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

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

OAuth 2 – это протокол авторизации, предназначенный для организации доступа клиентских приложений к ресурсам, или данным учетных записей, пользователя на другом сервисе. В качестве клиентских приложений выступают веб-сервисы, мобильные и десктопные приложения. В качестве сервисов – mail.ru, GitHub, Bitbucket и др. Протокол используют разработчики сторонних приложений.

Мы сталкиваемся с этим протоколом, когда:

1.авторизуемся на сторонних площадках через аккаунты соцсетей;

2.устанавливаем себе на мобильное устройство приложение, взаимодействующее с нашими данными в облачных сервисах типа Google или Яндекс;

3.используем сторонние приложения (боты в Telegram и других мессенджерах) для уведомлений и пр.

Доступ может быть ограничен правами пользователя или же областями видимости, что повышает гибкость использования протокола. Например, стороннее приложение может только читать наши данные, а не изменять их, либо же только изменять.

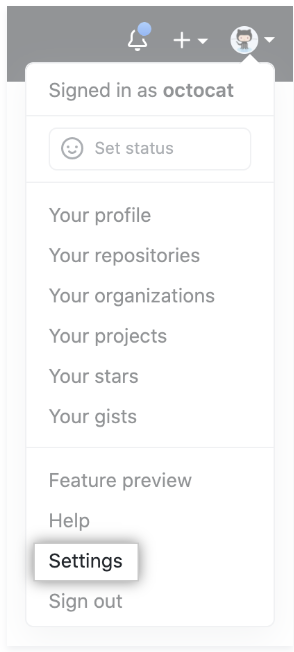
**Создание приложений OAuth**

***Creating an OAuth App***

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.

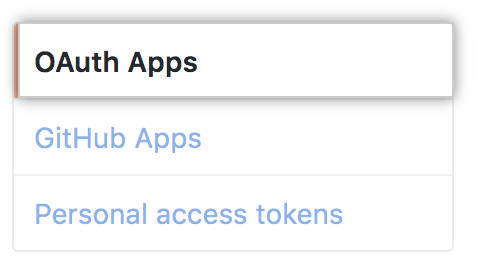
**Note**: Пользователю или организации могут принадлежать до 100 приложений OAuth.

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



2. In the left sidebar, click < > **Developer** **settings**.

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

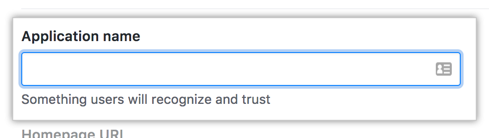


4. Нажмите **New OAuth App**.



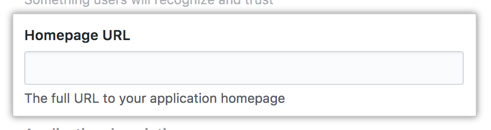
Note: Если вы еще не создали приложение, на этой кнопке будет написано «Зарегистрируйте новое приложение».

5. В "Application name",введите название своего приложения.

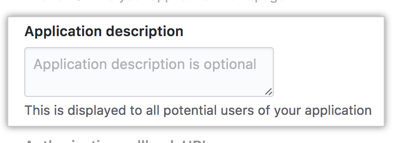


Warning: Используйте в своем приложении OAuth только ту информацию, которую вы считаете общедоступной. Избегайте использования конфиденциальных данных, таких как внутренние URL-адреса, при создании приложения OAuth.

6. В "Homepage URL", введите полный URL вашего сайта

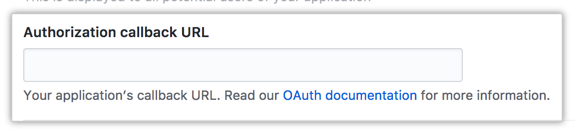


7. При желании в поле «Описание приложения» введите описание вашего приложения, которое увидят пользователи.



8. В "Authorization callback URL", введите URL-адрес обратного вызова вашего приложения

.

