

Uniwersytet Przyrodniczo-Humanistyczny w Siedlcach
Wydział Nauk Ścisłych
Kierunek Informatyka

Problemy bezpieczeństwa komputerowego w systemach informatycznych

Dokumentacja zadania indywidualnego

Autor:
Sebastian Kowalczyk
Informatyka

Prowadzący:
dr Piotr Świtalski

Siedlce 2021

1. Cel zadania indywidualnego

Celem projektu jest implementacja standardu autoryzacyjnego OAuth 2.0. Do wykonania zadania wykorzystany będzie serwis, który umożliwi autoryzację, za pomocą tego standardu. W ramach autoryzacji, zwrócone zostaną dane z serwisu w przypadku poprawnego logowania.

2. Instalacja

Projekt został stworzony przy użyciu programu MAMP PRO, który umożliwia uruchomienie programu napisanego w php. Projekt został napisany w wersji PHP 7.4.1.

3. Protokół OAuth 2.0

Jest to protokół, który umożliwia bezpieczną autoryzację z użyciem różnych platform WWW, przykładowo serwis GitHub. Standard ten jest opisany w dokumencie RFC 6749, znajduje się w nim między innymi informacja o tym, że jest to framework autoryzacyjny. Nie pojawia się tutaj pojęcie protokołu, co oznacza że nie trzeba trzymać się ściśle określonych zasad, przez co dopuszcza różne sposoby implementacji elementów tego standardu.

Głównym zadaniem OAuth2.0 jest autoryzacja zasobów. Właściciel może udostępnić swój zasób innemu podmiotowi. W ten sposób można udzielić prawa innym aplikacjom do odczytu.

Przykładowa autoryzacja składa się z następujących etapów:

- W przypadku chęci uzyskania danych użytkownika, aplikacja jest przekierowywana do serwera autoryzacyjnego
- Serwis autoryzujący, przedstawia formularz z informacją, że aplikacja chce uzyskać dostęp do wybranych elementów.
- W kolejnym kroku użytkownik musi zaakceptować wymagania aplikacji i następuje autoryzacja lub jej brak, w zależności od wyboru użytkownika
- Po udzieleniu autoryzacji, serwer przekierowuje z powrotem do aplikacji, która otrzymuje token umożliwiający pobranie wybranych danych, profilu użytkownika.

W procesie tym najważniejszy jest zwrócony token, który upoważnia do pobrania danych. Na jego podstawie na serwerze sprawdzane jest, czy podmiot przedstawiający token ma autoryzację do zasobów, które chce pobrać lub operacji, które chce wykonać.

4. Ustawienie autoryzacja w serwisie.

W pierwszej kolejności należy ustawić podstawowe dane do autoryzacji. Wymaganymi danymi są: nazwa aplikacji, adres strony domowej oraz adres, na który ma być skierowany użytkownik po procesie autoryzacji.

W przypadku serwisu GitHub należy wejść w ustawienia a następnie wybrać „Developer settings” a w kolejnym kroku „OAuth Apps”. Następnie kliknąć przycisk do utworzenia nowej aplikacji w standardzie OAuth.

Register a new OAuth application

Application name *

Something users will recognize and trust.

Homepage URL *

The full URL to your application homepage.

Application description

This is displayed to all users of your application.

Authorization callback URL *


Your application's callback URL. Read our [OAuth documentation](#) for more information.

Register application

Cancel

Po prawidłowym wypełnieniu formularza, widoczny jest id klienta, który będzie wykorzystany jako parametr w URL o nazwie client_id. Należy go skopiować i dodać do aplikacji, wykorzystującej autoryzację. W kolejnym kroku należy wygenerować nowy sekretny klucz klienta, który również jest wymagany w następnym parametrze do autoryzacji.

PBKwSI

 **xKoSeMi** owns this application.

Transfer ownership

You can list your application in the [GitHub Marketplace](#) so that other users can discover it.

List this application in the Marketplace

1 user


Revoke all user tokens

Client ID

e7c717a2e4b9c8d9d1d1

Client secrets

Generate a new client secret




Client secret

*****2992a4ee

Added 6 hours ago by xKoSeMi

Last used within the last week

Delete



Client secret

*****4ded0253

Added 11 hours ago by xKoSeMi

Last used within the last week

Delete

5. Implementacja

Z dokumentacji OAuth na GitHubie należy skopiować link do autoryzacji

1. Request a user's GitHub identity

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

Następnie z ustawień profilu należy skopiować Id klienta i dodać jako client_id.

```
<div class="container">
  <div class="center">
    <button class="button button1" onclick="location.href='https://github.com/login/oauth/authorize?client_id=e7c717a2e4b9c8d9d1d1'" type="button">
      Zaloguj się
    </button>
  </div>
</div>
```

```
https://github.com/login/oauth/authorize?client_id=e7c717a2e4b9c8d9d1d1'"
```

