

Jak zintegrować oprogramowanie zewnętrzne z programem eSZOK Pulpit Klienta Plus?



Dokument jest przeznaczony wyłącznie dla osób, które mają przynajmniej podstawową wiedzę z zakresu obsługi oprogramowania eSZOK, którego producentem jest Centrum Technologii Informatycznej.

Dokument przygotowała: Barbara Niżnikiewicz - Product Manager przy wsparciu zespołu programistycznego.

Data utworzenia : 2023.03.23

Wprowadzenie.

Oprogramowanie eSZOK jest zbiorem kilku programów, które wspierają polskie biura rachunkowe we wprowadzaniu danych księgowych do systemu księgowego ERP OPTIMA firmy Comarch SA.

Całe oprogramowanie zostało wymyślone w 2017 roku przez Zygmunta Wilder i napisane przez specjalistów z Centrum Technologii Informatycznej. Jest ono ciągle rozwijane pod potrzeby klientów i zmiany przepisów podatkowych.

Jednym z ważnych elementów oprogramowania jest funkcjonalność API pozwalająca na zintegrowanie zewnętrznego oprogramowania z Pulpitem Klienta, dzięki czemu mogą być przesyłane dane z programu zewnętrznego tzn. innego producenta do programu Pulpit Klienta.

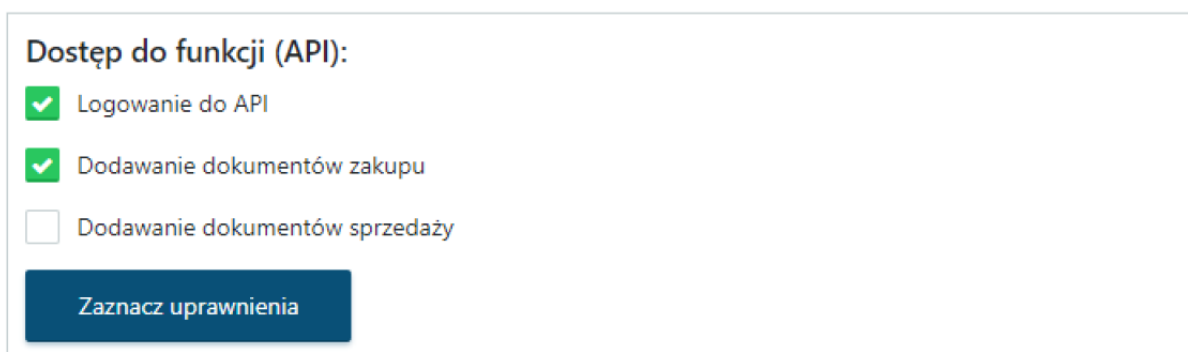
W Pulpicie Klienta od strony zaplecza w zakładce **Klienci** -> **Użytkownicy** możemy nadać użytkownikowi dostęp do funkcji API dzięki czemu uzyska możliwość logowania do API programu, będzie mógł dodawać dokumenty zakupu czy sprzedaży bezpośrednio z oprogramowania zewnętrznego.

Rozdział 1.

Nadanie dostępu do funkcjonalności API w Pulpicie Klienta.

W wersji rozszerzonej Pulpitu Klienta od strony zaplecza w zakładce **Klienci-> Użytkownicy** została dodana zakładka API.

API



Dostęp do funkcji (API):

- Logowanie do API
- Dodawanie dokumentów zakupu
- Dodawanie dokumentów sprzedaży

Zaznacz uprawnienia

Jeśli chcemy nadać dostęp zewnętrznej aplikacji do Pulpitu Klienta poprzez API, to nadajemy odpowiednie uprawnienia użytkownikowi Pulpitu Klienta, za pośrednictwem którego, będą się wymieniać informacje pomiędzy aplikacjami.

Dostępne metody:

- Logowanie do API
- Dodawanie dokumentów zakupu
- Dodawanie dokumentów sprzedaży

Należy pamiętać, że każda metoda Dodanie dokumentów zakupu/ Dodawanie dokumentów sprzedaży wymaga autoryzacji użytkownika czyli Logowania do API.

Rozdział 2.

