

Creazione clienti tramite richiesta POST HTTP

URL: <http://sms.host.tld/backoffice/client-add.php>

Parametri autenticazione:

- smsusername – Login amministratore
- smspassword – Password amministratore

Parametri obbligatori:

- name – Nome del cliente
- username – Username per l'accesso al pannello
- password – Password per l'accesso al pannello

Parametri opzionali:

- email – E-mail del cliente
- tpl_id – Identificativo univoco del profilo assegnato
- contact – Campo libero "Contatto"
- ref_id – Campo libero "Riferimento"
- reseller – Può essere impostato a 0 (cliente finale) o 1 (rivenditore, default)
- vhost – Dominio associato al rivenditore, sul quale dovranno loggarsi gli eventuali clienti

Il server risponderà con "OK <id>", ove <id> è l'identificativo numerico assegnato al nuovo cliente, in caso di successo. In caso di errore la risposta del server sarà "KO <testo_errore>".

N.B. L'accesso è limitato ai soli indirizzi IP o subnet abilitati al servizio back-office.

Assegnazione crediti tramite richiesta POST HTTP

URL: <http://sms.host.tld/backoffice/credit-add.php>

Parametri autenticazione:

- smsusername – Login amministratore
- smspassword – Password amministratore

Parametri obbligatori:

- u_id – Identificativo univoco cliente
- bill_id – Identificativo univoco tariffa

Parametri opzionali:

- credit – Credito, nel formato 12,3456 o 12.3456

Il server risponderà con "OK <id>", ove <id> è l'identificativo numerico assegnato al nuovo credito, in caso di successo. In caso di errore la risposta del server sarà "KO <testo_errore>".

N.B. L'accesso è limitato ai soli indirizzi IP o subnet abilitati al servizio back-office.

Controllo crediti tramite richiesta POST HTTP

URL: <http://sms.host.tld/backoffice/credit-get.php>

Parametri autenticazione:

- smsusername – Login amministratore
- smspassword – Password amministratore

Parametri obbligatori:

- u_id – Identificativo univoco cliente

Il server risponderà con "OK <credito>", ove <credito> è il corrispettivo in euro del credito disponibile per il cliente, con 4 cifre decimali e il carattere "." (punto) come separatore dei decimali (ad es. 123.4567). In caso di errore la risposta del server sarà "KO <testo_errore>".

N.B. L'accesso è limitato ai soli indirizzi IP o subnet abilitati al servizio back-office.

Controllo operazioni back-office tramite richiesta POST HTTP

URL: <http://sms.host.tld/backoffice/userlog-get.php>

Parametri autenticazione:

- smsusername – Login amministratore
- smspassword – Password amministratore

Parametri obbligatori:

- type – Tipologia di operazione ("credits" è l'unica operazione attualmente supportata)

Parametri opzionali:

- u_id – Identificativo univoco utente
- from – Data inizio report
- to – Data fine report

Il server risponderà con un report in cui la prima riga contiene l'intestazione dei campi e le successive i dati. I campi sono a lunghezza variabile separati da tabulazione e le righe sono terminate dai caratteri <CR><LF>. In caso di errore la risposta del server sarà "KO <testo_errore>".

N.B. L'accesso è limitato ai soli indirizzi IP o subnet abilitati al servizio back-office.

Creazione e assegnazione ricezione tramite richiesta POST HTTP

URL: <http://sms.host.tld/backoffice/recv-add.php>

Parametri autenticazione:

- smsusername – Login amministratore
- smspassword – Password amministratore

Parametri obbligatori:

- u_id – Identificativo univoco cliente
- dest – Numero di telefono ricezione

Parametri opzionali:

- num – Numero di codici da creare (max 20, default 1)

Il server risponderà con "OK <codici>", ove <codici> è l'elenco dei codici di condivisione creati separati da una virgola. In caso di errore la risposta del server sarà "KO <testo_errore>".

N.B. L'accesso è limitato ai soli indirizzi IP o subnet abilitati al servizio back-office.

Attivazione / Disattivazione clienti tramite richiesta POST HTTP

URL: <http://sms.host.tld/backoffice/client-status.php>

Parametri autenticazione:

- smsusername – Login amministratore
- smspassword – Password amministratore

Parametri obbligatori:

- u_id – Identificativo univoco utente
- active – 0 = disattivato / 1 = attivo

Il server risponderà con "OK <stato>", ove <stato> è l'operazione eseguita (Attivato o Disattivato). In caso di errore la risposta del server sarà "KO <testo_errore>".

N.B. L'accesso è limitato ai soli indirizzi IP o subnet abilitati al servizio back-office.

