



Dokumentacja formularz dodawania kontaktów

Spis treści

Wprowadzenie.....	3
Omówienie dostępnych metod.....	3
dodajGrupe.....	3
dodajKontakt.....	4
listaGrup.....	4
listaKontaktow.....	5
pobierzIdentyfikatory.....	6
przetworzPrzezND.....	6
setNadawca.....	7
setNDI.....	7
setPotwierdzenieSMS.....	8
setTrescKodSMS.....	8
setUsuniecieSMS.....	8
sprawdzKod.....	9
usunGrupe.....	9
usunKontakt.....	10
usunNumer.....	10
weryfikacjaSMS.....	10
wyslijPotwierdzenieDopisania.....	11
wyslijPotwierdzenieUsuniecia.....	11
wyslijSMS.....	12
Przykładowe skrypty.....	12
1. Formularz bez weryfikacji SMS.....	12
2. Weryfikacja SMS oraz wykorzystanie NDI.....	13
3. Weryfikacja SMS oraz zapis do kilku grup jednocześnie.....	14
Historia zmian.....	16

Wprowadzenie

Przedmiotem niniejszej dokumentacji jest klasa `SerwerSMS_Kontakty.class.php`, umożliwiającej przetworzenie danych z formularza na stronie WWW. Zawiera ona zbiór metod wspomagających oprogramowanie mechanizmu dopisywania numerów telefonów do swoich grup kontaktów w Panelu Klienta Serwer SMS. Omawiana klasa do działania wymaga podania loginu i hasła do konta w `SerwerSMS.pl`. Utworzenie obiektu wygląda następująco:

```
<?php
require_once("SerwerSMS_Kontakty.class.php");

$SerwerSMS = new SerwerSMS_Kontakty('login','haslo');

?>
```

Konstruktor klasy przyjmuje trzeci parametr, który jest nieobowiązkowy i powoduje włączenie trybu testowego. Tryb ten charakteryzuje się tym, że system podczas wysyłki zwraca poprawną odpowiedź, tak jak w przypadku normalnej wysyłki wiadomości, przy czym nie powoduje fizycznego jej wysłania i nie narusza stanu konta SMS. Utworzenie obiektu w trybie testowym wzbogacone jest o trzeci parametr o wartości **true**:

```
$SerwerSMS = new SerwerSMS_Kontakty('login','haslo', true);
```

Tryb weryfikacji numeru za pomocą kodu przesyłanego drogą SMS-ową wymaga uruchomienia sesji przeglądarki za pomocą poniższej komendy:

```
session_start();
```

Omówienie dostępnych metod

Programista ma do swojej dyspozycji kilka metod spełniających określone zadanie i pozwalające w łatwy sposób na obsługę danych.

dodajGrupe

Dodanie nowej grupy kontaktów w Panelu Klienta.

mixed `dodajGrupe`(string **\$nazwa**);

Przyjmowane wartości:

`nazwa`

Ciąg znaków będący nazwą nowej grupy.

Zwracane wartości:

Zwraca tablicę z id grupy oraz stanem w przypadku poprawnego utworzenia grupy lub wartość **false** w przypadku błędu.

```
Array
(
    [id] => 00000001
    [stan] => OK
)
```

Przykład:

```
<?php
require_once("SerwerSMS_Kontakty.class.php");

$SerwerSMS = new SerwerSMS_Kontakty("login","haslo");
$nowa = $SerwerSMS->dodajGrupe('Nowa grupa');

if(is_array($nowa){
    echo $nowa['id'];
    echo $nowa['stan'];
} elseif(!$nowa) {
    echo 'Wystąpił błąd';
}

?>
```

dodajKontakt

Dodanie nowego kontaktu do grupy w Panelu Klienta. Możliwe jest dodanie jednego numeru do kilku grup.

mixed **dodajKontakt**(array \$grupy, string \$numer, string \$email, string \$imie, string \$nazwisko, string \$firma);

Przyjmowane wartości:

grupy

Identyfikator pojedynczej grupy lub tablica z wieloma identyfikatorami grup. Numery te widoczne są w Panelu Klienta, po wybraniu edycji danej grupy.

numer

Ciąg znaków będących numerem telefonu.

email

Ciąg znaków będących adresem e-mail (parametr nieobowiązkowy).

imie

Ciąg znaków zapisywanych w polu imię (parametr nieobowiązkowy).

nazwisko

Ciąg znaków zapisywanych w polu nazwisko (parametr nieobowiązkowy).

firma

Ciąg znaków zapisywanych w polu firma (parametr nieobowiązkowy).

