

Title: Wyszukiwanie wpisów w rejestrze

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

Version: 7

Date: 11/21/24 20:42:17

## Table of Contents

*Wyszukiwanie wpisów w rejestrze*

3

## Wyszukiwanie wpisów w rejestrze

### Definicja parametrów:

```
<?php

/**
 * Wyszukuje wpisy w podanym rejestrze
 * Dla typów danych innych niż boolean wyszukiwanie jest tekstowe (wielkość liter nie ma znaczenia).
 * przykład:
 * {"ulica":"Stefana batorego"} w SQLu będzie równoznaczne z (ulica ~* 'Stefana' AND ulica ~* 'batorego')
 *
 * @param integer Id rejestru
 * @param string fields pola z rejestru (jedno lub więcej, wymienione po przecinku), których wartości mają zostać zwrócone
 * @param integer limit maksymalna ilość zwracanych wpisów w rejestrze (0 = bez limitu)
 * @param integer offset od którego rekordu mają zostać zwrócone dane (działa jeżeli określono limit większy od 0)
 * @param string json params dane do wyszukiwania (np. {"imie":"jan","nazwisko":"nowak"})
 *
 * @access public
 * @return string json
 */
public function searchRegisterEntries($register_id, $fields, $limit, $offset, $params)

?>
```

### 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(
        'name__' => 'oskar_reg_2',
    );
    $reg_data = $client->getRegister($data);
    var_dump($reg_data);
} catch(SoapFault $fault) {

    var_dump($fault);
}
```

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

try {
    $data = array(
        'imie' => 'jan',
    );
    $entries = $client->searchRegisterEntries($reg_data['id____'], 'id____,imie,nazwisko', 100, 0, $data);
    var_dump($entries);
} catch(SoapFault $fault) {

    var_dump($fault);

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

?>
```

Powyższy kod udostępniony jest na licencji LGPL <http://www.gnu.org/licenses/lgpl-3.0.txt>