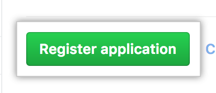


Note: Приложения OAuth не могут иметь несколько URL-адресов обратного вызова, в отличие от приложений GitHub

9. Если ваше приложение OAuth будет использовать поток устройств для идентификации и авторизации пользователей, нажмите «Включить поток устройств». Дополнительные сведения о потоке устройств см. в разделе «Авторизация приложений OAuth».

Screenshot showing field for enabling device flow

10. Нажмите **Register** **application**.



Вы можете разрешить другим пользователям авторизовать приложение OAuth.

Реализация OAuth в GitHub поддерживает стандартный тип гранта кода авторизации и грант авторизации устройства OAuth 2.0 для приложений, у которых нет доступа к веб-браузеру.

Если вы хотите пропустить авторизацию своего приложения стандартным способом, например, при тестировании своего приложения, вы можете использовать поток не-веб-приложения.

Чтобы авторизовать свое приложение OAuth, подумайте, какой поток авторизации лучше всего подходит для вашего приложения.

– поток веб-приложений: используется для авторизации пользователей для стандартных приложений OAuth, которые запускаются в браузере. (Неявный тип предоставления не поддерживается.)

– поток устройства: используется для автономных приложений, таких как инструменты CLI.

Поток веб-приложений

Примечание. Если вы создаете приложение GitHub, вы по-прежнему можете использовать поток веб-приложения OAuth, но в настройке есть некоторые важные отличия. Дополнительные сведения см. в разделе «Идентификация и авторизация пользователей для приложений GitHub».

Поток веб-приложения для авторизации пользователей для вашего приложения:

1. Пользователи перенаправляются для запроса своей личности GitHub.

2. Пользователи перенаправляются обратно на ваш сайт с помощью GitHub.

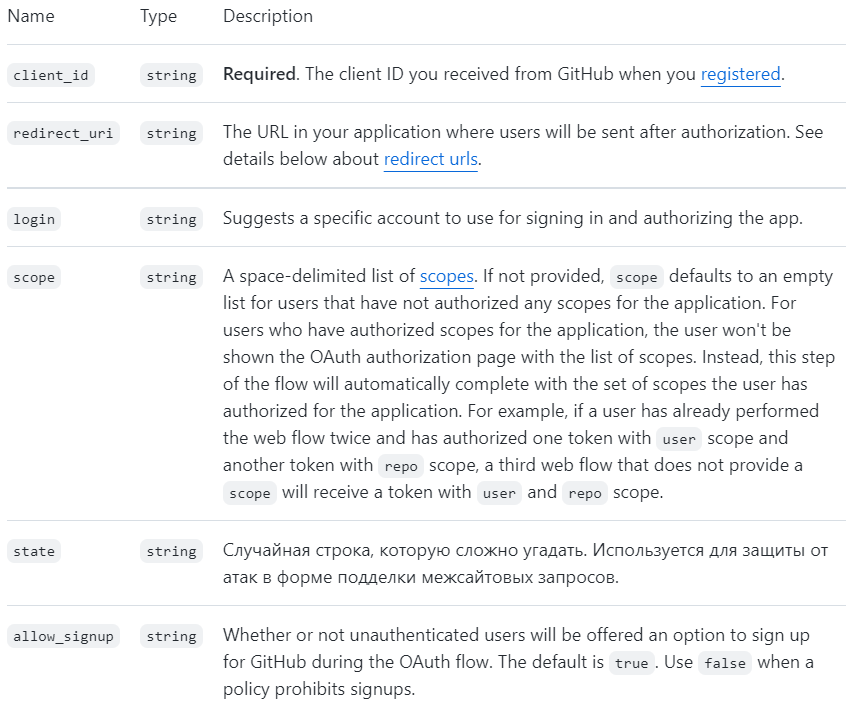
3. Ваше приложение обращается к API с токеном доступа пользователя.