Zwracane wartości:

Zwraca tablicę z id kontaktu oraz stanem w przypadku poprawnego utworzenia kontaktu lub wartość *false* w przypadku błędu.

```
Array
(
    [id] => 00000001
    [stan] => OK
)
```

Przykład:

```
<?php

require_once("SerwerSMS_Kontakty.class.php");

$SerwerSMS = new SerwerSMS_Kontakty("login","haslo");

// dodanie kontaktu do kilku grup, z pominięciem pola imię oraz nazwisko
$grupy = array('00000001','00000002');
$kontakt = $SerwerSMS->dodajKontakt($grupy,'500600700','adres@email.com',null,null,'SerwerSMS.pl');

// dodanie kontaktu do jednej grupy, z pominięciem pola e-mail oraz firma
$kontakt = $SerwerSMS->dodajKontakt('00000001','500600700',null,'Jan','Kowalski');

if(is_array($kontakt){

    echo $kontakt['id'];
    echo $kontakt['stan'];

} elseif(!$kontakt) {

    echo 'Wystąpił błąd';

}

?>
```

listaGrup

Pobranie listy identyfikatorów wszystkich grup w kontaktach użytkownika.

array **listaGrup**(void);

Przyjmowane wartości:

Brak przyjmowanych wartości.

Zwracane wartości:

Zwraca tablicę dwuwymiarową z nazwą grupy, identyfikatorem oraz liczbą kontaktów np.:

```
Array
```

```
(
    [0] => Array
        (
            [nazwa] => Nazwa grupy 1
            [id] => 00000001
            [ilosc] => 0
        )

    [1] => Array
        (
            [nazwa] => Nazwa grupy 2
            [id] => 00000002
            [ilosc] => 24
        )

    ...
)
```

Przykład:

```
<?php
require_once("SerwerSMS_Kontakty.class.php");

$SerwerSMS = new SerwerSMS_Kontakty('login','haslo');
$lista = $SerwerSMS->listaGrup();

foreach($lista as $grupa){
    echo $grupa['nazwa'].'. '. $grupa['id'].'. '. $grupa['ilosc'];
}

?>
```

listaKontaktow

Pobranie listy kontaktów z wybranej grupy lub jednego numeru z wszystkich lub konkretnej grupy.

```
array listaKontaktow(string $numer, int $grupa);
```

Przyjmowane wartości:

numer

Ciąg znaków będący numerem telefonu, który ma zostać wyszukany w grupach.

grupa

Identyfikator grupy, której kontakty mają zostać zwrócone. Numer ten widoczny jest w Panelu Klienta, po wybraniu edycji danej grupy.

Zwracane wartości:

Zwraca tablicę wielowymiarową z parametrami dotyczącymi grupy oraz kontaktów w niej znajdujących się.

```
Array
(
    [0] => Array
        (
            [grupa] => Nazwa grupy 1
            [id] => 00000001
            [ilosc] => 2
            [kontakty] => Array
                (
                    [0] => Array
                        (
                            [id] => 000000010
                            [telefon] => +48500600700
                            [email] => email@example.com
                            [firma] => Nazwa firmy
                            [imie] => Imie
                            [nazwisko] => Nazwisko
                        )
                    ...
                )
            ...
        )
    ...
)
```

Przykład:

```
<?php
require_once("SerwerSMS_Kontakty.class.php");

$SerwerSMS = new SerwerSMS_Kontakty("login","haslo");

// Listowanie wszystkich kontaktów w grupie
$lista = $SerwerSMS->listaKontaktow(null,'00000001');

// Wyszukiwanie numeru
$lista = $SerwerSMS->listaKontaktow('500600700');
```

```
// Wyszukiwanie numeru wewnątrz konkretnej grupy
$lista = $SerwerSMS->listaKontaktow('500600700','00000001');

foreach($lista as $grupa){
    echo $grupa['grupa'].' ', '.
    $grupa['id'].' ', '.
    $grupa['ilosc'];

    foreach($grupa['kontakty'] as $kontakt){
        echo $kontakt['id'].' ', '.
        $kontakt['telefon'].' ', '.
        $kontakt['email'].' ', '.
        $kontakt['firma'].' ', '.
        $kontakt['imie'].' ', '.
        $kontakt['nazwisko'];
    }
}

?>
```

pobierzIdentyfikatory

Pobiera listę identyfikatorów kontaktów w Panelu Klienta.

```
array pobierzIdentyfikatory(string $numer, int $grupa);
```

Przyjmowane wartości:

numer

Ciąg znaków będący numerem telefonu, który ma zostać wyszukany w grupach.

grupa

Identyfikator grupy, której kontakty mają zostać zwrócone. Numer ten widoczny jest w Panelu Klienta, po wybraniu edycji danej grupy.

