Wikiprint Book

Title: eDokumenty Api

Subject: eDokumenty - elektroniczny system obiegu dokumentów, workflow i CRM - DeployerGuide/Others/eDokumentyApi

Version: 97

Date: 12/06/25 07:36:08

Table of Contents

eDokumenty Api 3
- *Styl/format dokumentów SOAP* 3
- *Uwagi dotyczące haseł* 4
- *Obsługa błędów API* 4
- *Uwagi* 4
- *Dokumentacja API 2.2.0* 5

eDokumenty Api

System eDokumenty udostępnia API (Application Programming Interface) dzięki któremu jesteśmy w stanie przeprowadzać podstawowe czynności bez konieczności logowania do systemu.

Usługa jest zabezpieczona rozszerzeniem WSSecurity protokołu SOAP i wymaga podania w nagłówkach wywołania XML nazwy użytkownika i hasła.

Hasło i użytkownik to specjalne dane, które należy wprowadzić do pliku config.inc pod kluczami:

```
<?php
define('EDOK_API_LOGIN', 'edok_api_user');
define('EDOK_API_PASSWORD', 'edok_api_pass');

// Dodatkowa stała która umożliwia pominięcie autentykacji (jeśli ustawiamy na FALSE to nie sprawdza danych EDOK_API_LOGIN
// domyślnie ustawiona na TRUE, można wysłać GET lub dodatkowe nagłówki WSS
// define('EDOK_API_AUTH_MODE', TRUE);
?>
```

Wartości stałych w powyższym przykładzie konfiguracji są tylko danymi prezentacyjnymi i nie powinno się ich używać na produkcyjnej bazie.

Stałe te mogą mieć dowolne wartości ważne jednak aby te same wartości podać przy wywołaniu usługi SOAP w kliencie.

Od wersji 4.0 systemu eDokumenty jest możliwość autentykacji poprzez dowolne konto użytkownika (nie musi to być EDOK_API_LOGIN, jak we wcześniejszych wersjach).

Usługa jest dostępna pod adresem:

```
http://{host}:{port}/eDokumentyApi.php
```

Wartość {host} oraz {port} należy zamienić odpowiednimi wartościami zgodnymi z konfiguracją serwera instalacyjnego systemu eDokumenty.

Dodatkowo od wersji systemu 3.3 autentykacja nie wymaga nadpisania klienta Soap. Wystarczy do url z adresem serwisu dodać parametry GET w postaci

```
<?php
ini_set('display_errors', 'On');

define('EDOK_API_LOGIN', 'user');
define('EDOK_API_PASSWORD', 'password');
define('DEFAULT_ENTITY_SYMBOL', 'demo');

$ops = array(
   'location' => 'http://localhost/eDokumentyApi.php?a1='.EDOK_API_LOGIN.'&a2='.md5(md5(EDOK_API_PASSWORD).'_SOAP_eDok_api
   "uri" => "eDokumentyAPI",
   'encoding'=>'UTF-8'
);

$client = new SoapClient(null, $ops);

$params = array(
   'name_1' => 'Daniel Wąsala'
);

/* Invoke webservice method with your parameters, in this case: Function1 */
$response = $client->searchContacts($params);

/* Print webservice response */
var_dump($response);
```

Styl/format dokumentów SOAP

Api może działać w dwóch stylach/formatach:

1. RPC/encoded (domyślny). Adres usługi to http://{host}/eDokumentyApi.php (http://{host}/eDokumentyApi.php?wsdl)
2. Document/Literal. Adres usługi to http://{host}/eDokumentyApi.php/2 (http://{host}/eDokumentyApi.php/2?wsdl)

Uwagi dotyczące haseł

1. Hasło powinno być zakodowane przy użyciu funkcji md5.

```
$password = md5('haslo');
```

Jeżeli nie korzystamy z klasy **EDokApiClient**, to doklejamy do zakodowanego hasła ciąg "_SOAP_eDok_api" i ponownie tworzymy skrót md5.

```
$password = md5(md5('haslo').'_SOAP_eDok_api');
```