Po przekierowaniu na stronę z formularzem i poprawnym logowaniu, zwracany jest w URL jednorazowy kod, który trzeba pobrać np. za pomocą \$_GET.

```
$code = $_GET['code'];
```

W kolejnym kroku należy ustawić 3 wymagane parametry: id klienta, sekretny klucz oraz zwrócony kod. Dodatkowo, parametry muszą zostać wysłane na podany adres, który znajduje się w dokumentacji.

```
POST https://github.com/login/oauth/access_token
```

```
$clientId = "e7c717a2e4b9c8d9d1d1";
$clientSecretKey = "a576f94bf24b27bfa949b7686cb57ab82992a4ee";
$url = "https://github.com/login/oauth/access_token";

$postParams = [
    'client_id' => $clientId,
    'client_secret' => $clientSecretKey,
    'code' => $code
];
```

W kolejnym kroku przy użyciu cURL wysyłane jest zapytanie na podany url a także wymagane parametry. Należy również uwzględnić format, który również jest opisany w dokumentacji.

```
Accept: application/json
```

```

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL,$url);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $postParams);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
curl_setopt($ch, CURLOPT_HTTPHEADER,array('Accept: application/json'));

$response = curl_exec($ch);

```

W przypadku uzyskania poprawnej autoryzacji, zwracany jest response z serwera, który zawiera accessToken umożliwiające pobranie wybranych danych z serwera. W przypadku pomyślnego zakończenia operacji, token umieszczany jest w sesji a użytkownik zostanie przekierowany na stronę, gdzie wykonywane są zapytania o wybrane informacje z repozytorium.

Do dostępu do API należy skierować użytkownika pod wybrany adres.

```

Authorization: token OAUTH-TOKEN
GET https://api.github.com/user

```

Jako parametry w tym przypadku należy podać uzyskany token oraz User-Agent z nazwą aplikacji.

```

$url = "https://api.github.com/user";

$authHeader = "Authorization: token " . $accessToken;
$userAgentHeader = "User-Agent: PBKwSI";
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Accept: application/json', $authHeader, $userAgentHeader));

```

Po poprawnej autoryzacji, z serwera wysyłane są dane użytkownika takie jak login, id czy awatar.

```

echo '<div class="login">';
echo '';
echo '<div class="nick"><strong>Witaj ' . $data['login'] . '</br> ID: ' . $data['id'] . '</strong></div>';
echo '</div>';

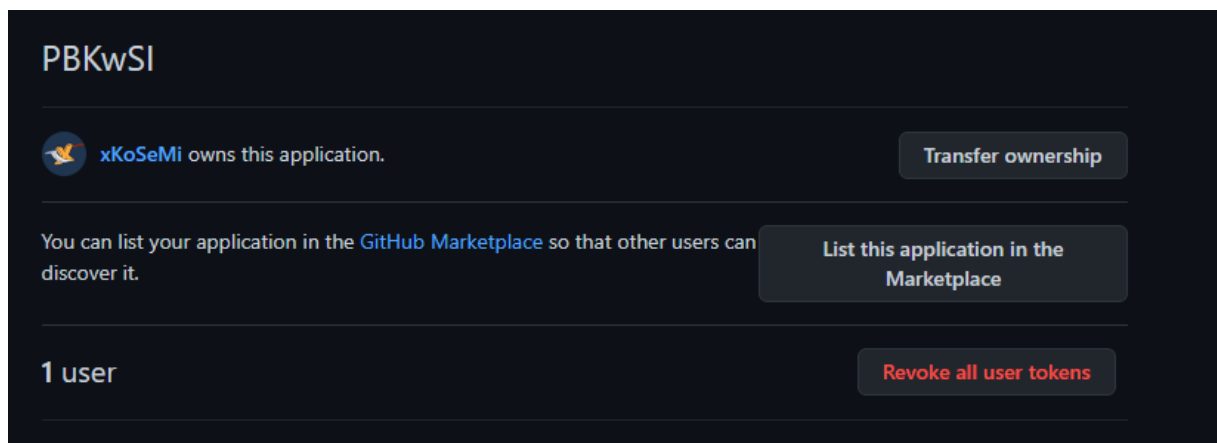
```