Logowanie do API

Zewnętrzne oprogramowanie, którym chcemy się połączyć do aplikacji webowej Pulpit Klienta Plus musi zawierać odpowiednie parametry zapytań :

Request: http://NAZWA_DOMENY/public/api/user/login (Typ: POST)

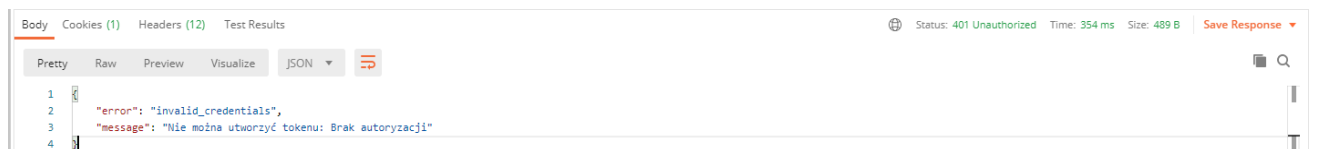
Wymagane parametry body:

- identyfikatorFirmy
- nazwaUzytkownika
- hasloUzytkownika

Przy udanej autoryzacji program zwróci prawidłową odpowiedź API.



Jeśli walidacja nie przebiegnie pomyślnie – czyli walidacja błędna, program zwróci następującą odpowiedź:



Jeśli chcielibyśmy zintegrować się z API Pulpitu Klienta PLUS za pomocą języka programowania C# należy zastosować następujące fragmenty kodu :

```
var client = new RestClient("http://NAZWA_DOMENY/public/api/user/login");
client.Timeout = -1;
var request = new RestRequest(Method.POST);
request.AddHeader("Cookie", "Cookie_1=value");
request.AlwaysMultipartFormData = true;
request.AddParameter("nazwaUzytkownika", "Jan Nowak");
request.AddParameter("hasloUzytkownika", "12345678");
request.AddParameter("identyfikatorFirmy", "FIRMA_DEMO");
IRestResponse response = client.Execute(request);
Console.WriteLine(response.Content);
```

W przypadku integracji z API Pulpitu Klienta Plus przy udziale języka programowania PHP cURL należy zastosować następujące fragmenty kodu:

```
<?php
```

```
$curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_URL => 'http://NAZWA_DOMENY/public/api/user/login',
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => '',
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 0,
    CURLOPT_FOLLOWLOCATION => true,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => 'POST',
    CURLOPT_POSTFIELDS => array('nazwaUzytkownika' => 'Jan Nowak', 'hasloUzytkownika' => '12345678', 'idenftyfikatorFirmy' => 'FIRMA_DEMO'),
    CURLOPT_HTTPHEADER => array(
        'Cookie: Cookie_1=value'
    ),
));

$response = curl_exec($curl);

curl_close($curl);
echo $response;
```

Rozdział 3.

Dodawanie dokumentów zakupu:

Istnieje możliwość dodawanie dokumentów zakupu z zewnętrznego oprogramowania do Pulpitu Klienta poprzez API. Oprogramowanie musi zawierać odpowiednie fragmenty zapytań :

Request: http://NAZWA_DOMENY/public/api/user/invoices/buy/addFile (Typ: POST) (Tylko PDF: application/pdf)

Wymagane parametry headers:

- Authorization (Bearer {token z logowania})
- Content-Type (application/json)

Wymagane parametry body:

- daneDokumentowZakupu (typ: Array) [Pole nazwa_pliku – bez rozszerzenia]