Obsługa błędów API

Zgodnie z przykładem dla PHP (pomijając całą otoczkę oraz brak parametrów) metoda do tworzenia dokumentu w przypadku błędy API lub wykonania kodu (parsowania klas etc) może zwrócić błędy z kodem do 100 lub powyżej.

```
$doc_id = NULL;

try {
   $doc_id = $client->createDocument(array());
   var_dump($doc_id);
} catch(SoapFault $fault) {

   var_dump($fault);

   if ($fault->faultcode < 100) {
       trigger_error("SOAP Fault: (faultcode: {$fault->faultcode}, faultstring: {$fault->faultstring})", E_USER_ERROR);
   }
}

Jak należy rozumieć ten kod:

- poniżej 100 - błędy techniczne wykonania kodu, logiki kodu, parsowania (błędy w kodzie) etc
- równe lub powyżej 100 - błędy walidacji API czyli wszystkie te błędy, które mogą pojawić się podczas sprawdzania paramet

Błędy z kodem poniżej 100 powinny (jest zalecane) przerywać działanie aplikacji zewnętrznej gdyż wyraźnie wskazują na niep

Obecnie mechanizm API zwraca błędy z kodami 100 lub 110, które wskazują na błąd parametrów a same kody błędów nie mają pom
```

Uwagi

Przekazywanie parametrów

We wszystkich funkcjach w których parametr jest określony jako (array)$data możliwe jest przekazanie parametrów jako ciąg JSON (więcej na http://www.json.org/). Dzięki temu nie potrzeba tworzyć pseudo struktur tablic asocjacyjnych dla języków programowania, w których takich typów nie ma.

Korzystanie z metod służących do wyszukiwania danych w kolumnach z datą np. searchContacts (od wersji 4.6.26, 4.7.2) W przypadku jeśli chcemy wyszukać obiekty w systemie np. klientów za pomocą metody searchContacts ale jako parametr wyszukiwania podajemy datę (format YYYY-MM-DD HH:MM lub YYYY-MM-DD) dodania (kolumna adddat) system umożliwia następujące sposoby wyszukiwania po dacie (wyszukuje klientów po kolumnie adddat):

- 2014-12-12 10:10 - wszyscy klienci dodani w dacie 2014-12-12 10:10 (przedział czasu od 10:10 do 10:11 kolumna z datą jest rzutowana bez sekund)
- 2014-12-12 - wszyscy klienci dodani w dacie 2014-12-12 (przedział wyszukiwania od 2014-12-12 00:00 do 2014-12-13 00:00)
- 2014-12-12 10:10:: - wszyscy klienci dodani w dacie 2014-12-12 10:10 lub później (przedział wyszukiwania od 2014-12-12 10:10 do teraz)

- 2014-12-12:: - wszyscy klienci dodani w dacie 2014-12-12 lub później (przedział wyszukiwania od 2014-12-12 00:00 do teraz)
- 2014-12-12 10:10::2014-12-12 - wszyscy klienci dodani w dacie 2014-12-12 10:10 do 2014-12-12 (przedział wyszukiwania od 2014-12-12 10:10 do 2014-12-13 00:00)
- ::2014-12-12 - wszyscy klienci dodani do 2014-12-12 (przedział wyszukiwania od początku świata do 2014-12-13 00:00)
- ::2014-12-12 10:10 - wszyscy klienci dodani do 2014-12-12 10:10 (przedział wyszukiwania od początku świata do 2014-12-12 10:10)

Na szczególną uwagę zasługuje modyfikator ::, który umożliwia definiowanie zakresów. Jeśli data wystąpi przed modyfikatorem wtedy jest uważana za początek zakresu. Jeśli po modyfikatorze wtedy za koniec zakresu.

Jeśli, któraś z dat (przed lub za modyfikatorem) nie zostanie podana (a sam modyfikator będzie dostępny w parametrze) wtedy system automatycznie dostosuje się do żądania o zakres.

W przypadku jeśli w parametrze data zostanie zdefiniowana bez godziny (YYYY-MM-DD) system automatycznie uzupełni godzinę według parametru i czy ma wyszukiwać po zakresie np.