***Authorizing OAuth Apps***

*1. Request a user's GitHub identity*

GET https://github.com/login/oauth/authorize

Когда ваше приложение GitHub указывает параметр login в систему, оно предлагает пользователям конкретную учетную запись, которую они могут использовать для входа и авторизации вашего приложения.**Parameters**



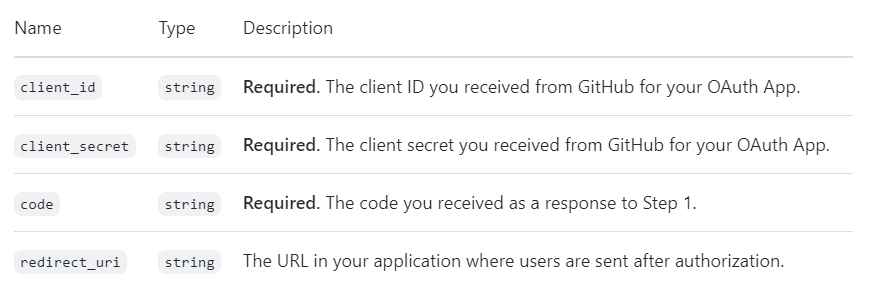
*2. Users are redirected back to your site by GitHub*

Если пользователь принимает ваш запрос, GitHub перенаправляет обратно на ваш сайт с временным кодом в параметре кода, а также с состоянием, которое вы указали на предыдущем шаге в параметре состояния. Срок действия временного кода истекает через 10 минут. Если состояния не совпадают, то запрос создан третьей стороной, и вам следует прервать процесс.

Exchange this code for an access token:

POST https://github.com/login/oauth/access\_token

**Parameters**



**Response**

By default, the response takes the following form:

access\_token=gho\_16C7e42F292c6912E7710c838347Ae178B4a&scope=repo%2Cgist&token\_type=bearer

Вы также можете получить ответ в разных форматах, указав формат в заголовке Accept. Например, Accept: application/json или Accept: application/xml:

Accept: application/json

{

"access\_token":"gho\_16C7e42F292c6912E7710c838347Ae178B4a",

"scope":"repo,gist",

"token\_type":"bearer"

}

Accept: application/xml

<OAuth>

<token\_type>bearer</token\_type>

<scope>repo,gist</scope>

<access\_token>gho\_16C7e42F292c6912E7710c838347Ae178B4a</access\_token>

</OAuth>

*3. Use the access token to access the API*

The access token allows you to make requests to the API on a behalf of a user.

Authorization: Bearer OAUTH-TOKEN

GET <https://api.github.com/user>

For example, in curl you can set the Authorization header like this:

curl -H "Authorization: Bearer OAUTH-TOKEN" https://api.github.com/user

Device flow

Примечание. Поток устройств находится в общедоступной бета-версии и может быть изменен.

Поток устройства позволяет авторизовать пользователей для автономного приложения, такого как инструмент CLI или диспетчер учетных данных Git.

Прежде чем вы сможете использовать поток устройства для авторизации и идентификации пользователей, вы должны сначала включить его в настройках вашего приложения. Дополнительные сведения о включении потока устройств в вашем приложении см. в разделах «Изменение приложения OAuth» для приложений OAuth и «Изменение приложения GitHub» для приложений GitHub.Overview of the device flow

1. Ваше приложение запрашивает коды подтверждения устройства и пользователя и получает URL-адрес авторизации, по которому пользователь вводит код подтверждения пользователя.

2. Приложение предлагает пользователю ввести код подтверждения пользователя на странице https://github.com/login/device.

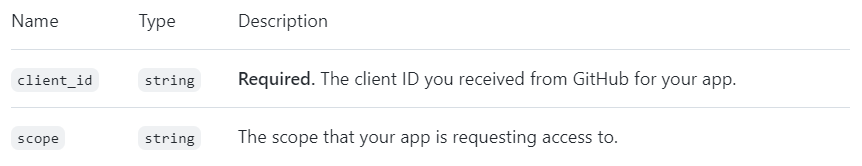
3. Приложение опрашивает статус аутентификации пользователя. Как только пользователь авторизует устройство, приложение сможет выполнять вызовы API с новым токеном доступа..