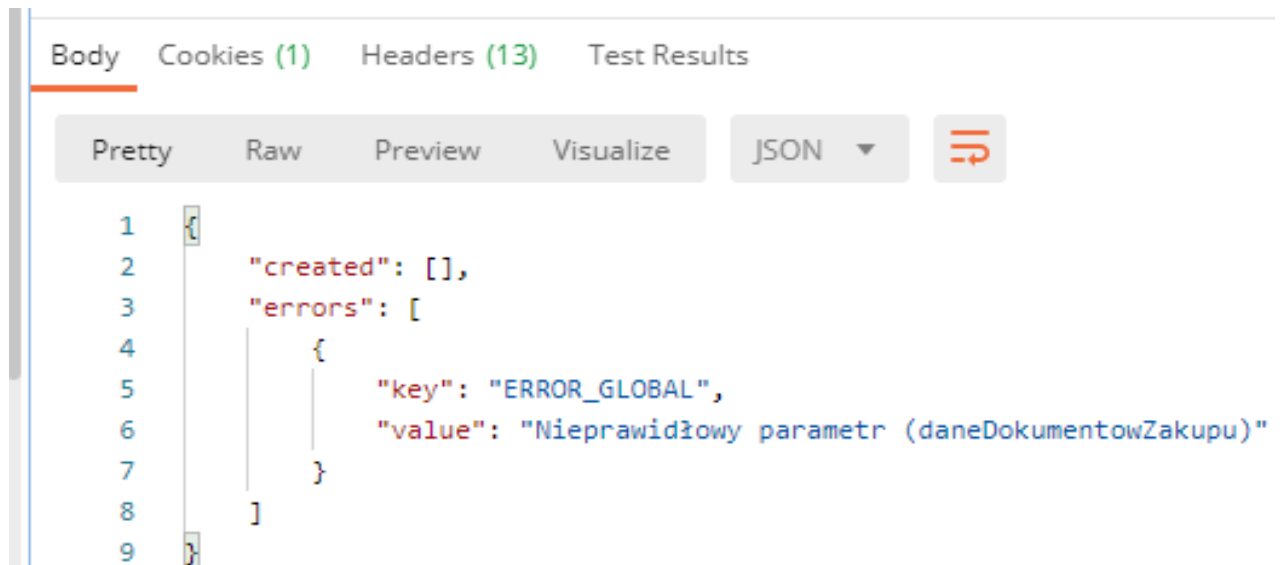
```
[daneDokumentowZakupu] => Array
(
  [0] => Array
  (
    [nazwa_pliku] => "nazwa pliku bez rozszerzenia"
    [base64_pliku] => "base64 plik"
  )
  [1] => Array
  (
    [nazwa_pliku] => "nazwa pliku bez rozszerzenia"
    [base64_pliku] => "base64 plik"
  )
)
```

Przykład prawidłowej odpowiedzi API (udanej autoryzacji z nieudanymi kluczami):



```
Body Cookies (1) Headers (13) Test Results
Pretty Raw Preview Visualize JSON ↕
1 {
2   "created": [
3     {
4       "uuid": "40e85a69-5fe0-44c9-8abb-7e99f6a0f91f",
5       "key": 0
6     },
7     {
8       "uuid": "08f9e4a7-0b0f-4b14-8bff-45889bc8fbe4",
9       "key": 3
10    }
11  ],
12  "errors": [
13    {
14      "key": "ERROR_DOCUMENT",
15      "value": "Nieprawidłowy parametr (nazwa_pliku) Klucz: 1"
16    },
17    {
18      "key": "ERROR_DOCUMENT",
19      "value": "Nieprawidłowy parametr (nazwa_pliku) Klucz: 2"
20    }
21  ]
22 }
```

Przykład błędnej walidacji:



The screenshot shows a REST client interface with tabs for 'Body', 'Cookies (1)', 'Headers (13)', and 'Test Results'. The 'Body' tab is active, and the response is displayed in 'Pretty' format. The JSON response is as follows:

```
1  {
2    "created": [],
3    "errors": [
4      {
5        "key": "ERROR_GLOBAL",
6        "value": "Nieprawidłowy parametr (daneDokumentowZakupu)"
7      }
8    ]
9  }
```

Jeśli chcielibyśmy przesyłać dokumenty zakupu do Pulpitu Klienta z aplikacją zewnętrzną opartej na języku programowania C# należy zastosować fragment kodu:

```
var client = new RestClient("http://NAZWA_DOMENY/public/api/user/invoices/buy/addFile");
client.Timeout = -1;
var request = new RestRequest(Method.POST);
request.AddHeader("Authorization", "Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOi8vbG9jYXRob3N0L2Vzem9rX2JyL3B1YmtpYy9hcGkvdXNlci9sb2dpbiIsIm1hdCI6MTY3NzY3Nzg2MywiZXhwIjoxNjc3Njg3NDYzLCJyYmYiOiJlE2Nzc2Nzc4NjMsImp0aSI6IkJaeVBdk9VaWpXZkZlEiLCJzdWwiOiJMsInBydiI6Ijg3ZTBhZjFlZjlmZDE1ODEyZmRlYzk3MTUzYTE0ZTBiMDQ3NTQ2YWEifQ.Ya745IcAvr-QLR_qwDEK1EqKKcd-kiQKhEMwHtoWHpo"); // Bearer {token z logowania}
request.AddHeader("Content-Type", "application/json");
request.AddHeader("Cookie", "Cookie_1=value");
request.AlwaysMultipartFormData = true;

request.AddParameter("daneDokumentowZakupu[0][nazwa_pliku]", "test2");
request.AddParameter("daneDokumentowZakupu[0][base64_pliku]", "base64 pliku");
//Opcjonalnie Lista: request.AddJsonBody(new { daneDokumentowZakupu = documents });

IRestResponse response = client.Execute(request);
Console.WriteLine(response.Content);
```