Zwracane wartości:

Zwraca tablicę z identyfikatorami kontaktów.

```
Array
(
    [0] => 000000001
    ...
)
```

Przykład:

```
<?php

require_once("SerwerSMS_Kontakty.class.php");

$SerwerSMS = new SerwerSMS_Kontakty("login","haslo");

// Wyszukiwanie numeru
$lista = $SerwerSMS->pobierzIdentyfikatory('500600700');

// Wyszukiwanie numeru wewnątrz konkretnej grupy
$lista = $SerwerSMS->pobierzIdentyfikatory('500600700','00000001');

foreach($lista as $id){
    echo $id;
}

?>
```

przetworzPrzezND

Przetwarza treść wiadomości w taki sam sposób jak została przetworzona poprzez wysłanie jej z telefonu komórkowego. Metoda wykorzystuje konfigurację Numeru Dostępowego z Panelu Klienta. Domyślnie wiadomość przetwarzana jest przez numer współdzielony. Aby przetworzyć treść przez numer indywidualny, w pierwszej kolejności należy skorzystać z metody *setNDI()*.

```
bool przetworzPrzezND(string $numer, string $haslo_dostepowe);
```

Przyjmowane wartości:

numer

Ciąg znaków będący numerem telefonu nadawcy wiadomości.

haslo_dostepowe

Treść wiadomości SMS będąca hasłem dostępowym.

Zwracane wartości:

Zwraca wartość *true* lub *false* w zależności od powodzenia operacji.

Przykład:

```
<?php
require_once("SerwerSMS_Kontakty.class.php");

$SerwerSMS = new SerwerSMS_Kontakty('login','haslo');
$SerwerSMS->setNDI('+48700700700');
$wynik = $SerwerSMS->przetworzPrzezND('500600700', 'HASLO');

if($wynik){
    echo 'Wiadomość została poprawnie przetworzona';
} else {
    echo 'Wystąpił błąd';
}

?>
```

setNadawca

Ustawienie nagłówka nadawcy wiadomości SMS. Nazwa musi być wcześniej zgłoszona i zatwierdzona w Panelu Klienta. Jeśli pole nadawcy pozostanie puste, wiadomość SMS wysłana zostanie jako typ ECO.

```
void setNadawca(string $string);
```

Przyjmowane wartości:

string

Ciąg nieprzekraczający 11 znaków, będący prawidłowo zgłoszoną nazwą nadawcy w Panelu Klienta.

Zwracane wartości:

Brak zwracanych wartości.

Przykład:

```
<?php
require_once("SerwerSMS_Kontakty.class.php");

$SerwerSMS = new SerwerSMS_Kontakty('login','haslo');
$SerwerSMS->setNadawca('INFORMACJA');

?>
```

setNDI

Ustawienie Numeru Dostępowego Indywidualnego. Numer musi być przypisany do konta użytkownika SerwerSMS. Jeśli wartość ta pozostanie pusta, wiadomości przekazane przez metodę *przetworzPrzezND()*, wysyłane będą na współdzielony numer dostępowy o numerze 503 340 503.

```
void setNDI(string $numer);
```

Przyjmowane wartości:

numer

Ciąg będący 9-cio cyfrowym Numerem Dostępowym Indywidualnym, lub 4-cyfrowym numerem skróconym (ShortCode).

Zwracane wartości:

Brak zwracanych wartości.

Przykład:

```
<?php
require_once("SerwerSMS_Kontakty.class.php");

$SerwerSMS = new SerwerSMS_Kontakty('login','haslo');
$SerwerSMS->setNDI('+48500600700');

?>
```

setPotwierdzenieSMS

Ustawienie treści wiadomości potwierdzającej dodanie numeru do kontaktów. Wiadomość wysyłana jest na numer wprowadzony w formularzu.

```
void setPotwierdzenieSMS(string $string);
```

Przyjmowane wartości:

string

Ciąg znaków będących treścią wiadomości SMS. Wartość domyślna to: „Twój numer został poprawnie zapisany do bazy kontaktów.”

Zwracane wartości:

Brak zwracanych wartości.