**Step 1: App requests the device and user verification codes from GitHub**

POST https://github.com/login/device/code

Ваше приложение должно запросить код подтверждения пользователя и URL-адрес подтверждения, которые приложение будет использовать, чтобы предложить пользователю пройти аутентификацию на следующем шаге. Этот запрос также возвращает код проверки устройства, который приложение должно использовать для получения токена доступа и проверки состояния проверки подлинности пользователя.

**Input Parameters**



**Response**

By default, the response takes the following form:

device\_code=3584d83530557fdd1f46af8289938c8ef79f9dc5&expires\_in=900&interval=5&user\_code=WDJB-MJHT&verification\_uri=https%3A%2F%github.com%2Flogin%2Fdevice

Вы также можете получить ответ в разных форматах, указав формат в заголовке Accept. Например, Accept: application/json или Accept: application/xml:

Accept: application/json

{

"device\_code": "3584d83530557fdd1f46af8289938c8ef79f9dc5",

"user\_code": "WDJB-MJHT",

"verification\_uri": "https://github.com/login/device",

"expires\_in": 900,

"interval": 5

}

Accept: application/xml

<OAuth>

<device\_code>3584d83530557fdd1f46af8289938c8ef79f9dc5</device\_code>

<user\_code>WDJB-MJHT</user\_code>

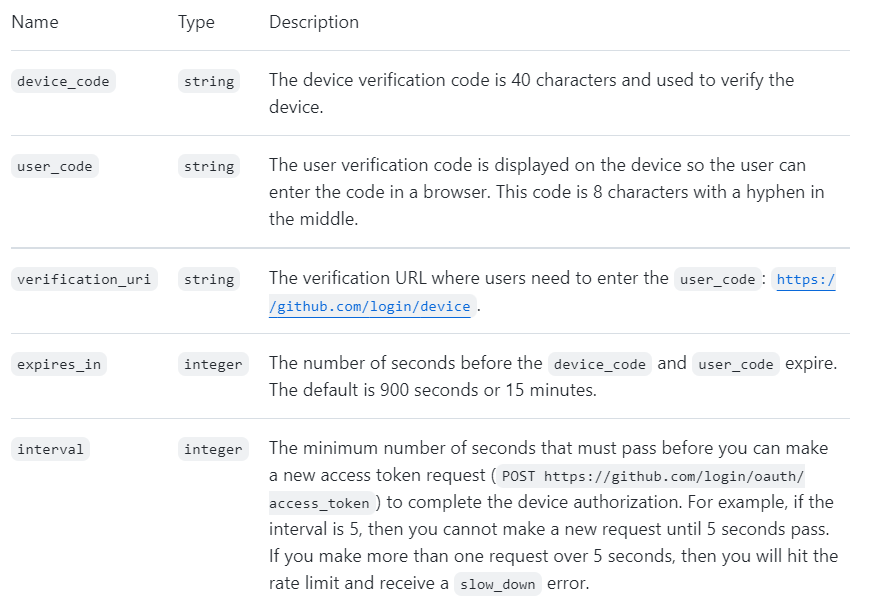
<verification\_uri>https://github.com/login/device</verification\_uri>

