Wikiprint Book

Title: Utwórz nowy dokument

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

Version: 12

Date: 07/16/26 11:47:43

Table of Contents

Utwórz nowy dokument *3*

Utwórz nowy dokument

Szczegóły dla poszczególnych typów dokumentów (wszystkie szczegółowe przykłady posiadają parametry jak w ogólnym przykładzie):

- Utwórz nowe pismo

Definicja parametrów:

```
<?php

/**
*  Tworzy nowy dokument
*
*  @param data Array - Tablica z parametrami
*  @param data['dscrpt'] String - opis dokumentu
*  @param data['adduid'] Int - Osoba dodająca dokument (users:usr_id)
*  @param data['dctpid'] Int - typ dokumentu (types_of_documents:dctpid)
*  @param data['dctptp'] String - typ dokumentu (types_of_documents:dctptp). Przykład: VatNote, Paper
*  @param data['target'] Int - identyfikator stanowiska (organization_units:orunid)
*  @param data['prc_id'] Int - identyfikator sprawy (processes:prc_id)
*  @param data['target_search'] String - fraza, która zostanie użyta do odszukania stanowiska na które dokument ma zostać
*  @param data['state_'] Int - rodzaj dokumenty (1-wychodzący, 2-przychodzący, 3-wewnętrzny)
*  @param data['number'] String- numer dokumentu
*  @param data['from_contact_symbol'] String - symbol nadawcy (contacts:symbol)
*  @param data['to_contact_symbol'] String - symbol odbiorcy (contacts:symbol)
*  @param data['symgen'] BOOLEAN - flaga czy ma generować numer dokumentu. Dotyczy tylko tych dokumentów, które na formatc
*  @param data ['from_contid'] int - id nadawcy (contacts.contid)
*  @param data ['to_contid'] int - id odbiorcy (contacts.contid)
*
*  @return Int - id dokumentu jeśli sukces (documents:doc_id), 0 w razie niepowodzenia
*
*  @throws Exception - SoapFault
*/
Int createDocument(Array data)

?>
```

Przykłady wywołań:

```
// Plik MyService.php umieszczony w apps/edokumenty.
// MyService.php
<?php

define('EDOK_API_LOGIN', 'developer');
define('EDOK_API_PASSWORD', 'developer');
define('DEFAULT_ENTITY_SYMBOL', 'demo');

require_once('./classes/eDokumentyApi/EDokApiClient.inc');

$options = array(
   'location' => 'http://{host}:{port}/eDokumentyApi.php',
   "uri" => "eDokumentyAPI",
   'encoding'=>'UTF-8'
);

$client = new EDokApiClient(NULL, $options);
$client->setUser(EDOK_API_LOGIN);
$client->setPass(md5(EDOK_API_PASSWORD));
$header = new SoapHeader('eDokumentyAPI', 'entity_symbol', DEFAULT_ENTITY_SYMBOL);
$client->__setSoapHeaders($header);
```

```php
$doc_id = NULL;

try {
   $data = array(
           'dscrpt' => 'SOAP TEST '.date('d H:m:s'),
           'prc_id' => 567,
           'dctpid' => 1,
           'target' => 54,
           'from_contact_symbol' => '12345',
           'to_contact_symbol' => '54321',
           );
   $doc_id = $client->createDocument($data);
   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);
   }
}

?>
```