- 2014-12-12:: uzupełni 2014-12-12 00:00:: - czyli szukam od początku dnia 2014-12-12
- ::2014-12-12 uzupełni 2014-12-13::00:00 - czyli szukam do końca dnia 2014-12-12 (aby uwzględnić wszystkie wpisy do dnia następuje automatyczne powiększenie zakresu do początku dnia następnego)

Dokumentacja API 2.2.0

Dokumentacja poszczególnych funkcji, parametrów oraz przykłady wywołań dostępne są poniżej:

```
http://{host}:{port}/apps/edokumenty/classes/eDokumentyApi/DokumentacjaAPI.txt
```

Dokumenty

- Utwórz nowy dokument
- Aktualizacja dokumentu
- Dodaj pracownika do uprawnionych w dokumencie
- Dodaj grupę do uprawnionych w dokumencie
- Dodaj załącznik do dokumentu
- Ustawia meta text załącznika (używany przy wyszukiwaniu)
- Tworzy nowy dokument typu raport dobowy z kasy
- Pobierz dane dotyczące typu dokumentu na podstawie jednego z poniższych parametrów
- Zarejestruj w dzienniku
- Wyszukaj dokument
- Pobierz dane dokumentu
- Dodaj pozycję do dokumentu (produkt)
- Pobierz pozycje dokumentu (produkty)
- Dodaje powiązanie dokumentu ze sprawą (bez ustawienia pola prc_id na dokumencie)

Produkty

- Dodanie nowego produktu
- Aktualizacja produktu
- Usuwanie produktu
- Wyszukanie produktów w bazie
- Pobierz dane produktu

Sprawy

- Dodanie nowej sprawy
- Aktualizacja sprawy
- Dodaj pracownika do uprawnionych w sprawie
- Dodaj grupę do uprawnionych w sprawie
- Pobierz dane o sprawie
- Wyszukanie sprawy w bazie

- Dodaj pozycję do sprawy (produkt)

Kontakty

- Dodanie nowego kontrahenta
- Aktualizuj dane kontrahenta
- Pobierz dane o kontrahencie
- Wyszukanie kontrahentów w bazie
- Pobierz dane adresu
- Dodanie nowej osoby kontaktowej
- Aktualizuj dane osoby kontaktowej
- Pobierz dane wszystkich osób kontaktowych wskazanego kontrahenta
- Łączenie klientów
- Pobierz dane osoby kontaktowej
- Tworzy konto pracownika dla klienta
- Aktualizuje konto pracownika przypisane do klienta

Zdarzenia

- Dodaj nowe zdarzenie
- Zaktualizuj zdarzenie
- Pobierz dane o zdarzeniu
- Zwraca listę zdarzeń według zadanych parametrów
- Dodaj załącznik do zdarzenia

Procedury

- Zwraca dane etapów danej procedury
- Wykonj dany etap procedury oraz aktywuj następny
- Aktywuj dany etap procedury

Zarządzanie użytkownikami i jednostkami

- Pobierz dane dotyczące jednostki organizacyjnej na podstawie jednego z poniższych parametrów
- Dodaje konto użytkownika systemu.
- Deaktywacja konta użytkownika
- Ponowna aktywacja konta użytkownika
- Zwraca dane użytkownika systemu.
- Zwraca dane grupy użytkowników
- Dodaje użytkownika do grupy
- Usuwa użytkownika z grupy
- Dodaje jednostkę organizacyjną.
- Zwraca dane jednostki organizacyjnej.
- Przypisuje użytkownika do jednostki organizacyjnej
- Zwalnia użytkownika ze stanowiska
- setUserPrivilages

Urządzenia

- Dodaj nowe urządzenie
- Dodaj lokalizację urządzenia
- Pobierz dane urządzenia
- Pobierz lokalizację urządzenia

Inne

- Wysyłanie powiadomień
- Ustawienie cechy

- Podpisz i wyślij dokument na emaila
- Wykonuje raport i zwraca jego wynik
- Publikacja teczki RWA w danej jednostce organizacyjnej
- Dodaj komentarz

MRD

- Pobranie definicji rejestru
- Dodanie/modyfikacja pozycji w rejestrze
- Dodanie załącznika do wpisu w rejestrze
- Pobranie wpisu z rejestru?
- Wyszukiwanie wpisów w rejestrze
- Pobranie daty ostatniej modyfikacji dla całego rejestru

Pliki

- Pobierz dane o pliku
- Nowa wersja pliku