**Wikiprint Book**

**Title: Przewodnik wdrożeniowca > Zaktualizuj zdarzenie**

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

**Version: 2**

**Date: 10/08/25 03:55:56**

## Table of Contents

*Przewodnik wdrożeniowca > Zaktualizuj zdarzenie* *3*

## Przewodnik wdrożeniowca > Zaktualizuj zdarzenie

**Definicja parametrów:**

```
<?php

/**
*  Aktualizuje dane zdarzenia
*
*  @param data Array            - Tablica z parametrami
*  @param data['evntid'] Int    - ID zdarzenia
*  @param data['sumary'] String - temat zdarzenia
*  @param data['dscrpt'] String - opis zdarzenia
*  @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['end___'] String - data zakończenia 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['ccalid'] Int    - identyfikator typu zdarzenia korporacyjnego dotyczy tylko zdarzeń tworzonych z parametre
*  @param data['prior_'] Int    - priorytet
*  @param data['witinf'] bool   - powiadom o wykonaniu zadania (true/false)
*  @param data['on_tod'] bool   - plan dnia (true/false)
*
*  @return bool - TRUE jeżeli operacja zakończyla się powodzeniem
*
*  @throws Exception - SoapFault
*/
bool updateEvent(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);

try {
   $data = array(
```

```
        'evntid' => 123,
        'start_' => date('Y-m-d H:i:s'),
    );
    $result = $client->updateEvent($data);
    var_dump($result);
} catch(SoapFault $fault) {
    var_dump($fault);

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

?>
```