

Dokumentacja API

wersja 1.1

Dzięki API eWindykator możesz w łatwy sposób zintegrować nasz system ze swoją aplikacją. Wszystkie metody są wywoływane przy pomocy protokołu SOAP, więc istnieje możliwość zarządzania swoimi importami z poziomu większości języków programowania. Wszystkie przykłady są pisane w PHP oraz przy użyciu standardowej klasy SoapClient tego języka. Metody oraz typy danych, wykorzystywane w naszym API są opisane przy pomocy WSDL.

Adres do pliku WSDL: <http://www.ewindykator.pl/application/webapi.xml>

TYPY DANYCH:

User

Obiekt User składa się z dwóch argumentów:

- username -> Twój login do systemu EWindykator
- password -> Twoje hasło do systemu EWindykator

User jest wykorzystywany przy wywoływaniu każdej metody i służy do identyfikacji użytkownika.

Entry

Obiekt entry opisuje pojedynczy wpis importu danych. Jego składowe są następujące:

- customer_name -> Imię i Nazwisko klienta
- document -> Identyfikator dokumentu
- price -> Należna kwota podawana jako liczba z dwoma miejscami po przecinku.

Separatorem miejsc dziesiętnych musi być kropka.

- up_date -> Data wystawienia dokumentu
- dead_date -> Data płatności
- date_format -> Format importowanej daty (Y - cyfry roku, m - cyfry miesiąca, d - cyfry dni)
- customer_group -> Grupa klienta
- document_description -> Opis dokumentu

- email -> Adres e-mail klienta
- telephone -> Numer telefonu komórkowego klienta
- fax_number -> Numer fax klienta
- address -> Adres klienta (w przypadku podania adresu należy także podać dwa poniższe pola)
- post_code -> Kod pocztowy do adresu
- city -> Miasto do adresu

Error

Obiekt zwracany przez API w przypadku wystąpienia błędów podczas importu danych. Do obiektu należą następujące argumenty:

- code -> Kod błędu
- message -> Treść błędu

Lista kodów błędu importu:

- 1001 - Błędne dane użytkownika
- 1002 - Pole nazwa płatnika jest wymagane
- 1003 - Pole numer dokumentu jest wymagane
- 1004 - Pole kwota do zapłaty jest wymagane
- 1005 - Pole format daty jest wymagane
- 1006 - Kwota do zapłaty musi być wartością liczbową
- 1007 - Format daty wystawienia dokumentu jest niezgodny z podanym formatem daty
- 1008 - Pole data płatności jest wymagane
- 1009 - Format daty płatności jest niezgodny z podanym formatem daty
- 1010 - Jedno z pól: e-mail, telefon, numer fax oraz adres jest wymagane
- 1011 - Błędny format adresu e-mail
- 1012 - Jeśli wybrano adres, należy podać także kod pocztowy oraz miasto
- 1013 - Błędny kod pocztowy

simpleResponse

Obiekt, który jest zwracany przy wywołaniu każdej metody. Tutaj mamy zawarte informacje o ostatnio napotkanym błędzie lub sukcesie oraz tablicę błędów, które pojawiły się podczas importu danych. Argumenty tego obiektu wyglądają następująco:

- code -> kod zwracany przez ostatnią operację
- message -> wiadomość podsumowania
- errors -> Tablica obiektów Error, jeśli wystąpiły jakieś błędy

TABLICE DANYCH

W naszym API mają Państwo także dostęp do korzystania z tablicy danych. Do wykorzystania są następujące typy danych:

ArrayOfEntry

Tablica obiektów Entry, wykorzystywana przy metodzie storeEntries()

Errors

Tablica obiektów Error, zwracana wraz z obiektem simpleResponse.

METODY

storeEntry()

Metoda, której zadaniem jest dodanie pojedynczego klienta. Do wywołania metody należy podać następujące argumenty:

- user -> Obiekt typu User do weryfikacji konta EWindykator
- entry -> Obiekt typu Entry czyli dane pojedynczego klienta do zaimportowania

Metoda zwraca obiekt simpleResponse w argumencie o nazwie „response”.

storeEntries()

Metoda, która jest odpowiednikiem metody storeEntry, z tą różnicą, że zamiast pojedynczego obiektu Entry podajemy tablicę klientów:

- user -> Obiekt typu User do weryfikacji konta EWindykator
- entries -> Tablica obiektów Entry czyli tablica klientów do zaimportowania

Metoda zwraca obiekt simpleResponse w argumencie o nazwie „response”.

clearEntries()

Metoda, która służy do czyszczenia zaimportowanych danych w panelu EWindykator. Do wywołania tej metody podajemy następujące argumenty:

- user -> Obiekt typu User do weryfikacji konta EWindykator

Metoda zwraca obiekt simpleResponse w argumencie o nazwie „response”.

PRZYKŁADY

```
storeEntry()

ini_set("soap.wsdl_cache_enabled", "0");
$client = new SoapClient("http://ewindykator.pl/application/webapi.xml");

// Obiekty typu User
$user = new stdClass();
$user->username = 'test';
$user->password = 'test';

// Obiekt typu Entry
$entry = new stdClass();
$entry->customer_name = 'test Entry';
$entry->document = 'test Entry';
$entry->price = '145,30';
$entry->up_date = '10.20.2012';
$entry->dead_date = '05.01.2012';
$entry->date_format = 'mm.dd.YYYY';
$entry->customer_group = 'test Entry';
$entry->document_description = 'test Entry';
$entry->email = "";
$entry->telephone = 'test Entry12311123';
$entry->fax_number = 'test Entry3333';
$entry->address = 'test Entry';
$entry->post_code = '00-111';
$entry->city = 'test Entry';

// Dodanie klienta do panelu
$response = $client->storeEntry($user, $entry);

// Pokaz zwrocone dane typu simpleResponse
print_r($response);
```

```
storeEntries();

ini_set("soap.wsdl_cache_enabled", "0");
$client = new SoapClient("http://ewindykator.pl/application/webapi.xml");

// Obiekty typu User
$user = new stdClass();
$user->username = 'test';
$user->password = 'test';

// Obiekt typu Entry
$entry = new stdClass();
$entry->customer_name = 'test Entry';
$entry->document = 'test Entry';
$entry->price = '145,30';
$entry->up_date = '10.20.2012';
$entry->dead_date = '05.01.2012';
$entry->date_format = 'mm.dd.YYYY';
$entry->customer_group = 'test Entry';
$entry->document_description = 'test Entry';
$entry->email = "";
$entry->telephone = 'test Entry12311123';
$entry->fax_number = 'test Entry3333';
$entry->address = 'test Entry';
$entry->post_code = '00-111';
$entry->city = 'test Entry';

// Stworz tablice wpisow
$entries = array($entry, $entry, $entry);

// Dodanie klienta do panelu
$response = $client->storeEntries($user, $entries);
// Pokaz zwrocone dane typu simpleResponse
print_r($response);

clearEntries()

ini_set("soap.wsdl_cache_enabled", "0");
$client = new SoapClient("http://ewindykator.pl/application/webapi.xml");

// Obiekty typu User
$user = new stdClass();
```

```
$user->username = 'test';
```

```
$user->password = 'test';
```

```
// Wyczysc dane zaimportowane
```

```
$response = $client->clearEntries($user);
```

```
print_r($response);
```