Do uzyskania dostępu do repozytorium należy wykonać ponowne zapytanie pod wybrany adres z takimi samymi parametrami:

```
$repoUrl = $data['repos_url'];

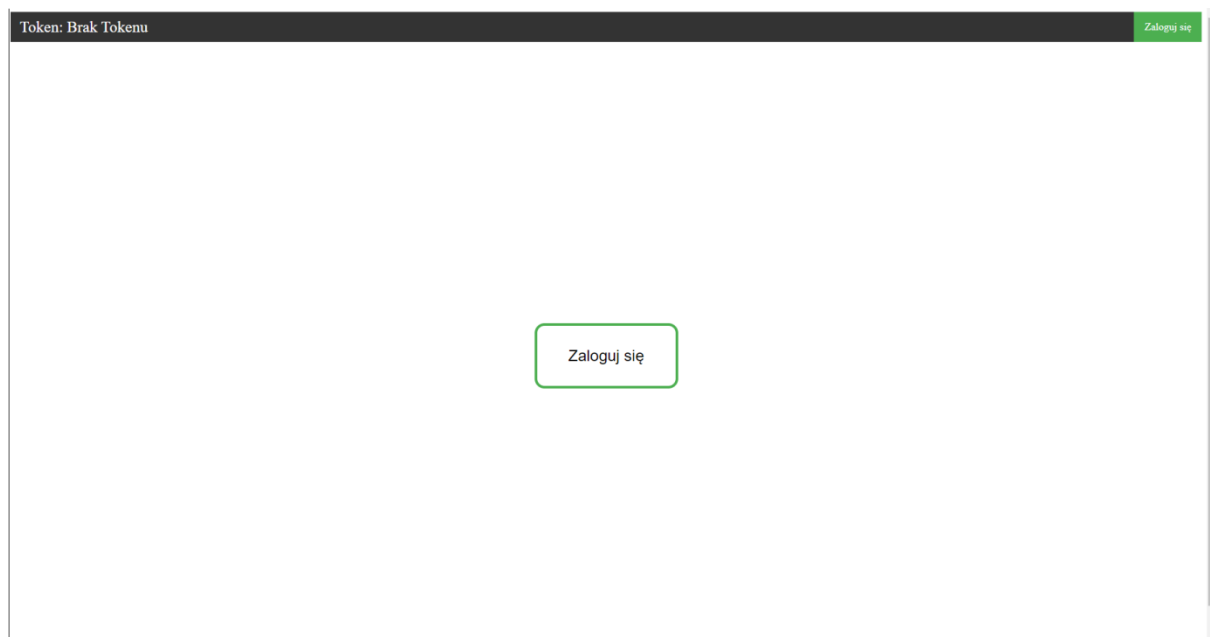
$ch2 = curl_init();
curl_setopt($ch2, CURLOPT_URL, $repoUrl);
curl_setopt($ch2, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch2, CURLOPT_SSL_VERIFYHOST, 0);
curl_setopt($ch2, CURLOPT_SSL_VERIFYPEER, 0);
curl_setopt($ch2, CURLOPT_HTTPHEADER, array('Accept: application/json', $authHeader, $userAgentHeader));
$response = curl_exec($ch2);
curl_close($ch2);
```

Właściciel dodatkowo widzi ilość autoryzowanych użytkowników, a także może ich usunąć.

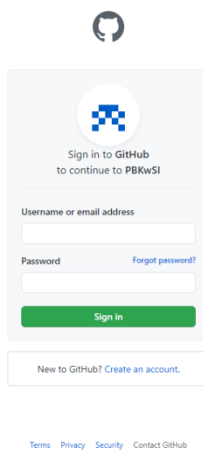


6. Działanie aplikacji

Po uruchomieniu aplikacji użytkownik widzi w lewym górnym rogu informację o tokenie oraz 2 przyciski z możliwością zalogowania się. Oba przyciski przekierowują na stronę z logowaniem do konta GitHub.



Po kliknięciu przycisku, jest on przekierowany na stronę z logowaniem.



Po poprawnym zalogowaniu się widać wartość tokenu, dane użytkownika oraz repozytorium.

Token: b29086815bdf70d20941604a90122791f9e4ea05

Wyloguj



Witaj xKoSeMi
ID: 55414290

Nazwa	Opis	Język	Data utworzenia	Ostatnia zmiana	Obserwujący
ConsoleApp6			2019-09-17T10:33:52Z	2019-09-17T10:56:33Z	0
ConsoleApp8	?	C#	2019-09-17T11:03:12Z	2019-09-17T16:54:18Z	0
hello-world	my decription		2019-09-16T21:52:46Z	2019-09-16T22:45:27Z	0
MD.github.io		HTML	2021-01-10T15:46:53Z	2021-01-10T15:48:49Z	0
Monika-Druzba		HTML	2021-01-10T16:23:19Z	2021-01-10T16:24:36Z	0
Monika-Druzba-Strona.github.io		HTML	2021-01-10T16:43:22Z	2021-01-10T16:45:57Z	0
Monika-Druzba.github.io		HTML	2021-01-10T16:18:24Z	2021-01-10T17:14:31Z	0
Zespo-owy		JavaScript	2020-01-29T19:50:18Z	2020-01-29T20:11:18Z	0

Po kliknięciu przycisku „Wyloguj” token zostanie usunięty z sesji a użytkownik przeniesiony na widok główny.