Przykład:

```
<?php  
  
require_once("SerwerSMS_Kontakty.class.php");  
  
$SerwerSMS = new SerwerSMS_Kontakty('login','haslo');  
$SerwerSMS->setPotwierdzenieSMS('To jest wiadomosc potwierdzajaca zapisanie numeru w bazie kontaktow.');
```

setTrescKodSMS

Ustawienie treści wiadomości przesyłanej w momencie weryfikacji numeru telefonu kodem SMS. Kod dołączany jest na końcu treści.

```
void setTrescKodSMS(string $string);
```

Przyjmowane wartości:

string

Ciąg znaków będący treścią wiadomości SMS. Wartość domyślna to: „Twój kod potwierdzający to:”

Zwracane wartości:

Brak zwracanych wartości.

Przykład:

```
<?php  
  
require_once("SerwerSMS_Kontakty.class.php");  
  
$SerwerSMS = new SerwerSMS_Kontakty('login','haslo');  
$SerwerSMS->setTrescKodSMS('Twój kod potwierdzający to:');
```

setUsuniecieSMS

Treść wiadomości potwierdzającej usunięcie numeru z bazy kontaktów. Wiadomość wysyłana jest na numer wprowadzony w formularzu.

```
void setUsuniecieSMS(string $string);
```

Przyjmowane wartości:

string

Ciąg znaków będący treścią wiadomości SMS. Wartość domyślna to: „Twój numer został poprawnie usunięty z bazy kontaktów”

Zwracane wartości:

Brak zwracanych wartości.

Przykład:

```
<?php
require_once("SerwerSMS_Kontakty.class.php");

$SerwerSMS = new SerwerSMS_Kontakty('login','haslo');
$SerwerSMS->setUsuniecieSMS('To jest tresc wiadomosci potwierdzajacej usuniecie numeru z bazy kontaktow');

?>
```

sprawdzKod

Porównuje podany kod, z kodem zapisanym w sesji przeglądarki użytkownika.

bool **sprawdzKod**(string \$kod) ;

Przyjmowane wartości:

kod

Ciąg znaków będący kodem SMS.

Zwracane wartości:

Zwraca wartość *true* lub *false* w zależności od powodzenia operacji.

Przykład:

```
<?php
require_once("SerwerSMS_Kontakty.class.php");

$SerwerSMS = new SerwerSMS_Kontakty('login','haslo');
$wynik = $SerwerSMS->sprawdzKod('15d8es');

if($wynik){
    echo 'Kod jest poprawny';
} else {
    echo 'Kod jest niepoprawny';
}

?>
```

usunGrupe

Usuwanie grupy o podanym identyfikatorze z Panelu Klienta.

bool **usunGrupe**(int \$grupa) ;

Przyjmowane wartości:

grupa

Identyfikator grupy do usunięcia. Numer ten widoczny jest w Panelu Klienta, po wybraniu edycji danej grupy.

Zwracane wartości:

Zwraca wartość *true* lub *false* w zależności od powodzenia operacji.

Przykład:

```
<?php
require_once("SerwerSMS_Kontakty.class.php");

$SerwerSMS = new SerwerSMS_Kontakty("login","haslo");

$wynik = $SerwerSMS->usunGrupe('00000001');

if($wynik){
    echo 'Grupa została poprawnie usunięta';
} else {
    echo 'Wystąpił błąd';
}
```

```
}  
?>
```

usunKontakt

Usuwanie kontaktu o podanym identyfikatorze z Panelu Klienta.

```
bool usunKontakt(int $kontakt);
```

Przyjmowane wartości:

kontakt

Identyfikator kontaktu do usunięcia. Numer ten można pobrać za pomocą metody *pobierzIdentyfikator()*;

Zwracane wartości:

Zwraca wartość *true* lub *false* w zależności od powodzenia operacji.

Przykład:

```
<?php  
  
require_once("SerwerSMS_Kontakty.class.php");  
  
$SerwerSMS = new SerwerSMS_Kontakty("login","haslo");  
  
$wynik = $SerwerSMS->usunKontakt('00000001');  
  
if($wynik){  
    echo 'Kontakt został poprawnie usunięty';  
} else {  
    echo 'Wystąpił błąd';  
}  
  
?>
```

usunNumer

Usuwanie wszystkich kontaktów o podanym numerze telefonu z wszystkich grup.

```
bool usunNumer(string $numer);
```

Przyjmowane wartości:

numer

Numer telefonu do usunięcia.

Zwracane wartości:

Zwraca wartość *true* lub *false* w zależności od powodzenia operacji.