<expires\_in>900</expires\_in>

<interval>5</interval>

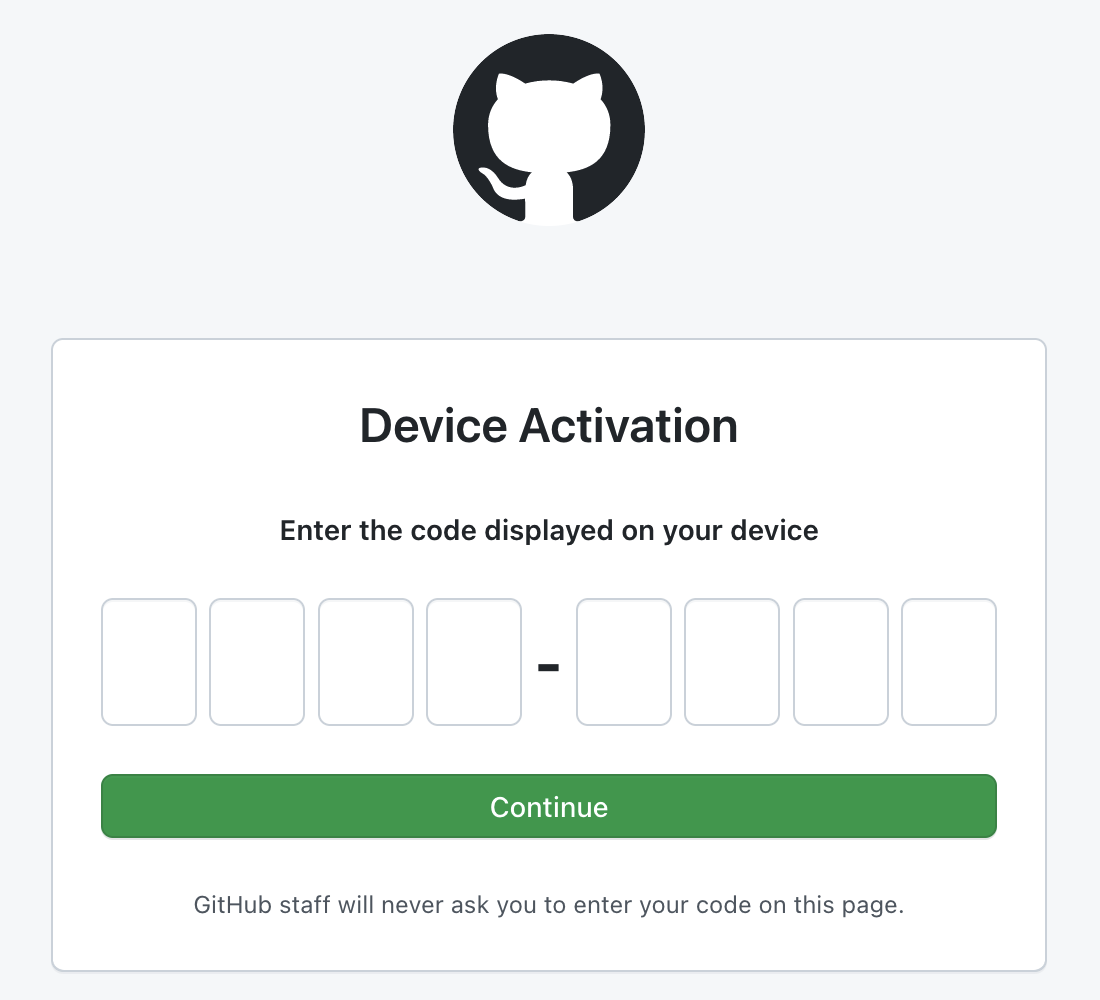
</OAuth>

**Response parameters**



**Step 2: Prompt the user to enter the user code in a browser**

Your device will show the user verification code and prompt the user to enter the code at https://github.com/login/device.



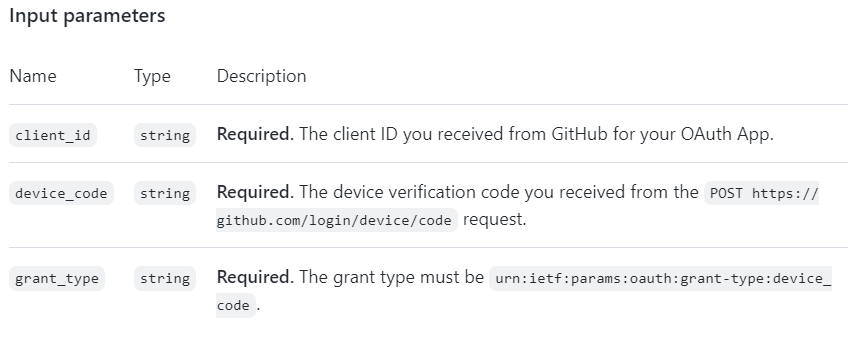
**Step 3: App polls GitHub to check if the user authorized the device**

POST https://github.com/login/oauth/access\_token

Your app will make device authorization requests that poll POST https://github.com/login/oauth/access\_token, until the device and user codes expire or the user has successfully authorized the app with a valid user code. The app must use the minimum polling interval retrieved in step 1 to avoid rate limit errors. For more information, see "Rate limits for the device flow."