Invio singolo SMS tramite richiesta POST/GET HTTP

URL: <http://sms.host.tld/sms/send.php>

Parametri autenticazione:

- user – Login cliente
- pass – Password cliente

Parametri obbligatori:

- rcpt – Numero destinatario nel formato internazionale +XXYYZZZZZZZZ
- data – Testo del messaggio (max 160 caratteri)
- sender – Mittente del messaggio (max 11 caratteri alfanumerici o numero +XXYYZZZZZZZZ)

Parametri opzionali:

- qty – Qualità del messaggio: (l, i, a, h, n – bassa, media, automatica, alta, notifica)

Il server risponderà con "OK <costo>", ove <costo> è il credito scalato per l'invio del messaggio. In caso di errore la risposta del server sarà "KO <testo_errore>".

N.B. L'accesso è limitato ai soli indirizzi IP o subnet abilitati al servizio di invio sms tramite richiesta HTTP.

Invio multiplo SMS tramite richiesta POST/GET HTTP

URL: <http://sms.host.tld/sms/batch.php>

Parametri autenticazione:

- user – Login cliente
- pass – Password cliente

Parametri obbligatori:

- rcpt – Elenco numeri nel formato internazionale +XXYYZZZZZZZ, separati da virgola.
- data – Testo del messaggio (max 160 caratteri)
- sender – Mittente del messaggio (max 11 caratteri alfanumerici o numero +XXYYZZZZZZZZ)

Parametri opzionali:

- qty – Qualità del messaggio: (l, i, a, h, n – bassa, media, automatica, alta, notifica)

Il server risponderà con "OK". In caso di errore la risposta del server sarà "KO <testo_errore>".

N.B. L'accesso è limitato ai soli indirizzi IP o subnet abilitati al servizio di invio sms tramite richiesta HTTP.

Controllo credito / SMS rimanenti tramite richiesta POST/GET HTTP

URL: <http://sms.host.tld/sms/credit.php>

Parametri autenticazione:

- user – Login cliente
- pass – Password cliente

Parametri opzionali:

- type – Tipo di controllo ("credit" credito residuo – default –, "lqs" messaggi bassa qualità, "hqs" messaggi alta qualità)

Il server risponderà con "OK <valore>", ove <valore> è il numero di sms o il credito residuo in euro, a seconda del parametro type specificato. In caso di errore la risposta del server sarà "KO <testo_errore>".

N.B. L'accesso è limitato ai soli indirizzi IP o subnet abilitati al servizio di invio sms tramite richiesta HTTP.

Controllo messaggi ricevuti tramite Web Service SOAP

I file da utilizzare sono due:

lib-nusoap.inc.php (scarica da <http://www.natan.it/docs/lib-nusoap.inc.zip>)
ws.php (scarica da <http://www.natan.it/docs/ws.zip>)

Ambedue i file vanno messi sotto la stessa path e il primo viene richiamato all'interno del secondo. Il primo file (lib-nusoap.inc.php) non é altro che una libreria php che permette di effettuare la chiamata soap. Il secondo (ws.php) é lo script contenente i parametri per la chiamata.

Riportiamo di seguito il codice del secondo file (ws.php) :

```
<!-----INIZIO----->
<?php
$host = 'sms.tol.it'; // Host
$user = ""; // Username
$pass = ""; // Password
$rcpt = '+393202043297'; // Numero
$code = 'xxxx' // Cod. condivisione
$mess = 10; // N. messaggi

require('./lib-nusoap.inc.php');

$client = new soapclient('http://'.$host.'/wsdl/?wsdl', true);

$error = $client->getError();
if ($error)
{
echo '<h2>Constructor error</h2><pre>' . $error . '</pre>';
exit;
}

$result = $client->call('receiveSms', array($user, $pass, $rcpt, $code, $mess));

if ($client->fault)
{
echo '<h2>Fault</h2><pre>';
print_r($result);
echo '</pre>';
}
else
{
$error = $client->getError();
if ($error)
{
echo '<h2>Error</h2><pre>' . $error . '</pre>';
exit;
}
}

echo '<h2>Result</h2><pre>';
print_r($result);
echo '</pre>';

?>

<!-----FINE----->
```

Controllo MNC tramite richiesta POST HTTP

I file da utilizzare sono tre:

lib-scriptsms.inc.php (scarica da <http://www.natan.it/docs/lib-scriptsms.inc.zip>)
SendMNC.php (scarica da <http://www.natan.it/docs/SendMNC.zip>)
Status.php (scarica da <http://www.natan.it/docs/Status.zip>)

Tutti e tre i file vanno messi sotto la stessa path e il primo viene richiamato all'interno del secondo e del terzo. Il primo file (lib-scriptsms.inc.php) non è altro che una libreria php che permette di effettuare la verifica MNC. Il secondo (SendMNC.inc.php) è lo script contenente i parametri per la chiamata MNC.

Il terzo script verifica lo status

Dato che la chiamata MNC è asincrona è necessario salvare il risultato della verifica MNC su di un file testuale (mnc.txt) che poi verrà letto dal terzo script Status.php il quale darà in output il risultato.

