

УДК 004.77

Студ. А. Н. Гусакова, студ. А. А. Яговдик
Науч. рук. доц. Д. М. Романенко, асс. С. А. Осоко
(кафедра информатики и веб-дизайна)

ОСОБЕННОСТИ РЕГИСТРАЦИИ И АВТОРИЗАЦИИ С ИСПОЛЬЗОВАНИЕМ СОЦИАЛЬНЫХ СЕТЕЙ НА ПЛАТФОРМЕ РНР

Авторизация через социальные сети упрощает процесс регистрации всего до пары кликов.

В настоящее время существует более 30 ресурсов, которые можно использовать для входа на сайт. Разные ресурсы содержат разные данные о пользователе. Поэтому владельцам сайтов необходимо определять для себя, какая информация о пользователях для них важна. И уже в соответствии с этим выбирать ресурсы, с помощью которых будет возможна авторизация на их сайтах.

В данной работе показано, как можно добавить авторизацию и регистрацию через социальную сеть «ВКонтакте» на платформе РНР. Была выбрана именно эта социальная сеть, т.к. в нашем регионе она является наиболее популярной сетью у пользователей Интернета.

На основании проведенного анализа литературных источников [1] для добавления авторизации и регистрации необходимо действовать по следующему алгоритму.

Добавление будет происходить в 6 этапов:

1. Регистрация нового приложения
2. Генерация ссылки для аутентификации
3. Получение токена
4. Получение информации о пользователе
5. Извлечение информации о пользователе
6. Регистрация на сайте или авторизация

Этап 1. Регистрация нового приложения

Для начала необходимо создать приложение на сайте социальной сети «ВКонтакте»

В открывшейся форме (рисунок 1) необходимо ввести название приложения; выбрать тип “Веб-сайт”; В качестве адреса сайта ввести путь к папке с проектом на локальном сервере. В нашем случае, это `http://localhost/vk-auth`. Базовый домен: `localhost`.

После регистрации приложения откроется следующая форма с настройками приложения (рисунок 2).

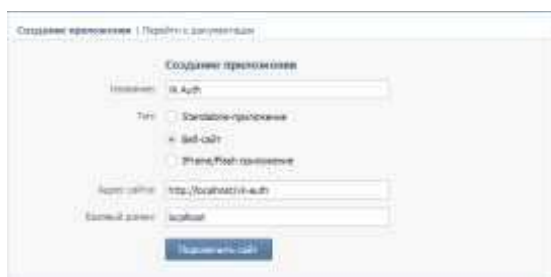


Рисунок 1 – Форма регистрации приложения

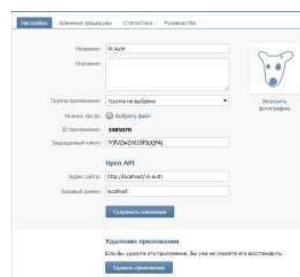


Рисунок 2 – Форма настройки приложения

Из данной формы понадобятся такие данные, как `ID приложения`, `Защищённый ключ`, `Адрес сайта`. Они записываются в специальные переменные в файле index.php (рисунок 3):

```
<?php
$client_id = '3485070'; // ID приложения
$client_secret = '1YjfUZwZmlJJFIqQFAj'; // Защищённый ключ
$redirect_uri = 'http://localhost/vk-auth'; // Адрес сайта
```

Рисунок 3 – Пример кода

Этап 2. Генерация ссылки для аутентификации

Для генерации ссылки потребуется адрес аутентификации и специальные параметры (рисунок 4):

```
$url = 'http://oauth.vk.com/authorize';
$params = array(
    'client_id' => $client_id,
    'redirect_uri' => $redirect_uri,
    'response_type' => 'code' );
```

Рисунок 4 – Пример кода

С помощью функции `http_build_query`, передав туда массив параметров, получается чередование ключей и значений, как в url адресе (рисунок 5):

```
echo $link = '<p><a href="' . $url . '?' . urlencode(http_build_query($params)) . '">Аутентификация через ВКонтакте</a></p>';
```

Рисунок 5 – Пример кода

Этап 3. Получение токена

Для того, чтобы получить специальный токен доступа, с помощью которого, в дальнейшем, будет получаться информация о пользователе, необходим специальный параметр `code`.

