## Вход через Яндекс в файле PHP, использующем CURL с получением токена

Авторизация с помощью API паспорта Яндекса облегчает вход, авторизацию пользователям вашего сайта. Не требуется вводить логин, пароль, капчу при авторизации, и вообще не нужна регистрация на сайте. На текущей странице моего блога реализован этот механизм. Если вы уже авторизованы на Яндексе, то чтобы войти на сайт, например для написания комментария, потребуется только один клик по значку «Яндекс вход». Если не выполнен вход в аккаунт (учетную запись) Яндекса, то потребуется 3, 4 клика, в зависимости от количества аккаунтов в Яндексе.

## Порядок входа

Если посетитель этого блога не авторизован на ресурсе поисковика Яши, то после клика по значку **входа** механизм получения доступа следующий:

* откроется страница passport.yandex.ru для выбора аккаунтов входа,
* после произведенного выбора появится окно для ввода пароля,
* остается ввести пароль аккаунта яндекса и нажать «войти».

В случае наличия единственного аккаунта у поисковика Яши кликов на один шаг меньше, т.е. только 3. Такой порядок входа значительно уменьшает количество движений у юзера.

Для начала получения этого сервиса на нужных страницах сайта устанавливается ссылка входа.

После клика она отправляет посетителя на OAuth-сервер и имеет следующий вид:

<a href="https://oauth.yandex.ru/authorize?response\_type=code&client\_id=<*ID\_ПРИЛОЖЕНИЯ*>></a>

Про <*ID\_ПРИЛОЖЕНИЯ*> написано в этой статье ниже. Угловые скобки и то что внутри скобок необходимо заменить конкретным значением.

## Реализация входа через Яндекс на этом блоге

На OAuth-сервере API паспорта использует протокол OAuth 2.0. Подробности в документации *yandex.ru/dev/oauth/doc/dg/concepts/about.html*. После пройденных шагов по регистрации сайта, получению доступов вернутся ID ПРИЛОЖЕНИЯ и ПАРОЛЬ ПРИЛОЖЕНИЯ, которые и надо реализовать в скрипте для получения токена, а далее и защищенных данных пользователя.

У меня это реализовано в файле *yandex.php*. Скрипт этого файла отправляется на OAuth-сервер Яндекса POST-запросом. Там он обрабатывается и этот же файл *yandex.php* примет результат обработки.

**<?php** /\*авторизация через Яндекс\*/  
**if** (!$\_GET['code']) {**exit**('error code');}  
session\_start();  
//Инициализируем сеанс  
$curl = curl\_init();  
//Устанавливаем адрес для подключения  
curl\_setopt($curl, *CURLOPT\_URL*, 'https://oauth.yandex.ru/token');  
//Указываем, что мы будем вызывать методом POST  
curl\_setopt($curl, *CURLOPT\_POST*, 1);  
//Передаем параметры методом POST  
curl\_setopt($curl, *CURLOPT\_POSTFIELDS*, 'grant\_type=authorization\_code&code='.$\_GET['code'].'&client\_id=<ID\_ПРИЛОЖЕНИЯ>&client\_secret==<ПАРОЛЬ\_ПРИЛОЖЕНИЯ>');  
//Говорим, что надо возвратить результат  
curl\_setopt($curl, *CURLOPT\_RETURNTRANSFER*, **true**);  
//Запрещаем проверку сертификата удаленного сервера  
curl\_setopt($curl, *CURLOPT\_SSL\_VERIFYPEER*, **false**);  
//Декодируем строку JSON в переменную PHP  
$token = json\_decode(curl\_exec($curl), **true**);  
//Закрываем CURL-соединение  
curl\_close($curl);  
//Получаем данные пользователя  
$data = json\_decode(file\_get\_contents('https://login.yandex.ru/info?oauth\_token='.$token['access\_token']), **true**);  
//Сохраняем их в сессию  
$\_SESSION["name\_ya"] = $data["real\_name"];  
$\_SESSION['login\_ya'] = $data["login"];  
$\_SESSION['email\_ya'] = $data["emails"][0];  
$\_SESSION["avatar\_id"] = $data["default\_avatar\_id"];  
$\_SESSION["avatar\_empty"] = $data["is\_avatar\_empty"];

//Редирект на последнюю страницу  
**if**(**isset**($\_GET['state']))  
{  
 $link = 'http://' . $\_SESSION['ya'] . '#newComment';  
}  
**else** $link = 'http://' . $\_SESSION['ya'];  
  
header("Location: $link");  
**exit**();

Отправляем POST-запрос на адрес «oauth.yandex.ru» и получаем результат с помощью CURL.

