

Title: Dodanie nowego produktu

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

Version: 7

Date: 02/04/26 19:33:35

Table of Contents

Dodanie nowego produktu

3

Dodanie nowego produktu

Definicja parametrów:

```

<?php

/**
 * Tworzy nowy produkt
 * W przypadku jeśli dany produkt już istnieje wyrzucany jest wyjątek
 *
 * @param data Array - Tablica z parametrami
 * @param data['symbol'] Text - symbol nowego produktu musi być unikalny w obrębie jednostki. Parametr ten jest wymagany
 * @param data['name__'] Text - nazwa
 * @param data['dscrpt'] Text - opis
 * @param data['pkwiu_'] Text - PKWiU
 * @param data['type__'] Int - typ produktu: 1 - usługa, 2 - towar
 * @param data['unitnm'] Text - nazwa jednostki obecnie obsługiwane: PIECE - sztuka, METER - metr,
 * METER2 - metr kwadratowy, KILOGRAM - kilogram, LITER - litr,
 * HOUR - godzina, KM - kilometr, SET - komplet,
 * METERC - metr bieżący, PACKAGE - pakiet, WRAP - opakowanie, REAM - ryza
 * @param data['pgr_id'] Int - grupa (product_groups.pgr_id)
 * @param data['spl_id'] Int - dostawca (contacts.contid)
 * @param data['netto_'] Numeric(12, 2) - cena netto produktu
 * @param data['cursmb'] Text - waluta (currency.cursmb)
 * @param data['vat_id'] Int - identyfikator stawki VAT (vat.vat_id)
 * @param data['acوريد'] Int - identyfikator jednostki rozliczeniowej (organization_units.orunid)
 * @param data['tovcid'] Int - identyfikator rodzaju kosztów (types_of_vcots.tovcid)
 * @param data['quantm'] Numeric(12, 2) - ilość na magazynie
 * @param data['quamin'] Numeric(12, 2) - ilość minimalna
 * @param data['prior_'] Int - liczba porządkowa
 *
 * @return Int - id produktu jeśli sukces (depository:depoid), 0 w razie niepowodzenia
 *
 * @throws Exception - SoapFault
 */
Int createProduct(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);

$depoid = NULL;

try {
    $data = array(
        'symbol' => 'SOAP TEST MKY-09',
        'name__' => 'Maszynka do golenia SOAP TEST',
        'type__' => 2,
        'unitnm' => 'PIECE',
        'pgr_id' => 1,
        'vat_id' => 1,
    );
    $depoid = $client->createProduct($data);
    var_dump($depoid);
} catch(SoapFault $fault) {
    var_dump($fault);

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

?>
```