

Specifiche funzionali protocollo HTTP Gateway VolaSMS Plus Ver. 3.0.0

Aggiornamenti

Data	Versione	Note
01-10-2008	3.0.0	<ul style="list-style-type: none">- Nuovi comandi per la gestione dei mittenti con obbligo di autenticazione per i mittenti numero di cellulare.- Nuovo comando di invio SMS con richiesta di conferma

Indice

Indice.....	2
1. Il gateway SMS.....	5
2. Lo script volasms_gw_plus2.php (HTTP [80] e HTTPS [443]).....	5
3. Come inviare i comandi al gateway.....	6
3.1 Richiesta credito residuo (CMD = 1).....	8
3.1.1 CMD (obbligatorio).....	8
3.1.2 UID (obbligatorio).....	8
3.1.3 PWD (obbligatorio).....	8
3.1.4 SERIAL (obbligatorio).....	8
3.1.5 Risposta positiva del gateway.....	8
3.2 Ricezione messaggi (CMD=4).....	9
3.2.1 CMD (obbligatorio).....	9
3.2.2 UID (obbligatorio).....	9
3.2.3 PWD (obbligatorio).....	9
3.2.4 SERIAL (obbligatorio).....	9
3.2.5 KEY (facoltativo).....	9
3.2.6 Risposta positiva del gateway.....	9
3.3 Ack ricezione messaggi (CMD = 5).....	10
3.3.1 CMD (obbligatorio).....	10
3.3.2 UID (obbligatorio).....	10
3.3.3 PWD (obbligatorio).....	10
3.3.4 SERIAL (obbligatorio).....	10
3.3.5 Risposta positiva del gateway.....	10
3.4 Comando CheckMessagesMO (CMD=6).....	11
3.4.1 CMD (obbligatorio).....	11
3.4.2 UID (obbligatorio).....	11
3.4.3 PWD (obbligatorio).....	11
3.4.4 SERIAL (obbligatorio).....	11
3.4.5 KEY (facoltativo).....	11
3.4.6 Risposta positiva del gateway.....	11
3.5 Invio messaggio SendOptimized (CMD = 14).....	12
3.5.1 CMD (obbligatorio).....	12
3.5.2 UID (obbligatorio).....	12
3.5.3 PWD (obbligatorio).....	12
3.5.4 SERIAL (obbligatorio).....	12
3.5.5 SENDDATA (obbligatorio).....	12
3.5.6 Parametro VCODE (facoltativo).....	13
3.5.7 Risposta positiva del gateway.....	13
3.6 Invio messaggi SendOptimized con richiesta di conferma (CMD = 44).....	15
3.7 Conferma di invio messaggi (CMD 45).....	15
3.7.1 CMD (obbligatorio).....	15
3.7.2 UID (obbligatorio).....	15
3.7.3 PWD (obbligatorio).....	15
3.7.4 SERIAL (obbligatorio).....	15
3.7.5 ORDERID (obbligatorio).....	15
3.7.6 Risposta positiva del gateway.....	15
3.8 Richiesta notifiche (CMD = 10).....	16
3.8.1 CMD (obbligatorio).....	16
3.8.2 UID (obbligatorio).....	16
3.8.3 PWD (obbligatorio).....	16
3.8.4 SERIAL (obbligatorio).....	16
3.8.5 QUERYDATA (obbligatorio).....	16

