

[Przewodnik wdrożeniowca](#) > eDokumentyApi

System eDokumenty udostępnia API (Application Programming Interface) dzięki któremu jesteśmy w stanie przeprowadzać podstawowe czynności bez konieczności logowania do systemu.

Usługa jest zabezpieczona rozszerzeniem WSSecurity protokołu SOAP i wymaga podania w nagłówkach wywołania XML nazwy użytkownika i hasła.

Hasło i użytkownik to specjalne dane, które należy wprowadzić do pliku config.inc pod kluczami:

```
<?php
define('EDOK_API_LOGIN', 'edok_api_user');
define('EDOK_API_PASSWORD', 'edok_api_pass');

// Dodatkowa stała która umożliwia pominięcie autentykacji (jeśli ustawiamy na FALSE to nie sprawdza danych EDOK_API_LOGIN
// domyślnie ustawiona na TRUE, można wysłać GET lub dodatkowe nagłówki WSS
// define('EDOK_API_AUTH_MODE', TRUE);
?>
```

Wartości stałych w powyższym przykładzie konfiguracji są tylko danymi prezentacyjnymi i nie powinno się ich używać na produkcyjnej bazie.

Stale te mogą mieć dowolne wartości ważne jednak aby te same wartości podać przy wywołaniu usługi SOAP w kliencie.

Usługa jest dostępna pod adresem:

```
http://{host}:{port}/apps/edokumenty/classes/eDokumentyApi/EDokApiServer.php
```

Opcjonalnie od wersji 0.9.8 usługa jest dostępna pod skróconym adresem:

```
http://{host}:{port}/eDokumentyApi.php
```

Wartość {host} oraz {port} należy zamienić odpowiednimi wartościami zgodnymi z konfiguracją serwera instalacyjnego systemu eDokumenty.

Dodatkowo od wersji systemu 3.3 autentykacja nie wymaga nadpisania klienta Soap. Wystarczy do url z adresem serwisu dodać parametry GET w postaci

```
<?php
// a1 - login,
// a2 - hasło,
// a3 - symbol jednostki
/*
$ops = array(
    'location' => 'http://{host}:{port}eDokumentyApi.php?a1=' . EDOK_API_LOGIN . '&a2=' . md5(md5(EDOK_API_PASSWORD) . '_SOAP_eDok...',
    'uri' => "eDokumentyAPI",
    'encoding' => 'UTF-8'
);
?>
```

Przykład wywołania serwisu oraz funkcji dodającej kontrahenta:

```
// Plik MyService.php umieszczony w apps/edokumenty.
// MyService.php
<?php
// stałe są w config.inc i są to dane potrzebne do autentykacji usługi
require_once('../framework/lib/util/Translator/Translator.inc');
require_once('./config.inc');

// klient SOAP systemu eDokumenty wysyłający nagłówki WSSecurity zmienić na
// odpowiednią ścieżkę w zależności od lokalizacji pliku MyService.php
require_once('./classes/eDokumentyApi/EDokApiClient.inc');
```

```

// wartość {host}:{port} zmienić na odpowiednią dla serwera
$ops = array(
    'location' => 'http://{host}:{port}eDokumentyApi.php',
    'uri' => "eDokumentyAPI",
    'encoding'=>'UTF-8'
);

// Dodatkowo od wersji 3.3 autentykacja nie wymaga nadpisania klienta Soap. Wystarczy do url z adresem serwisu dodać param
// a1 - login,
// a2 - hasło,
// a3 - symbol jednostki
/*
$ops = array(
    'location' => 'http://{host}:{port}eDokumentyApi.php?a1='.EDOK_API_LOGIN.'&a2='.md5(md5(EDOK_API_PASSWORD).'_SOAP_eDok
    'uri' => "eDokumentyAPI",
    'encoding'=>'UTF-8'
);

// bezpośrednie wywołanie klienta Soap
$client = new SoapClient(NULL, $ops);
*/
$client = new EDokApiClient(null, $ops);
$client->setUser(EDOK_API_LOGIN); // ten sam co w config.inc
$client->setPass(md5(EDOK_API_PASSWORD)); // to samo co w config.inc

$header = new SoapHeader('eDokumentyAPI', 'entity_symbol', DEFAULT_ENTITY_SYMBOL);
$client->__setSoapHeaders($header);

// Tworzy kontakt
$contid = NULL;

try {
    $data = array(
        'name_1' => 'SOAP TEST'.date('d H:m:s'),
        'name_2' => 'SOAPEST',
        'nip__' => 1111111111,
        'street' => 'Główna',
        'symbol' => 'FGH99'
    );
    $contid = $client->createContact($data);
    var_dump($contid);
} catch(SoapFault $fault) {

    var_dump($fault);

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

?>

```

WAŻNE Hasło powinno być zakodowane przy użyciu funkcji md5.