Сперва формируются нужные параметры для этого запроса. Далее надо отправить GET запрос, передав перечисленные параметры (рисунок 6):

```

if (isset($_GET['code'])) {
    $params = array(
        'client_id' => $client_id,
        'client_secret' => $client_secret,
        'code' => $_GET['code'],
        'redirect_uri' => $redirect_uri );
    $token= json_decode(file_get_contents('https://oauth.vk.com/access_token'.
.urldecode(http_build_query($params))), true); }

```

Рисунок 6 – Пример кода

В результате, при успешном выполнении запроса в переменную \$token будет записан ответ от ВКонтакте в JSON формате. Данная строка содержит 3 параметра: access_token, который мы будем использовать в следующих запросах для извлечения информации о пользователе, expires_in - время жизни токена, user_id - id пользователя, который прошёл аутентификацию (рисунок 7):

```

{"access_token":"2c6276b767b5e2f35f908e89d61416beea17b6d1ebcd3d14e20ac910281d306bb506ec78e75518ed614e9","expires_in":86399,"user_id":14966712}

```

Рисунок 7 – Пример кода

Для того чтобы далее можно было работать с данными параметрами, JSON строка декодируется с помощью функции json_decode и помещаются данные в массив, передав в качестве второго аргумента true.

Этап 4. Получение информации о пользователе

Для начала подготавливается массив с параметрами, которые в последствии станут фрагментами url строки.

В параметр uids записывается id пользователя; в fields перечисляется через запятую поля, которые надо извлечь. В качестве последнего параметра передаётся 'access_token'.

Для получения информации о пользователе сформированные параметры надо отправить GET запросом (рисунок 8):

```

if (isset($token['access_token'])) {
    $params = array(
        'uids' => $token['user_id'],
        'fields'=> 'uid, first_name, last_name, sex, bdate',
        'access_token' => $token['access_token'] );
    $userInfo= json_decode(file_get_contents('https://api.vk.com/method/users.get' . '?' . urldecode(http_build_query($params))), true); } }

```

Рисунок 8 – Пример кода

В результате получается JSON ответ (рисунок 9):

```
{'response':[{"uid":14966712,"first_name":"Саша","last_name":"Гусакова","sex":1,"bdate":'6.11.1997"}]}
```

Рисунок 9 – Пример кода

JSON ответ преобразуется в массив и идёт обращение к нулевому элементу, хранящемуся в массиве, доступному по ключу response (рисунок 10):

```
if (isset($UserInfo['response'][0]['uid'])) {  
    $UserInfo = $UserInfo['response'][0];  
    $result = true; } }
```

Рисунок 10 – Пример кода

Этап 5. Извлечение информации о пользователе

Теперь извлекать информацию о пользователе мы можем из массива, хранящегося в переменной \$UserInfo по ключам uid, first_name, last_name, screen_name, sex, bdate, photo_big (рисунок 11):

```
if ($result) {  
    echo "Социальный ID пользователя: " . $UserInfo['uid'] . '<br />';  
    echo "Имяпользователя: " . $UserInfo['first_name'] . '<br />';  
    echo "Пол пользователя: " . $UserInfo['sex'] . '<br />';  
    echo "ДеньРождения: " . $UserInfo['bdate'] . '<br />';  
    echo "<br />"; } ?>
```

Рисунок 11 – Пример кода

Этап 6. Регистрация на сайте или авторизация

После получения ID пользователя, необходимо проверить его наличие в базе данных. Если пользователя с таким ID не существует, значит он авторизуется впервые, и ID вносится в базу. Если пользователь уже есть, можно проверить актуальность его данных. После этого создаётся сессия, в которую помещается информация о пользователе (рисунок 12):

```
$_SESSION['user'] = $UserInfo;
```

Рисунок 12 – Пример кода

Регистрация и авторизация через другие социальные сети происходит аналогичным образом.

ЛИТЕРАТУРА

1. Ruseller – частная коллекция качественных материалов для тех, кто делает сайты [Электронный ресурс] / Евгений Попов –2018. – Режим доступа: <https://ruseller.com/lessons.php?rub=37&id=1659> – Дата доступа: 30.03.2018.