3.8.6 Risposta positiva del gateway	16
3.9 Get autoresponder (CMD = 16)	19
3.9.1 CMD (obbligatorio)	19
3.9.2 UID (obbligatorio)	19
3.9.3 PWD (obbligatorio)	19
3.9.4 SERIAL (obbligatorio)	19
3.9.5 KEY (obbligatorio)	19
3.9.6 Risposta positiva del gateway	19
3.10 Set autoresponder (CMD = 17)	20
3.10.1 CMD (obbligatorio)	20
3.10.2 UID (obbligatorio)	20
3.10.3 PWD (obbligatorio)	20
3.10.4 SERIAL (obbligatorio)	20
3.10.5 AUTORESPONDER (facoltativo)	20
3.10.6 KEY (obbligatorio)	20
3.10.7 Risposta positiva del gateway	20
3.11 Inserimento di un nuovo mittente	21
3.11.1 CMD (obbligatorio)	21
3.11.2 UID (obbligatorio)	21
3.11.3 PWD (obbligatorio)	21
3.11.4 SERIAL (obbligatorio)	21
3.11.5 SENDER (obbligatorio)	21
3.11.6 Risposta positiva del gateway	21
3.11.7 Codici di errore	21
3.12 Richiesta pincode per autenticare un numero di cellulare	22
3.12.1 CMD (obbligatorio)	22
3.12.2 UID (obbligatorio)	22
3.12.3 PWD (obbligatorio)	22
3.12.4 SERIAL (obbligatorio)	22
3.12.5 SENDER (obbligatorio)	22
3.12.6 Risposta positiva del gateway	22
3.13 Validazione del pincode per l'autenticazione di un mittente (numero di cellulare)	23
3.13.1 CMD (obbligatorio)	23
3.13.2 UID (obbligatorio)	23
3.13.3 PWD (obbligatorio)	23
3.13.4 SERIAL (obbligatorio)	23
3.13.5 SENDER (obbligatorio)	23
3.13.6 PINCODE (obbligatorio)	23
3.13.7 Risposta positiva del gateway	23
3.14 Elenco mittenti impostati	24
3.14.1 CMD (obbligatorio)	24
3.14.2 UID (obbligatorio)	24
3.14.3 PWD (obbligatorio)	24
3.14.4 SERIAL (obbligatorio)	24
3.14.5 Risposta positiva del gateway	24
3.15 Cancellazione di un mittente	25
3.15.1 CMD (obbligatorio)	25
3.15.2 UID (obbligatorio)	25
3.15.3 PWD (obbligatorio)	25
3.15.4 SERIAL (obbligatorio)	25
3.15.4 SENDER (obbligatorio)	25
3.15.5 Risposta positiva del gateway	25
4 Esempi script PHP	26
4.1 Richiesta credito residuo SMS	26
4.2 Ricezione messaggi	27

4.3 Invio SMS.....	29
4.4 Richiesta notifiche	30
4.5 Inserimento di un nuovo mittente	32
4.6 Inserimento di un numero di cellulare come mittente.....	33
4.7 Invio SMS con metodo di conferma.....	34
5. Principali problemi.....	36
5.1 Problemi generici	36
5.2 Inserimento di un nuovo mittente (CMD 30)	36
5.4 Richiesta di invio PINCODE per autenticazione numero di cellulare come mittente (CMD 34)	37

1. Il gateway SMS

VolaSMS è il servizio di invio SMS di VOLA SpA.

L'invio dei messaggi avviene chiamando uno script lato server e passando i parametri relativi ai destinatari e al testo del messaggio. La connessione sfrutta il protocollo HTTP.

2. Lo script volasms_gw_plus2.php (HTTP [80] e HTTPS [443])

Lo script da chiamare si trova all'indirizzo

http://sms.vola.it/cgi/volasms_gw_plus2.php

ovvero sulla porta 80 del server sms.vola.it

E' possibile anche utilizzare una connessione sicura SSL in modo che il flusso dei dati tra client e server avvenga in maniera protetta. In questo caso la url da richiamare e'

https://sms.vola.it/cgi/volasms_gw_plus2.php

ovvero sulla porta 443 del server sms.vola.it

Per inviare le richieste, si consiglia di utilizzare sempre il metodo POST del protocollo http rispetto al metodo GET alternativo, soprattutto quando il numero di dati da inviare e' grande.