W przypadku przesyłania dokumentów zakupu do Pulpitego Klienta Plus z aplikacji zewnętrznej opartej na języku programowania PHP cURL należy zastosować następujący fragment kodu:

```
<?php
```

```
$curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_URL => 'http:// NAZWA_DOMENY/public/api/user/invoices/buy/addFile',
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => '',
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 0,
    CURLOPT_FOLLOWLOCATION => true,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => 'POST',
    CURLOPT_POSTFIELDS => array('daneDokumentowZakupu[0][nazwa_pliku]' => 'test2', 'daneDokume
ntowZakupu[0][base64_pliku]' => 'base64 pliku'),
    CURLOPT_HTTPHEADER => array(
        'Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOi8vbG9jYXVz
ob3N0L2Vzem9rX2JyL3B1YmtpYy9hcGkvdXNlc29sb2dpbiIsIm1hdCI6MTY3NzY3Nzg2MywiZXhwIjoxNjczNjc3ND
YzL0JmZTE0ZDZmRlZk3MTUzYTE0ZTBiMDQ3NTQ2YWEifQ.Ya745IcAvr-QLR_qwDEK1EqKkcd-kiQKhEMwHtoWHpo',
        'Content-Type: application/json',
        'Cookie: Cookie_1=value'
    ),
));

$response = curl_exec($curl);

curl_close($curl);
echo $response;
```

Rozdział 4.

Dodawanie dokumentów sprzedaży:

Istnieje możliwość dodawania dokumentów sprzedaży z zewnętrznego oprogramowania do Pulpitego Klienta poprzez API. Oprogramowanie musi zawierać odpowiednie fragmenty zapytań:

Request: http://NAZWA_DOMENY/public/api/user/invoices/sell/addFile (Typ: POST) (Tylko PDF: application/pdf)

Wymagane parametry headers:

- Authorization (Bearer {token z logowania})
- Content-Type (application/json)

Wymagane parametry body:

- daneDokumentowSprzedazy (typ: Array) [Pole nazwa_pliku – bez rozszerzenia]

```
[daneDokumentowSprzedazy] => Array
(
  [0] => Array
  (
    [nazwa_pliku] => "nazwa pliku bez rozszerzenia"
    [base64_pliku] => "base64 plik"
  )
  [1] => Array
  (
    [nazwa_pliku] => "nazwa pliku bez rozszerzenia"
    [base64_pliku] => "base64 plik"
  )
)
```

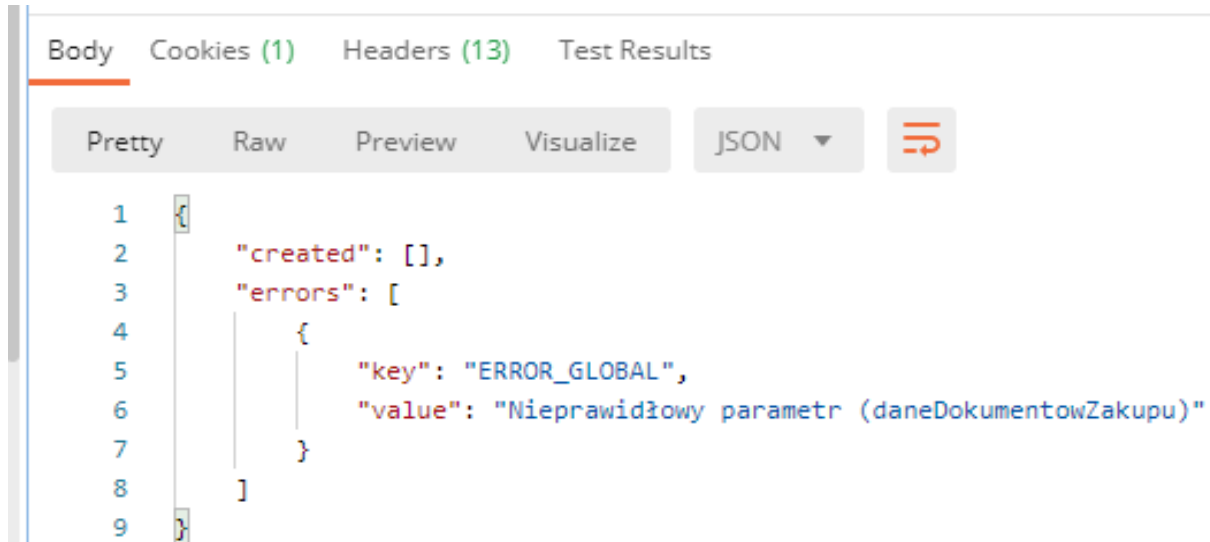
Przykład poniżej przedstawia prawidłową odpowiedź API- udana autoryzacja z nieudanymi kluczami:

