

Title: Przewodnik wdrożeniowca > Dodaj nowe zdarzenie

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

Version: 9

Date: 04/11/25 09:23:14

Table of Contents

Przewodnik wdrożeniowca > Dodaj nowe zdarzenie

3

Przewodnik wdrożeniowca > Dodaj nowe zdarzenie

Definicja parametrów:

```
<?php

/**
 * Tworzy nowe zdarzenie
 *
 * @param data Array - Tablica z parametrami
 * @param data['summary'] String - temat zdarzenia
 * @param data['dscrpt'] String - opis zdarzenia
 * @param data['trmtyp'] String - typ zdarzenia (TODO - zadanie, PHONECALL - rozmowa tel., MEETING - spotkanie)
 * @param data['type__'] String - typ rozmowy tel. jeśli trmtyp = PHONECALL (IN - przychodząca, OUT - wychodząca, INT - we
 * @param data['start_'] String - data rozpoczęcia w formacie YYYY-MM-DD HH:MM:SS np 2010-01-01 08:00:00
 * @param data['duratn'] String - czas trwania np 2h30m - co oznacza 2 godziny 30 minut
 * @param data['usr_id'] Mixed - identyfikator pracownika (może być tablica identyfikatorów), któremu zleca się wykonanie
 * @param data['contid'] Int - identyfikator kontaktu (contacts:contid) pod którym zostanie utworzone zdarzenie
 * @param data['prc_id'] Int - identyfikator sprawy (processes:prc_id) do której zostanie dołączone zdarzenie
 * @param data['adduid'] Int - identyfikator pracownika (users:usr_id) który zleca zdarzenie, musi należeć do jednostki
 * @param data['evcaid'] Int - identyfikator kategorii zdarzenia właściwy dla danego typu patrz słownik kategorii zdarz
 * @param data['ccalid'] Int - identyfikator typu zdarzenia korporacyjnego dotyczy tylko zdarzeń tworzonych z parametre
 * @param data['witin'] bool - powiadom o wykonaniu zadania (true/false)
 *
 * @return Int - id event jeśli sukces (events:evntid), 0 w razie niepowodzenia
 *
 * @throws Exception - SoapFault
 */
Int createEvent(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);

$evntid = NULL;

try {
```

```
$data = array(
    'summary' => 'SOAP TEST',
    'trmtyp' => 'TODO',
    'start_' => date('Y-m-d H:i:s'),
    'duratn' => '2h 30m',
    'usr_id' => 2,
);
$evntid = $client->createEvent($data);
var_dump($evntid);
} catch(SoapFault $fault) {
    var_dump($fault);

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