Lo script espone una serie di funzioni o comandi che vengono di seguito elencati. Ogni comando viene identificato dal parametro obbligatorio **CMD**:

Parametro **CMD** : valori ammessi:

- 1 richiesta credito residuo
- 4 ricezione messaggi MO (Mobile Originated)
- 5 ack ricezione messaggi MO (Mobile Originated)
- 6 check ricezione messaggi MO (Mobile Originated)
- 10 richiesta notifiche su messaggi (MT)
- 14 invio messaggi MT (Mobile Terminated)
- 16 get autoresponder
- 17 set autoresponder
- 30 inserimento di un nuovo mittente
- 31 cancellazione di un mittente
- 32 elenco mittenti impostati
- 33 autenticazione di un mittente (solo se è un numero di cellulare)
- 34 richiesta invio pincode su numero di cellulare da autenticare come mittente
- 44 invio SMS con conferma
- 45 conferma di invio degli SMS inviati al Gateway tramite il comando 44

Ad ogni richiesta il server risponde con una particolare sintassi che viene di seguito descritta:

[header di risposta server]

<HTML>\r\n

<BODY>\r\n

[cod_res][space][data_res]\r\n

```
</BODY>\r\n
</HTML>
```

NB: i caratteri \r\n alla fine di ogni riga si riferiscono alla costante Visual Basic "vbCrLf" oppure ai caratteri ascii 10 /LF) e 13 (CR)

dove:

[Header di risposta server] e' costituito dagli header con cui il server invia le informazioni di risposta al client.

Contiene il risultato della richiesta... esempio 200, 404, 401 ecc. ecc. Per maggiori informazioni si rimanda alla specifica del protocollo HTTP.

Il parsing degli header puo' essere utile per catturare una eccezione di comunicazione con il server.

[cod_res] e' il codice che esprime il risultato dell'operazione ed e' composto da **2 caratteri** e puo' valere:

01 - Comando eseguito correttamente. Maggiori informazioni sul risultato dell'operazione possono essere ottenute facendo il parsing della stringa che segue [cod_res], ovvero [space][data_res], dove [data_res] dipende dal tipo di comando effettuato

88 - GATEWAY_SERIAL_ERROR (i parametro SERIAL non e' valorizzato correttamente)

89 - GATEWAY_CMD_ERROR (i parametri inviati nella richiesta non sono corretti)

93 - GATEWAY_AUTORESPONDER_ERROR (il parametro AUTORESPONDER non e' valorizzato correttamente)

94 - GATEWAY_VCODE_ERROR (il parametro VCODE non e' valorizzato correttamente)

96 - GATEWAY_ERROR (errore interno del gateway)

97 - GATEWAY_CREDIT_ERROR (credito sms insufficiente per l'invio corrente)

99 - GATEWAY_AUTH_ERROR (errore di autenticazione, username e password errate)

La lista completa dei codici di errore restituiti e' descritta nel paragrafo 5.

3. Come inviare i comandi al gateway

La comunicazione con il gateway di VolaSMS si basa sul protocollo http.

Ogni comando viene inviato mediante la chiamata di un'apposita URL alla quale passare una lista di parametri. L'ordine dei parametri non conta. La chiamata del client verso il server puo' avvenire utilizzando i metodi GET e POST, ma quest'ultimo e' da preferibile visto le limitazioni del metodo GET.

Esempio di chiamata GET:

```
http://sms.vola.it/cgi/volasms_gw_plus.php2?UID=xxxxx&PWD=xxxx&CMD=14&SENDDATA=1%09%2
B393471234567%09%2B393391234567%09Prova%090000-00-
00%0900:00&SERIAL=12345678900987654321
```

NB: nell'esempio sopra riportato si osserva come i caratteri tabulazione (rappresentato graficamente con \t in tutto il documento) e il carattere '+' vengono rappresentati mediante la codifica url encoding.