The screenshot shows a REST client interface with tabs for 'Body', 'Cookies (1)', 'Headers (13)', and 'Test Results'. The 'Body' tab is active, and the response is displayed in 'Pretty' format as JSON. The JSON structure is as follows:

```

1  {
2    "created": [
3      {
4        "uuid": "40e85a69-5fe0-44c9-8abb-7e99f6a0f91f",
5        "key": 0
6      },
7      {
8        "uuid": "08f9e4a7-0b0f-4b14-8bff-45889bc8f8be4",
9        "key": 3
10     }
11   ],
12   "errors": [
13     {
14       "key": "ERROR_DOCUMENT",
15       "value": "Nieprawidłowy parametr (nazwa_pliku) Klucz: 1"
16     },
17     {
18       "key": "ERROR_DOCUMENT",
19       "value": "Nieprawidłowy parametr (nazwa_pliku) Klucz: 2"
20     }
21   ]
22 }
```


W przypadku błędnej walidacji, program zwróci informację:



```
Body Cookies (1) Headers (13) Test Results
Pretty Raw Preview Visualize JSON
1 {
2   "created": [],
3   "errors": [
4     {
5       "key": "ERROR_GLOBAL",
6       "value": "Nieprawidłowy parametr (daneDokumentowZakupu)"
7     }
8   ]
9 }
```

W przykład dodawania dokumentów sprzedaży przy zastosowaniu aplikacji/ mechanizmu w języku C# należy zastosować następujący fragment kodu:

```
var client = new RestClient("http://NAZWA_DOMENY/public/api/user/invoices/sell/addFile");
client.Timeout = -1;
var request = new RestRequest(Method.POST);
request.AddHeader("Authorization", "Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOi8vbG93YWxob3N0L2Vzem9rX2JyL3B1Ym9yYy9hcGkvdXNlci9sb2dpbiIsIm1hdCI6MTY3NzY3Nzg2MywiZXhwIjoxNjc3Njg3NDYzLCJyYm9yYyE2Nzc2Nzc4NjMsImp0aSI6IkJaevBBDk9VaWpXZk9Vb1EiLCJzdWIiOiJMsInBydiI6Ijg3ZTBhZjFlZjlmZDE1ODEyZmRlYzk3MTUzYTE0ZTBiMDQ3NTQyWEifQ.Ya745IcAvr-QLR_qwDEK1EqKKcd-kiQkHEmWhtoWHpo"); // Bearer {token z logowania}
request.AddHeader("Content-Type", "application/json");
request.AddHeader("Cookie", "Cookie_1=value");
request.AlwaysMultipartFormData = true;

request.AddParameter("daneDokumentowSprzedazy[0][nazwa_pliku]", "test2");
request.AddParameter("daneDokumentowSprzedazy[0][base64_pliku]", "base64 pliku");
//Opcjonalnie Lista: request.AddJsonBody(new { daneDokumentowZakupu = documents });

IRestResponse response = client.Execute(request);
Console.WriteLine(response.Content);
```

W przykład dodawania dokumentów sprzedaży przy zastosowaniu aplikacji/ mechanizmu w języku PHP cURL należy zastosować następujący fragment kodu :

```
<?php

$curl = curl_init();

curl_setopt_array($curl, array(
  CURLOPT_URL => 'http:// NAZWA_DOMENY/public/api/user/invoices/sell/addFile',
```

