

[Przewodnik wdrożeniowca](#) > Zaktualizuj dane dokumentu

Definicja parametrów:

```

<?php

/**
 * Aktualizuje dane istniejącego dokumentu
 *
 * @param data Array - Tablica z parametrami
 * @param data['doc_id'] int - ID dokumentu (wymagane)
 * @param data['dscrpt'] String - opis pisma
 * @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['state_'] Int - rodzaj dokumenty (1-wychodzący, 2-przychodzący, 3-wewnętrzny)
 * @param data['from_contact_symbol'] String - symbol nadawcy (contacts:symbol)
 * @param data['to_contact_symbol'] String - symbol odbiorcy (contacts:symbol)
 * * @param data Array - Tablica z parametrami
 * @param data['dscrpt'] String - opis pisma
 * @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['state_'] Int - rodzaj dokumenty (1-wychodzący, 2-przychodzący, 3-wewnętrzny)
 * @param data['from_contact_symbol'] String - symbol nadawcy (contacts:symbol)
 * @param data['to_contact_symbol'] String - symbol odbiorcy (contacts:symbol)
 * @param data['fixinf'] String - uwagi
 * @param data['tpstid'] Int - identyfikator statusu (types_of_processes_states:tpstid). Jeśli statusu przypisano dla pism
 * @param data['prtpid'] Int - identyfikator procedury (procedures_def:prtpid)
 *
 * @return bool - TRUE jeżeli operacja zakończyła się powodzeniem
 *
 * @throws Exception - SoapFault
 */
Int updateDocument(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(
        'doc_id' => 1221,
        'prc_id' => 567,
    );
    $result = $client->updateDocument($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);
    }
}

?>
```