Esempio di chiamata POST:

```
POST cgi/volasms_gw_plus.php2 HTTP/1.1<cr_lf>
```

Host : sms.vola.it<cr_lf >
 Content-type: application/x-www-form-urlencoded<cr>
 Content-length: <length> <cr_lf >
 <cr_lf >
 UID=xxxx&PWD=xxxx&CMD=14&SENDDATA=1%09%2B393471234567%09%2B393391234567%09Prova
 %090000-00-00%0900:00&SERIAL=12345678900987654321<cr_lf >

N.B. Nel caso di chiamata http GET o POST e' bene effettuare l'url encoding dei valori dei parametri affinché il server possa interpretare correttamente i caratteri.

Se si utilizza il metodo POST e' importante anche specificare il parametro "HOST" della richiesta HTTP che deve essere "sms.vola.it"

Di seguito la tabella di riepilogo che mostra i caratteri speciali che valgono il doppio se inseriti nel testo del messaggio SMS.

[\u005B
\	\u005C
]	\u005D
^	\u005E
{	\u007B
	\u007C
}	\u007D
~	\u007E
€	\u20AC

Per cui il testo:
 [ciao]

vale 8 caratteri e non 6 caratteri per la presenza dei caratteri '[' e ']'

3.1 Richiesta credito residuo (CMD = 1)

Il comando GetCredit: restituisce il numero credito residuo di VolaSMS. Il credito e' espresso in numero di messaggi.

3.1.1 CMD (obbligatorio)

E' il parametro che indica il tipo di operazione ed ha il valore **1**

3.1.2 UID (obbligatorio)

E' il parametro che contraddistingue l'account del cliente. Il valore deve essere nel formato MD5

3.1.3 PWD (obbligatorio)

E' il parametro che contraddistingue la password dell'account del cliente. Il valore deve essere nel formato MD5

3.1.4 SERIAL (obbligatorio)

Numero seriale che va valorizzato obbligatoriamente con **TR45GDLBO730HDIUEQJ5**.

3.1.5 Risposta positiva del gateway

Se l'esecuzione del comando e' avvenuta correttamente il gateway risponde al client nel seguente modo:

```
01<space><credit_sms>\r\n
```

dove

<credit_sms> e' il credito residuo di sms. Il valore e' espresso con 2 decimali. Il carattere separatore dei decimali e' il punto (.).

3.2 Ricezione messaggi (CMD=4)

Restituisce la lista dei nuovi MO ricevuti dall'account dall'ultima esecuzione del comando andata a buon fine. La lista degli MO restituita dipende dal client applicativo che effettua la richiesta. La discriminazione viene fatta in base al valore del parametro KEY (chiave di ricezione) inviato nella richiesta.

Se non vi sono nuovi MO la lista risulterà vuota ed il comando restituirà la stringa di valore "NULL".

3.2.1 CMD (obbligatorio)

E' il parametro che indica il tipo di operazione ed ha il valore **4**

3.2.2 UID (obbligatorio)

E' il parametro che contraddistingue l'account del cliente. Il valore deve essere nel formato MD5

3.2.3 PWD (obbligatorio)

E' il parametro che contraddistingue la password dell'account del cliente. Il valore deve essere nel formato MD5

3.2.4 SERIAL (obbligatorio)

Numero seriale che va valorizzato obbligatoriamente con **TR45GDLBO730H DU IEQJ5**.

3.2.5 KEY (facoltativo)

E' il valore della chiave associata agli MO. Se viene specificato il valore verranno restituiti solo gli MO ultimi ricevuti dell'utente associati alla chiave di ricezione indicata. Se il valore non viene specificato vengono restituiti tutti gli MO ultimi ricevuti dell'account.

3.2.6 Risposta positiva del gateway

Se l'esecuzione del comando e' avvenuta correttamente il gateway risponde al client nel seguente modo:

```
01<space><num_mo><space><MO_list>\r\n
```

dove:

<num_mo> e' il numero di SMS MO nuovi ricevuti

<MO_list> e' la lista degli SMS MO ricevuti

- se <num_mo> vale zero allora <MO_list> viene valorizzato con la stringa "NULL"
- se <num_mo> e' un numero maggiore di 0 allora <MO_list> ha il seguente formato

```
<date>\t<time>\t<sender>\t<msg>\t<channel>\t<key>\x1A
```

il carattere \t e' la tabulazione e serve per separare i campi di ogni messaggio SMS MO.