Il file mnc.txt viene sovrascritto ogni volta che viene eseguita una verifica MNC in modo da memorizzare solo l'ultima ricerca (questo per semplicità) ma può anche essere scritto in append ovviamente si dovrà modificare a quel punto anche Status.php per la lettura.

Riportiamo di seguito il codice del secondo file (SendMNC.php) :

```
<!-----INIZIO----->

<?php
require('./lib-scriptsms.inc.php');
$sms = new mobytSms('username', 'password'); //credenziali del servizio sms
$sms->setDomain('http://sms.tol.it');
$result = $sms->sendMNC('+3933811111111,+393382222222',1);//numeri da verificare
if (substr($result, 0, 2) == 'OK') {
echo 'Richiesta inviata correttamente'.$result;
$file = "./mnc.txt";
$handle = fopen($file,"w+");
fwrite($handle, $result);
fclose($handle);
}
else
{ echo 'Richiesta NON inviata correttamente'.$result;}
?>

<!-----FINE----->
```

Nota: all'interno della funzione sendMNC(...) vanno messi tutti i numeri da verificare separati da virgola

Riportiamo di seguito il codice del terzo file (Status.php) :

```
<!-----INIZIO----->

<?php
// Verifica stato della spedizione
require('./lib-scriptsms.inc.php');
$sms = new mobytSms('username', 'password');//credenziali del servizio sms
$sms->setDomain('http://sms.tol.it');
$file = "./mnc.txt";
$handle = fopen($file,"r+");
$content = fread($handle, filesize($file));
$stato = trim(strstr($content, ' '));
$result = $sms->sendStatus($stato,'mnc');
if (substr($result, 0, 2) == 'KO')
echo 'Richiesta NON inviata correttamente'.$result;
else
echo $result;
?>

<!-----FINE----->
```

Controllo stato spedizioni tramite richiesta POST/GET HTTP

Ad oggi è possibile controllare lo stato delle spedizioni tramite script; di seguito riportiamo i parametri necessari:

URL da invocare:
`http://sms.host.tld/sms/batch-status.php`

Parametri autenticazione:
user Login cliente
pass Password cliente

Parametri obbligatori:
id Identificativo della spedizione (request_id specificato al momento dell'invio della richiesta)
type Tipo di report desiderato (queue = stato accodamento messaggi, notify = stato delle notifiche dei messaggi inviati, mnc = stato delle richieste MNC)
schema Schema del report (1 Ã" 'unico schema attualmente supportato).

Il server rispondera' con i dati del report richiesto in formato CSV con i campi separati da virgola, dove la prima riga conterrà i nomi delle colonne.
In caso di errore la risposta del server sarà KO <testo_errore>.

Codici errore:

- KO Accesso non consentito (Utente non abilitato).
- KO Accesso non consentito (IP non abilitato).
- KO Parametro <nome parametro> non impostato.
- KO Parametro <nome parametro> non corretto.
- KO Errore interno.
- KO Accesso non consentito.
- KO Parametri non corretti.

Di seguito riportiamo il codice di esempio:

```
#####ESEMPIO#####  
<?php  
var_dump(httpPost(array(  
  'user' => 'xxx',  
  'pass' => 'xxx',  
  'id' => 'xxxxxxxx',  
  'type' => 'notify',  
  'schema' => '1'  
),  
  'sms.suodominio'  
));  
function httpPost($fields, $host = 'sms.suodominio', $url = '/batch-status.php')  
{  
  $qs = array();  
  foreach ($fields as $k => $v)  
    $qs[] = $k.'='.urlencode($v);  
  $qs = join('&', $qs);  
  $errno = $errstr = '';  
  if ($fp = @fsockopen('sms.suodominio', 80, $errno, $errstr, 30))  
  {  
    fputs($fp, "POST ".$url." HTTP/1.0\r\n");  
    fputs($fp, "Host: ".$host."\r\n");  
    fputs($fp, "User-Agent: PHP/".phpversion()."\r\n");  
    fputs($fp,  
    "Content-Type:application/x-www-form-urlencoded\r\n");  
    fputs($fp, "Content-Length: ".strlen($qs)."\r\n");  
    fputs($fp, "Connection: close\r\n");  
    fputs($fp, "\r\n".$qs);  
    $content = '';  
    while (!feof($fp))  
      $content .= fgets($fp, 1024);  
    fclose($fp);  
    return preg_replace("/^\.*?\r\n\r\n/s", "", $content);  
  }  
  return false;  
}
```

```
}  
?>
```

```
#####
```

NOTA: Il batch va copiato interamente senza alterare nulla tranne i tre parametri necessari:

```
'user' => 'xxx',  
'pass' => 'xxx',  
'id' => 'xxxxxxx',
```