Dokumentacja poszczególnych funkcji, parametrów oraz przykłady wywołań dostępne są poniżej:

<http://{host}:{port}/apps/edokumenty/classes/eDokumentyApi/DokumentacjaAPI.txt>

Dokumentacja API

Dokumenty

- [Utwórz nowy dokument](#)
- [Dodaj pracownika do uprawnionych w dokumencie](#)
- [Dodaj grupę do uprawnionych w dokumencie](#)
- [Dodaj załącznik do dokumentu](#)
- [Ustawia meta text załącznika \(używany przy wyszukiwaniu\)](#)
- [Tworzy nowy dokument typu raport dobowy z kasy](#)
- [Pobierz dane dotyczące typu dokumentu na podstawie jednego z poniższych parametrów](#)

Produkty

- [Dodanie nowego produktu](#)
- [Aktualizacja produktu](#)
- [Usuwanie produktu](#)

Sprawy

- [Dodanie nowej sprawy](#)
- [Pobierz dane o sprawie](#)

Kontakty

- [Dodanie nowego kontrahenta](#)
- [Pobierz dane o kontrahencie](#)
- [Wyszukanie kontrahentów w bazie](#)
- [Dodanie nowej osoby kontaktowej](#)
- [Aktualizuj dane osoby kontaktowej](#)
- [Pobierz dane wszystkich osób kontaktowych wskazanego kontrahenta](#)

Zdarzenia

- [Dodaj nowe zdarzenie](#)
- [Pobierz dane o zdarzeniu](#)
- [Zwraca listę zdarzeń według zadanych parametrów](#)

Procedury

- [Zwraca dane etapów danej procedury](#)
- [Wykonaj dany etap procedury oraz aktywuj następny](#)

Inne

- [Pobierz dane dotyczące jednostki organizacyjnej na podstawie jednego z poniższych parametrów](#)
- [Ustawienie cechy](#)

```
// Plik MyService.php umieszczony w apps/edokumenty.  
// MyService.php  
<?php  
// stałe są w config.inc i są to dane potrzebne do autentykacji usługi  
require_once(' ../../framework/lib/util/Translator/Translator.inc');  
require_once(' ./config.inc');  
  
// klient SOAP systemu eDokumenty wysyłający nagłówki WSSecurity zmienić na  
// odpowiednią ścieżkę w zależności od lokalizacji pliku MyService.php  
require_once(' ./classes/eDokumentyApi/EDokApiClient.inc');
```

```

// wartość {host}:{port} zmienić na odpowiednią dla serwera
$ops = array(
    'location' => 'http://{host}:{port}eDokumentyApi.php',
    'uri' => "eDokumentyAPI",
    'encoding'=>'UTF-8'
);

// Dodatkowo od wersji 3.3 autentykacja nie wymaga nadpisania klienta Soap. Wystarczy do url z adresem serwisu dodać param
// a1 - login,
// a2 - hasło,
// a3 - symbol jednostki
/*
$ops = array(
    'location' => 'http://{host}:{port}eDokumentyApi.php?a1='.EDOK_API_LOGIN.'&a2='.md5(md5(EDOK_API_PASSWORD)).'_SOAP_eDok_
    'uri' => "eDokumentyAPI",
    'encoding'=>'UTF-8'
);

// bezpośrednie wywołanie klienta Soap
$client = new SoapClient(NULL, $ops);
*/
$client = new EDokApiClient(null, $ops);
$client->setUser(EDOK_API_LOGIN); // ten sam co w config.inc
$client->setPass(md5(EDOK_API_PASSWORD)); // to samo co w config.inc

$header = new SoapHeader('eDokumentyAPI', 'entity_symbol', DEFAULT_ENTITY_SYMBOL);
$client->__setSoapHeaders($header);

// Tworzy kontakt
$contid = NULL;

try {
    $data = array(
        'name_1' => 'SOAP TEST'.date('d H:m:s'),
        'name_2' => 'SOAPEST',
        'nip__' => 111111111,
        'street' => 'Główna',
        'symbol' => 'FGH99'
    );
    $contid = $client->createContact($data);
    var_dump($contid);
} catch(SoapFault $fault) {

    var_dump($fault);

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

// Tworzy sprawę
$prc_id = NULL;

try {
    $data = array(
        'dscprt' => 'SOAP TEST '.date('d H:m:s'),
        'briefcase_symbol' => 'DK.023',
        'orunid' => 49,
        'rspoid' => 54,

```

```

        'fxtrid' => 60,
        'prtpid' => 1,
    );
    $prc_id = $client->createProcess($data);
    var_dump($prc_id);
} catch(SoapFault $fault) {

    var_dump($fault);

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

// tworzy dokument w sprawie
if ($prc_id) {

    $doc_id = NULL;

    try {
        $data = array(
            'dscprt' => 'SOAP TEST '.date('d H:m:s'),
            'prc_id' => $prc_id,
            'dctpid' => 1,
            'target' => 54,
            'from_contact_symbol' => '12345',
            'to_contact_symbol' => '54321',
        );
        $doc_id = $client->createDocument($data);
        var_dump($doc_id);

    } catch(SoapFault $fault) {

        var_dump($fault);

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

    // dodaje załącznik do utworzonego dokumentu
    // o treści Testowy plik i nazwie dokument.txt
    if ($doc_id) {
        try {
            $filePath = './test.pdf';
            $content = base64_encode(file_get_contents(filePath));

            $dd = $client->addAttachmentToDocument($content, basename(filePath), $doc_id);
            var_dump($dd);

        } catch(SoapFault $fault) {

            var_dump($fault);

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

```

```
// zwraca dane (tablica) o sprawie wszystko z tabeli processes gdzie prc_id = $prc_id
try {

    $data = $client->getProcess($prc_id);
    var_dump($data);

} catch(SoapFault $fault) {

    var_dump($fault);

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

?>
```

W przypadku jeśli chcemy utworzyć własnego klienta SOAP dla innego języka programowania np. JAVA należy skontaktować się z działem oprogramowania celem wyszczególnienia specyfikacji takiego klienta.