Przykład:

```
<?php  
  
require_once("SerwerSMS_Kontakty.class.php");  
  
$SerwerSMS = new SerwerSMS_Kontakty("login","haslo");  
  
$wynik = $SerwerSMS->usunNumer('500600700');  
  
if($wynik){  
    echo 'Numer został poprawnie usunięty';  
} else {  
    echo 'Wystąpił błąd';  
}  
  
?>
```

weryfikacjaSMS

Generuje kod SMS, zapisuje go w sesji użytkownika oraz wysyła wiadomość SMS z wygenerowanym kodem na podany numer telefonu.

bool **weryfikacjaSMS**(string \$numer);

Przyjmowane wartości:

numer

Ciąg znaków będący numerem telefonu, na który ma zostać wysłana wiadomość.

Zwracane wartości:

Zwraca wartość *true* lub *false* w zależności od powodzenia operacji.

Przykład:

```
<?php
require_once("SerwerSMS_Kontakty.class.php");

$SerwerSMS = new SerwerSMS_Kontakty('login','haslo');
$wynik = $SerwerSMS->weryfikacjaSMS('+48500600700');

if($wynik){
    echo 'Wiadomość została wysłana';
} else {
    echo 'Wystąpił błąd';
}

?>
```

wyslijPotwierdzenieDopisania

Wysyła wiadomość SMS na numer podany w argumencie, o treści ustawionej przez metodę *setPotwierdzenieSMS()*;

bool **wyslijPotwierdzenieDopisania**(string \$numer);

Przyjmowane wartości:

numer

Ciąg znaków będący numerem telefonu, na który ma zostać wysłane potwierdzenie.

Zwracane wartości:

Zwraca wartość *true* lub *false* w zależności od powodzenia operacji.

Przykład:

```
<?php
require_once("SerwerSMS_Kontakty.class.php");

$SerwerSMS = new SerwerSMS_Kontakty('login','haslo');
$wynik = $SerwerSMS->wyslijPotwierdzenieDopisania('+48500600700');

if($wynik){
    echo 'Wiadomość została wysłana';
} else {
    echo 'Wystąpił błąd';
}

?>
```

wyslijPotwierdzenieUsuniecia

Wysyła wiadomości na podany numer o treści ustawionej za pomocą metody *setUsuniecieSMS()*;

bool **wyslijPotwierdzenieUsuniecia**(string \$numer);

Przyjmowane wartości:

numer

Ciąg znaków będący numerem telefonu, na który ma zostać wysłane potwierdzenie.

Zwracane wartości:

Zwraca wartość *true* lub *false* w zależności od powodzenia operacji.

Przykład:

```
<?php
require_once("SerwerSMS_Kontakty.class.php");

$SerwerSMS = new SerwerSMS_Kontakty('login','haslo');
$wynik = $SerwerSMS->wyslijPotwierdzenieUsuniecia('+48500600700');

if($wynik){
    echo 'Wiadomość została wysłana';
} else {
    echo 'Wystąpił błąd';
}

?>
```

wyslijSMS

Wysyła wiadomość SMS na dany numer o dowolnej treści.

bool **wyslijSMS**(string **\$numer**, string **\$tresc**);

Przyjmowane wartości:

numer

Ciąg znaków będący numerem telefonu, na który ma zostać wysłana wiadomość.

tresc

Ciąg znaków będący treścią wiadomości SMS.

Zwracane wartości:

Zwraca wartość **true** lub **false** w zależności od powodzenia operacji.

Przykład:

```
<?php
require_once("SerwerSMS_Kontakty.class.php");

$SerwerSMS = new SerwerSMS_Kontakty('login','haslo');
$wynik = $SerwerSMS->wyslijSMS('+48500600700','Tresc wiadomosci SMS');

if($wynik){
    echo 'Wiadomość została wysłana';
} else {
    echo 'Wystąpił błąd';
}

?>
```