<date> indica la data di ricezione

<time> indica l'ora di ricezione

<sender> indica il mittente del messaggio SMS MO

<msg> indica il testo del messaggio SMS MO

<channel> indica numero di cellulare a cui e' stato inviato il messaggio MO
(canale di ricezione)

<key> indica la chiave di ricezione associata al messaggio MO

il carattere esadecimale "1A" rappresentato nel documento con '\x1A' (equivalente a NULL) e' utilizzato per separare un SMS MO dall'altro.

Alcuni esempi:

01 0 NULL\r\n

Non vi sono nuovi SMS MO

01 1 2005-07-09\t18:09:05\t393471234567\tVOLA ciao come stai?\393202043252\tkey1\r\n
Un nuovo messaggio ricevuto il 9 luglio 2005 alle ore 18:09:05 dal numero +393471234567
con testo "VOLA ciao come stai?"

01 2 2005-07-09\t18:09:05\t393471234567\tciao ciao ciao\393202043252\tkey1\x1A\2003-12-
09\t18 :10 :01\t393351234567\tpippos\393202043252\tkey1\r\n
Due nuovi messaggi ricevuti, notare il carattere « \x1A\ » che separa un messaggio dall'altro.

3.3 Ack ricezione messaggi (CMD = 5)

Il comando AckMessagesMO esegue l'ack con il server per confermare la corretta ricezione dei messaggi SMS MO effettuata mediante l'ultima esecuzione del comando GetMessagesMO. Questo comando deve essere eseguito dopo il comando CMD=4 per confermare al gateway di VolaSMS l'avvenuta ricezione dei messaggi. Se il comando non viene inviato, il gateway di VolaSMS restituirà, alla ricezione del comando CMD=4, nuovamente l'elenco degli sms MO.

3.3.1 CMD (obbligatorio)

E' il parametro che indica il tipo di operazione ed ha il valore 5

3.3.2 UID (obbligatorio)

E' il parametro che contraddistingue l'account del cliente. Il valore deve essere nel formato MD5

3.3.3 PWD (obbligatorio)

E' il parametro che contraddistingue la password dell'account del cliente. Il valore deve essere nel formato MD5

3.3.4 SERIAL (obbligatorio)

Numero seriale che va valorizzato obbligatoriamente con **TR45GDLBO730HDIUEQJ5**.

3.3.5 Risposta positiva del gateway

Se l'esecuzione del comando e' avvenuta correttamente il gateway risponde al client nel seguente modo:

01\r\n

3.4 Comando CheckMessagesMO (CMD=6)

Il comando CheckMessagesMO controlla se vi sono nuovi messaggi MO sul server e restituisce il numero di nuovi messaggi ricevuti. Quindi a differenza del comando GetMessageMO (CMD=4) non viene restituita la lista dei messaggi.

3.4.1 CMD (obbligatorio)

E' il parametro che indica il tipo di operazione ed ha il valore **6**

3.4.2 UID (obbligatorio)

E' il parametro che contraddistingue l'account del cliente. Il valore deve essere nel formato MD5

3.4.3 PWD (obbligatorio)

E' il parametro che contraddistingue la password dell'account del cliente. Il valore deve essere nel formato MD5

3.4.4 SERIAL (obbligatorio)

Numero seriale che va valorizzato obbligatoriamente con **TR45GDLBO730HDIUEQJ5**.

3.4.5 KEY (facoltativo)

E' il valore della chiave associata agli MO. Se viene specificato il valore verra' restituito il numero degli MO nuovi ricevuti dell'utente associati alla chiave di ricezione. Se il valore non viene specificato viene restituito il numero totale degli MO nuovi ricevuti associati all'account.