The user must enter a valid code within 15 minutes (or 900 seconds). After 15 minutes, you will need to request a new device authorization code with POST https://github.com/login/device/code.

Once the user has authorized, the app will receive an access token that can be used to make requests to the API on behalf of a user.



**Response**

By default, the response takes the following form:

access\_token=gho\_16C7e42F292c6912E7710c838347Ae178B4a&token\_type=bearer&scope=repo%2Cgist

Вы также можете получить ответ в разных форматах, указав формат в заголовке Accept. Например, Accept: application/json или Accept: application/xml:

Accept: application/json

{

"access\_token": "gho\_16C7e42F292c6912E7710c838347Ae178B4a",

"token\_type": "bearer",

"scope": "repo,gist"

}

Accept: application/xml

<OAuth>

<access\_token>gho\_16C7e42F292c6912E7710c838347Ae178B4a</access\_token>

<token\_type>bearer</token\_type>

<scope>gist,repo</scope>

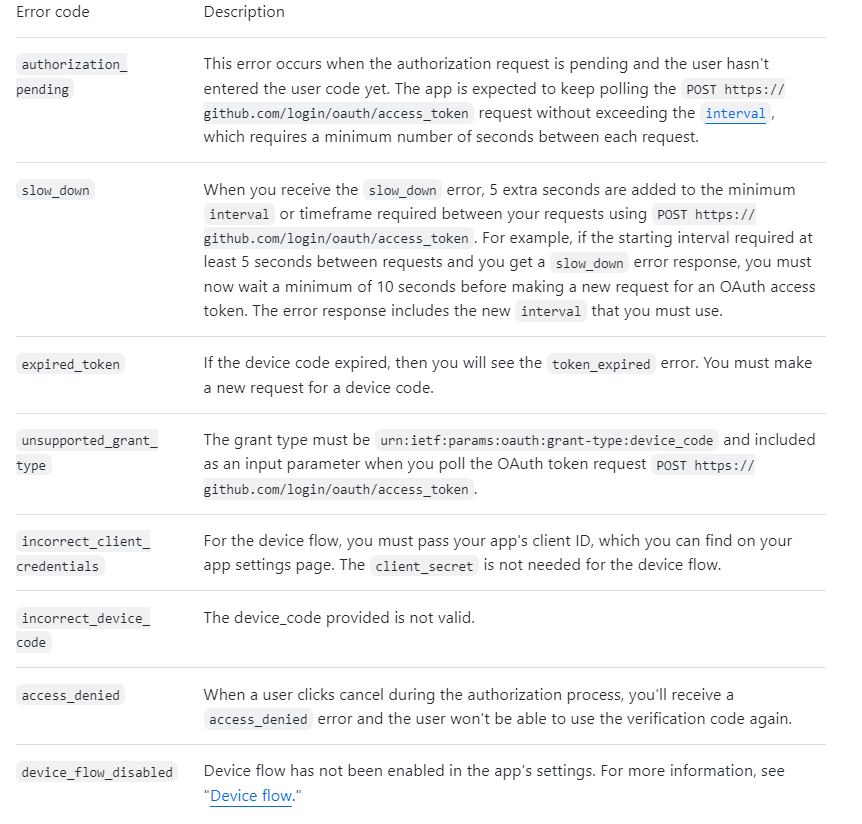
</OAuth>

**Rate limits for the device flow**

When a user submits the verification code on the browser, there is a rate limit of 50 submissions in an hour per application.

If you make more than one access token request (POST https://github.com/login/oauth/access\_token) within the required minimum timeframe between requests (or interval), you'll hit the rate limit and receive a slow\_down error response. The slow\_down error response adds 5 seconds to the last interval. For more information, see the Errors for the device flow.

**Error codes for the device flow**



For more information, see the "OAuth 2.0 Device Authorization Grant."