Przykładowe skrypty

1. Formularz bez weryfikacji SMS

Formularz posiada jedno pole do wpisania numeru oraz przycisk *Dodaj*. Zapis odbywa się bez potwierdzenia kodem SMS, a numer trafia do jednej, zadeklarowanej grupy kontaktów. Po prawidłowym zapisaniu numeru w bazie, wysyłana jest wiadomość SMS z potwierdzeniem poprawnie wykonanej operacji. Do danych jakie należy wprowadzić w skrypcie należą: login i hasło do Serwer SMS, numer id grupy kontaktów. Identyfikator grupy widoczny jest w Panelu Klienta, po wybraniu edycji danej grupy.

```
<?php
session_start();

function czyscDane($string){
    return addslashes(htmlspecialchars(trim($string)));
}

require_once("SerwerSMS_Kontakty.class.php");

$SerwerSMS = new SerwerSMS_Kontakty("login","haslo");
$SerwerSMS->setNadawca('INFORMACJA'); // Nagłówek nadawcy wiadomości SMS
$SerwerSMS->setPotwierdzenieSMS("To jest tresc smsa potwierdzajacego dopisanie do kontaktow");

$wynik = null;
$numer = null;
$id_grupy = '00000001';

if(isset($_POST['numer'])){
```

```
$numer = czyscDane($_POST['numer']);
$dodaj = $SerwerSMS->dodajKontakt($id_grupy,$numer);
if($dodaj['stan'] == 'OK'){
    $SerwerSMS->wyslijPotwierdzenieDopisania($numer);
    $wynik = 1;
} else {
    $wynik = 2;
}
} else {
    $wynik = null;
}
?>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>SerwerSMS Kontakty - Przykład 1</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
<p>
<?php
switch ($wynik) {
case 1:
echo '<span style="color: green">Twój numer został poprawnie dodany do bazy. Za chwilę otrzymasz SMS z
potwierdzeniem</span>';
break;
case 1:
echo '<span style="color: red">Błąd zapisu numeru</span>';
break;
}
?>
</p>

<form method="post">
<label for="numer">Numer telefonu komórkowego: </label>
<input type="text" name="numer" id="numer" />
<input type="submit" value="Dodaj" />
</form>
</body>
</html>
```

2. Weryfikacja SMS oraz wykorzystanie NDI

Formularz pozwala na dodanie oraz usunięcie numeru z grupy kontaktów. Po podaniu swojego numeru telefonu, wyboru grupy oraz akcji *Zapisz/Wypisz* użytkownik zostaje przekierowany do drugiego kroku, w którym to na wpisany przez niego numer telefonu wysłany zostaje kod weryfikacyjny. Aby potwierdzić swój numer, właściciel telefonu musi przepisać otrzymany kod do pola formularza w kroku drugim. Dane takie jak numer telefonu, grupa, typ operacji i kod potwierdzający zapisywane są w sesji przeglądarki. Po wpisaniu prawidłowego kodu, skrypt pobiera te informacje i na ich podstawie tworzy hasło dostępne. W poniższym przykładzie użytkownik może zostać dopisany do jednej z trzech grup, gdzie każda z nich ma osobne hasło dostępne. Jeśli zaś właściciel telefonu wybrał opcję usunięcia numeru, przypisane zostanie mu hasło USUN. Skrypt przekazuje hasło dostępne za pomocą metody *przetworzPrzezNDI()* i dalej wykonywana będzie akcja ustawiona w konfiguracji numeru dostępnego w Panelu Klienta SerwerSMS.pl. Parametry jakie należy wprowadzić do skryptu to login i hasło konta w SerwerSMS.pl oraz numer dostępowy indywidualny.

```
<?php

session_start();

function czyscDane($string){