### Коротко о CURL

Расширение CURL (Client URL Library) обеспечивает широкие средства управления сетевыми операциями компактным способом. Позволяет взаимодействовать с серверами по различным протоколам с синтаксисом URL. Эта библиотека функций предоставляет возможность посылать запросы из PHP скрипта методами POST, GET, PUT.

Начиная с версии v1803 операционной системы Windows 10, служебная программа уже установлена и настроена. Её наличие проверяется вводом команды “curl -help“ в командную строку. Если ошибок нет, то можно пользоваться curl-ом. Для того чтобы функции библиотеки CURL были доступны из PHP-скрипта, в конфигурационном файле php.ini необходимо подключить расширение php\_curl.dll, сняв комментарий (точка с запятой ;) с директивы *extension*.

Подробное комментирование входа в файле yandex.php

OAuth-сервер Яши производит редирект на адрес Callback URI, указанный нами при регистрации сайта. В моем случае на файл *yandex.php.* При этом к адресу добавляется параметр code (‘code’ - <код\_подтверждения>). Наличие его проверяется в самом начале PHP скрипта *yandex.php*, а затем этот код (*$\_GET['code']*)используется функцией для получения нашего токена.

Начинаем новый сеанс cURL функцией *curl\_init(),* возвращающей дескриптор *$curl*, который используется последующими функциями.

Функция *curl\_setopt()* позволяет задать параметры текущего соединения. Константа *CURLOPT\_URL,* в качестве параметра функции, описывает URL, с которым будет производиться операция. Задаем полный адрес удаленного сервера, именно полный – с префиксом https://, так как расширение CURL работает с несколькими видами протоколов. Нашему скрипту необходимо выполнить POST-запрос на [*https://oauth.yandex.ru/token*](https://oauth.yandex.ru/token).

Туда передаем параметры. Для этого в функции *curl\_setopt()* используется константа *CURLOPT\_POSTFIELDS* (данные, передаваемые в HTTP POST-запросе). Её значение - *'grant\_type=authorization\_code&code='.$\_GET['code'].'&client\_id=<ID\_ПРИЛОЖЕНИЯ>&client\_secret=<* *ПАРОЛЬ\_ПРИЛОЖЕНИЯ>'*.

ID и ПАРОЛЬ ПРИЛОЖЕНИЯ – из зарегистрированного на OAuth-сервере нашего сайта. Их нам сгенерировал и затем предоставил Яша после регистрации сайта.

Подобным образом устанавливаем параметры о том, что результат надо возвратить, у удаленного сервера сертификат не проверять.

После установки всех необходимых параметров при помощи функции *curl\_exec($curl)* выполняется запрос к удаленному серверу, и затем производим декодирование полученной JSON строки в переменную PHP *$token*.

Ключ *[‘access\_token’]* массива *$token* как раз и содержит значение нашего искомого OAuth-токен.

Завершаем сеанс cURL функцией *curl\_close()*.

### Токен от Яндекса

Далее работаем со значением ключа *[‘access\_token’].* Это и есть токен, предоставляющий доступ к защищённым данным пользователя, который на нашем сайте выполнил вход для авторизации через Яндекс без необходимости передачи логина и пароля.

OAuth-токен требуется указывать при выполнении запроса к API. Информация о пользователе, о приложении заложена в токене. Токен информирует, от какого приложения (в нашем случае – сайта), от какого пользователя выполняется запрос, разрешил ли пользователь этому сайту доступ к своим данным в Паспорте, имеет ли пользователь полномочия для обращения к этим материалам.

Для каждого пользователя отправляется свой отдельный токен.

Таким образом, для работы с API Паспорта нет необходимости отправлять пароль пользователя, достаточно отправить его OAuth-токен.

Нельзя никому сообщать значение токена, или давать возможность его получения. С его помощью злоумышленник получает доступ к секретным материалам юзера.

