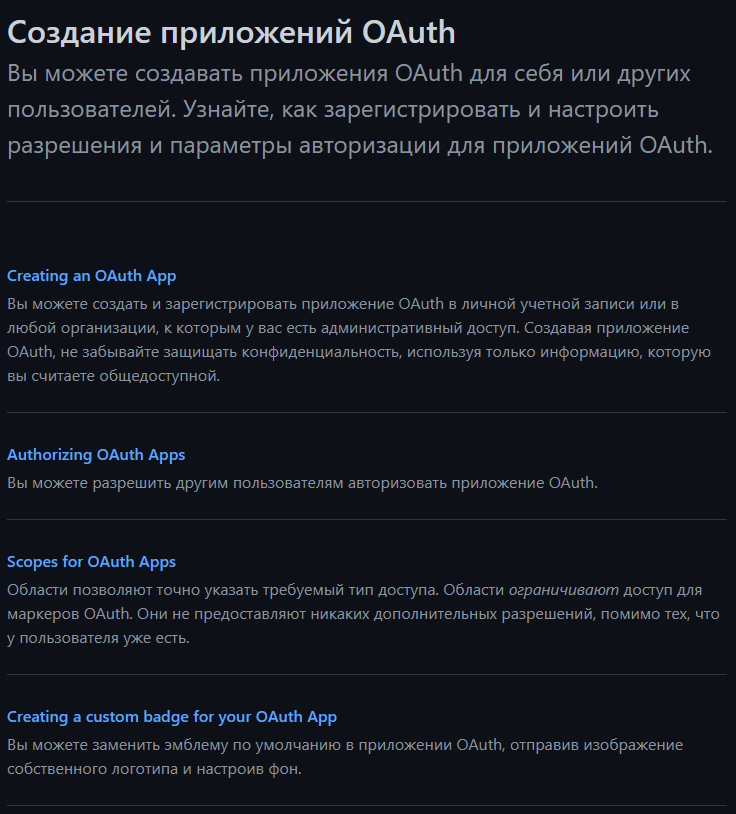
*Авторизация по логину и паролю*

Авторизация по логину и паролю представляет простой POST-запрос, в результате которого возвращается *access token*. Такая схема не представляет из себя ничего нового, но вставлена в стандарт для общности и рекомендуется к применению только, когда другие варианты авторизации не доступны.

*Восстановление предыдущей авторизации*

Обычно, *access token* имеет ограниченный срок годности. Это может быть полезно, например, если он передается по открытым каналам. Чтобы не заставлять пользователя проходить авторизацию после истечения срока действия *access token*'а, во всех перечисленных выше вариантах, в дополнение к *access token*'у может возвращаться еще *refresh token*. По нему можно получить *access token* с помощью HTTP-запроса, аналогично авторизации по логину и паролю.

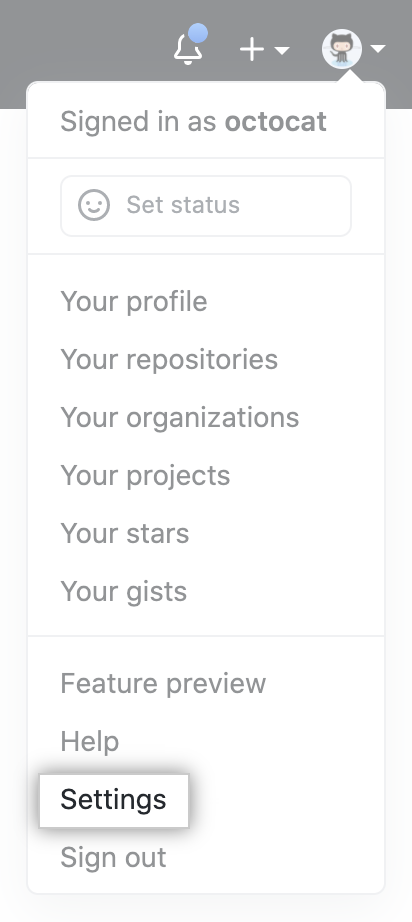
**Бизнес-процесс регистрации приложения и реализации протокола OAuth от GitHub.**

****

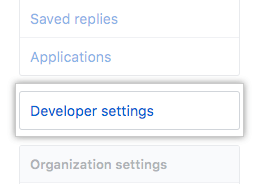
**Рисунок 1 – Страница GitHub с информацией по OAuth**

«Вы можете создать и зарегистрировать приложение OAuth в личной учетной записи или в любой организации, к которым у вас есть административный доступ. Создавая приложение OAuth, не забывайте защищать конфиденциальность, используя только информацию, которую вы считаете общедоступной».

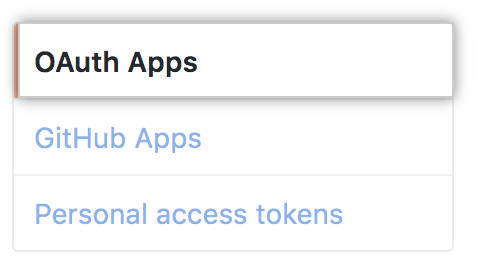
1. В правом верхнем углу любой страницы щелкните фотографию профиля, а затем выберите **Параметры**.



1. In the left sidebar, click **Developer settings**.



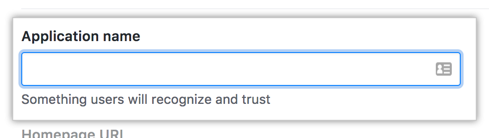
1. На левой боковой панели щелкните **OAuth Apps** .



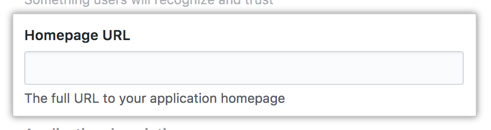
1. Click **New OAuth App**. **Note:** If you haven't created an app before, this button will say, **Register a new application**.



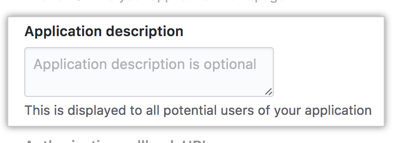
1. In "Application name", type the name of your app. **Warning:** Only use information in your OAuth app that you consider public. Avoid using sensitive data, such as internal URLs, when creating an OAuth App.



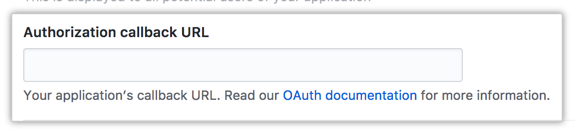
1. In "Homepage URL", type the full URL to your app's website.



1. Optionally, in "Application description", type a description of your app that users will see.



1. In "Authorization callback URL", type the callback URL of your app. **Note:** OAuth Apps cannot have multiple callback URLs, unlike GitHub Apps.



1. Click **Register application**.