    return addslashes(htmlspecialchars(trim($string)));
}

require_once("SerwerSMS_Kontakty.class.php");

$SerwerSMS = new SerwerSMS_Kontakty("login","haslo");
// Wiadomość bez nazwy nadawcy (ECO)
// $SerwerSMS->setNadawca('INFORMACJA'); // Nagłówek nadawcy wiadomości SMS
$SerwerSMS->setPotwierdzenieSMS("To jest tresc smsa potwierdzającego dopisanie do kontaktow");
$SerwerSMS->setTrescKodSMS("To jest tresc smsa z kodem potwierdzającym. Twój kod to:");
$SerwerSMS->setNDI('+48000000000'); // Ustawienie Numeru Dostępnego Indywidualnego

$weryfikacja = null;
$wynik = null;

if(isset($_POST['numer'])){
    $numer = czyscDane($_POST['numer']);
    $_SESSION['numer'] = $numer;
    $_SESSION['grupa'] = (int) $_POST['grupa'];
    $_SESSION['wybor'] = (int) $_POST['wybor'];
    $sms = $SerwerSMS->weryfikacjaSMS($numer);
    if($sms){
        $weryfikacja = 1;
    } else {
        $weryfikacja = 0;
    }
} else if(isset($_POST['kodsms'])){
    $kodsms = czyscDane($_POST['kodsms']);
    if($SerwerSMS->sprawdzKod($kodsms)){
        if($wybor == 2){
            $haslo = 'USUN';
        } else {
            switch($_SESSION['grupa']){
                case 1: $haslo = 'WARSZAWA'; break;
                case 2: $haslo = 'KATOWICE'; break;
                case 3: $haslo = 'GDANSK'; break;
                default: $haslo = 'BRAK'; break;
            }
        }
    }
}
```

```

$wyslij = $SerwerSMS->przetworzPrzezND($_SESSION['numer'],$haslo);
if($wyslij){
    $wynik = 1;
}

unset($_SESSION);
session_destroy();
} else {
    $wynik = 2;
    $weryfikacja = 1;
}
} else {
    $wynik = 4;
}
?>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>SerwerSMS Kontakty - Przykład 2</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
<p>
<?php
switch ($wynik) {
    case 1:
        echo '<span style="color: green">Operacja zakończona powodzeniem. Za chwilę otrzymasz SMS z
potwierdzeniem</span>';
        break;
    case 2:
        echo '<span style="color: red">Wpisany kod jest niepoprawny. Spróbuj ponownie</span>';
        break;
}
?>
</p>

<?php
if (!isset($weryfikacja) or $weryfikacja != '1') {
?>

<form method="post">
<label for="numer">Numer telefonu komórkowego: </label>
<input type="text" name="numer" id="numer" />
<br />
<input type="radio" name="wybor" id="zapisz" value="1" checked="checked" />
<label for="zapisz">Zapisz</label>
<input type="radio" name="wybor" id="wypisz" value="2" />
<label for="wypisz">Wypisz</label>

<ul>
<li>
<input type="radio" name="grupa" value="1" id="grupa1" />
<label for="grupa1">Warszawa</label>
</li>
<li>
<input type="radio" name="grupa" value="2" id="grupa2" />
<label for="grupa2">Katowice</label>
</li>
<li>
<input type="radio" name="grupa" value="3" id="grupa3" />
<label for="grupa3">Gdańsk</label>
</li>
</ul>
<input type="submit" value="Wyślij zgłoszenie" />
</form>

<?php
} else { // ELSE
?>
<form method="post">
<label for="kodsms">Potwierdzający kod SMS*:</label>
<input type="text" name="kodsms" id="kodsms" />

<br />
<input type="submit" value="Potwierdź" />

</form>
<?php
} // END IF
?>
</body>
</html>