### Работа с данными

Теперь в переменную *$data*, используя ключ *[‘access\_token’]* массива *$token*, возвращаем ассоциативный массив с материалами о пользователе.

Потребуется обратиться к документации на странице [*yandex.ru/dev/passport/doc/dg/reference/response.html*](https://yandex.ru/dev/passport/doc/dg/reference/response.html), чтобы расшифровать формат ответа и выяснить, как извлечь имя, логин, адрес почты, аватарку и т.п. пользователя.

Требуемые для дальнейшей работы данные сохраняем в сессию. Эти данные потребуются для ввода в скрипты на страницах сайта.

### Редирект на страницы блога

Прошу обратить внимание, что API позволяет в URL, указанный при регистрации приложения, как Callback URL, добавлять после символа # некоторые данные о токене. Именно на этот URL Яндекс.Oauth перенаправляет пользователя после обработки токена. После # можно добавить параметр *‘state’* со значением строки. Это значение в адресной строке возвращается без изменения. Я его (*$\_GET['state']*) использую для **редиректа с помощью заголовка Location**, перенаправляя посетителя блога на конкретное место страницы, откуда он произвел авторизацию через Яндекс. Это улучшает юзабилите настоящего блога. Осуществляется путем установки якоря.

Ссылка редиректа содержит элемент указателя *$\_SESSION['ya']*. Значение его необходимо записывать в сессию на страницах сайта следующим образом:

<?php

session\_start();

$\_SESSION['ya'] = $\_SERVER['HTTP\_HOST'] . $\_SERVER['REQUEST\_URI'];

?>

Такой редирект работает надежно.

## Итог

Вот и все. Скрипт авторизации через Яндекс небольшой, а комментарий получился обширный. Итог файла используется в методах классов, функциях настоящего блога, реализующих вход.

Описанный способ авторизации предоставляет преимущества не только посетителям ресурса, но и облегчает работу разработчику как готовая и бесплатная технология.

Ссылка на этот якорь <a href=/article.php?id=10#redir">Редирект с помощью заголовка Location</a>

**<?php** /\*авторизация через Яндекс\*/  
**if** (!$\_GET['code']) {**exit**('error code');}  
session\_start();  
//Инициализируем сеанс  
$curl = curl\_init();  
//Устанавливаем адрес для подключения  
curl\_setopt($curl, *CURLOPT\_URL*, 'https://oauth.yandex.ru/token');  
//Указываем, что мы будем вызывать методом POST  
curl\_setopt($curl, *CURLOPT\_POST*, 1);  
//Передаем параметры методом POST  
curl\_setopt($curl, *CURLOPT\_POSTFIELDS*, 'grant\_type=authorization\_code&code='.$\_GET['code'].'&client\_id=<ID\_ПРИЛОЖЕНИЯ>&client\_secret==<ПАРОЛЬ\_ПРИЛОЖЕНИЯ>');  
//Говорим, что надо возвратить результат  
curl\_setopt($curl, *CURLOPT\_RETURNTRANSFER*, **true**);  
//Запрещаем проверку сертификата удаленного сервера  
curl\_setopt($curl, *CURLOPT\_SSL\_VERIFYPEER*, **false**);  
//Декодируем строку JSON в переменную PHP  
$token = json\_decode(curl\_exec($curl), **true**);  
//Закрываем CURL-соединение  
curl\_close($curl);  
//Получаем данные пользователя  
$data = json\_decode(file\_get\_contents('https://login.yandex.ru/info?oauth\_token='.$token['access\_token']), **true**);  
//Сохраняем их в сессию  
$\_SESSION["name\_ya"] = $data["real\_name"];  
$\_SESSION['login\_ya'] = $data["login"];  
$\_SESSION['email\_ya'] = $data["emails"][0];  
$\_SESSION["avatar\_id"] = $data["default\_avatar\_id"];  
$\_SESSION["avatar\_empty"] = $data["is\_avatar\_empty"];

//Редирект на последнюю страницу  
**if**(**isset**($\_GET['state']))  
{  
 $link = 'http://' . $\_SESSION['ya'] . '#newComment';  
}  
**else** $link = 'http://' . $\_SESSION['ya'];  
  
header("Location: $link");  
**exit**();

90% со скриптом, 95% без скрипта

яндекс, вход, curl, токен, регистрации сайта, авторизации яндекс, входа яндекс, php