```

3. Weryfikacja SMS oraz zapis do kilku grup jednocześnie

Formularz w tym przykładzie również posiada weryfikację kodem SMS, a co za tym idzie wymagana jest rejestracja sesji przeglądarki. Poza numerem telefonu wprowadzić można dodatkowe dane takie jak imię, nazwisko oraz adres e-mail. Ponadto możliwe jest zapisanie się do więcej niż jednej grupy kontaktów. W przeciwieństwie do przykładu nr 2, brakuje haseł dostępowych, a do dodania kontaktu wykorzystana jest metoda *dodajKontakt()* oraz do usuwania *usunKontakt()*. Również wiadomości SMS z komunikatami potwierdzającymi wysyłane są z poziomu skryptu. Po wpisaniu niezbędnych danych, w momencie kliknięcia *Wyślij zgłoszenie*, na numer telefonu wpisany w formularzu wysyłana jest wiadomość z potwierdzającym kodem SMS. Użytkownik musi przepisać kod z SMS-a do specjalnego pola formularza, które ukaże się w drugim kroku. Po tym zabiegu skrypt pobiera z sesji przeglądarki wprowadzone wcześniej dane i zależnie o wybranej akcji usuwa lub dodaje numer. Po poprawnie wykonanej operacji wysyłane są komunikaty z potwierdzeniem dopisania lub usunięcia numeru. Aby skrypt działał poprawnie, przy utworzeniu obiektu klasy *SerwerSMS_Kontakty*, należy podać swój login i hasło do platformy *SerwerSMS.pl* oraz w polach formularza typu checkbox wprowadzić identyfikatory grup kontaktów. Numery te widoczne są w Panelu Klienta, po wybraniu edycji danej grupy.

```
<?php
session_start();

function czyscDane($string){
    return addslashes(htmlspecialchars(trim($string)));
}

require_once("SerwerSMS_Kontakty.class.php");

$SerwerSMS = new SerwerSMS_Kontakty("login","haslo");
$SerwerSMS->setNadawca('INFORMACJA'); // Nagłówek nadawcy wiadomości SMS
$SerwerSMS->setPotwierdzenieSMS("To jest tresc smsa potwierdzajacego dopisanie do kontaktow");
$SerwerSMS->setUsuniecieSMS("To jest tresc smsa potwierdzajacego usunieie numeru z kontaktow");
$SerwerSMS->setTrescKodSMS("To jest tresc smsa z kodem potwierdzajacym. Twój kod to:");

$weryfikacja = null;
$wynik = null;

if (isset($_POST['numer'])) {
    $numer = czyscDane($_POST['numer']);
    $_SESSION['numer'] = $numer;
    $_SESSION['email'] = czyscDane($_POST['email']);
    $_SESSION['imie'] = czyscDane($_POST['imie']);
    $_SESSION['nazwisko'] = czyscDane($_POST['nazwisko']);
    $_SESSION['grupa'] = $_POST['grupa'];
    $_SESSION['wybor'] = (int) $_POST['wybor'];
    $sms = $SerwerSMS->weryfikacjaSMS($numer);
    if ($sms) {
        $weryfikacja = 1;
    } else {
        $weryfikacja = 0;
    }
} else if (isset($_POST['kodsms'])) {
    $kodsms = czyscDane($_POST['kodsms']);
    if ($SerwerSMS->sprawdzKod($kodsms)) {
        if ($_SESSION['wybor'] == 2) {
            $blad = false;
            foreach ($_SESSION['grupa'] as $id_grupy) {
                $identyfikatory[] = $SerwerSMS->pobierzIdentyfikatory($_SESSION['numer'], $id_grupy);
            }

            foreach ($identyfikatory as $kontakt) {
                foreach($kontakt as $id_kontaktu){
                    $usun = $SerwerSMS->usunKontakt($id_kontaktu);
                    if (!$usun) {
                        $blad = true;
                    }
                }
            }
            //$usun = $SerwerSMS->usunNumer($_SESSION['numer']);
            if (!$blad) {
                $SerwerSMS->wyslijPotwierdzenieUsuniecia($_SESSION['numer']);
                $wynik = 3;
            }
        } else {
            $blad = false;
            foreach ($_SESSION['grupa'] as $id_grupy) {
                $dodaj = $SerwerSMS->dodajKontakt($id_grupy, $_SESSION['numer'], $_SESSION['email'], $_SESSION['imie'],
                $_SESSION['nazwisko']);
                if ($dodaj['stan'] != 'OK') {
                    $blad = true;
                }
            }
            if (!$blad) {
                $SerwerSMS->wyslijPotwierdzenieDopisania($_SESSION['numer']);
                $wynik = 1;
            }
        }
        unset($_SESSION);
        session_destroy();
    } else {
        $wynik = 2;
        $weryfikacja = 1;
    }
} else {
    $wynik = 4;
}
?>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>SerwerSMS Kontakty - Przykład 3</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
<p>
<?php
switch ($wynik) {
    case 1:
        echo '<span style="color: green">Twój numer został poprawnie dodany do bazy. Za chwilę otrzymasz SMS z
potwierdzeniem</span>';
        break;
    case 2:
        echo '<span style="color: red">Wpisany kod jest niepoprawny. Spróbuj ponownie</span>';
        break;
    case 3:
        echo '<span style="color: green">Numer został poprawnie usunięty. Za chwilę otrzymasz SMS z
potwierdzeniem</font>';
        break;
}
?>

</p>

<?php
if (!isset($weryfikacja) or $weryfikacja != '1') {
    ?>

<form method="post">
<label for="imie">Imię:</label>
```

```

<input type="text" name="imie" id="imie" />
<br />
<label for="nazwisko">Nazwisko:</label>
<input type="text" name="nazwisko" id="nazwisko" />
<br />
<label for="numer">Numer telefonu komórkowego:</label>
<input type="text" name="numer" id="numer" />
<br />
<label for="email">Adres e-mail:</label>
<input type="text" name="email" id="email" />
<br />
<input type="radio" name="wybor" id="zapisz" value="1" checked="checked" />
<label for="zapisz">Zapisz</label>
<input type="radio" name="wybor" id="wypisz" value="2" />
<label for="wypisz">Wypisz</label>

<ul>
  <li>
    <input type="checkbox" name="grupa[]" value="00000001" id="grupa1" />
    <label for="grupa1">Grupa 1</label>
  </li>
  <li>
    <input type="checkbox" name="grupa[]" value="00000002" id="grupa2" />
    <label for="grupa2">Grupa 2</label>
  </li>
  <li>
    <input type="checkbox" name="grupa[]" value="00000003" id="grupa3" />
    <label for="grupa3">Grupa 3</label>
  </li>
</ul>
<input type="submit" value="Wyślij zgłoszenie" />
</form>

<?php
} else { // ELSE
  ?>
  <form method="post">
    <label for="kodsms">Potwierdzający kod SMS:</label>
    <input type="text" name="kodsms" id="kodsms" />

    <br />
    <input type="submit" value="Potwierdź" />

  </form>

<?php
} // END IF
?>
</body>
</html>

```

Historia zmian

Wersja 1.0 - 2013-08-01

- Pierwsza wersja dokumentacji zawierająca podstawowe informacje na temat metod klasy *ServerSMS_Kontakty* oraz przykłady zastosowania.