3.4.6 Risposta positiva del gateway

Se l'esecuzione del comando e' avvenuta correttamente il gateway risponde al client nel seguente modo:

01<space><num_mo>\r\n

dove <num_mo> e' il numero di messaggi SMS MO presenti sul server non ancora "scaricati" dal client.

3.5 Invio messaggio SendOptimized (CMD = 14)

Il comando SendOptimized consente l'invio di messaggi multipli in una unica richiesta, diminuendo così il numero di connessioni tra client e server ed aumentando la performance di comunicazione. Per migliorare ancora di più il flusso di comunicazione tra client e server ed evitare l'invio non voluto di SMS anche a fronte di errori di comunicazione è stato introdotto l'invio dei messaggi con richiesta di conferma. Vedere a tal proposito il paragrafo 3.6 (CMD 44).

3.5.1 CMD (obbligatorio)

È il parametro che indica il tipo di operazione ed ha il valore **14**

3.5.2 UID (obbligatorio)

È il parametro che contraddistingue l'account del cliente. Il valore deve essere nel formato MD5

3.5.3 PWD (obbligatorio)

È il parametro che contraddistingue la password dell'account del cliente. Il valore deve essere nel formato MD5

3.5.4 SERIAL (obbligatorio)

Numero seriale che va valorizzato obbligatoriamente con **TR45GDLBO730HDIUEQJ5**.

3.5.5 SENDDATA (obbligatorio)

È il parametro che contiene i dati relativi all'invio dei messaggi ed ha la seguente struttura:

<group_sms_list>

dove <group_sms_list> contiene la lista dei gruppi di sms da inviare separata dal carattere <cr_lf> (\r\n) ed ha la forma:

<group_sms><cr_lf><group_sms>

<group_sms> contiene le informazioni (separate dal carattere di tabulazione <tab> (\t)) relative ad un gruppo di invio ed ha la forma :

<cid><tab><from><tab><to_list><tab><msg><tab><date><tab><time>

<cid> indica il codice identificativo del gruppo di invio lato client. Il valore del <cid> può assumere un valore alfanumerico e deve essere univoco per ogni gruppo di sms specificato nella stessa richiesta.

Nel caso in cui fossero inseriti due identici <cid> il gateway non accetterà la richiesta e restituirà un codice di errore.

<from> è il mittente del gruppo di invio. È il mittente con cui verrà inviato l'sms al destinatario o ai destinatari del messaggio. Il campo avrà effetto solo se l'account utilizzata supporta l'invio di messaggi con campo mittente modificabile. In quest'ultimo caso il gateway di VolaSMS utilizzerà il valore di default "vola.it" o "SMS", nel caso in cui il profilo di listino impostato lo supporti.

Il campo può contenere al max 11 caratteri alfanumerici, oppure 14 caratteri numerici.

Nel caso in cui il campo non sia correttamente valorizzato il gateway restituirà un errore.

<to_list> è la lista dei destinatari separata dal carattere "," (virgola) che ha la seguente struttura:

<msisdn>,<msisdn>...

il parametro <msisdn> deve avere la forma:

<country><numero>

dove <country> e' il prefisso del paese internazionale cui il parametro <numero> si riferisce. **Attenzione! DEVE essere indicato il carattere "+" davanti al country code (Codice Paese Internazionale).**

Il gateway di VolaSMS effettua un controllo sul country code di ogni destinatario e nel caso in cui non lo supportasse, lo restituirà nella lista dei destinatari errati.

<msg> e' il testo del messaggio e **NON può contenere i caratteri <tab> e <cr_lf>.**

Se il testo del messaggio supera i 160 caratteri il gateway di VolaSMS effettua uno "split" del testo in modo che ad ogni destinatario vengano inviati N messaggi ciascuno lungo al massimo 160 caratteri. Il numero massimo di caratteri che può contenere il campo messaggio e' 640 caratteri.

<date> e' la data di invio del messaggio della forma yyyy-mm-gg (anno-mese-giorno)

<time> e' l'ora di invio del messaggio della forma hh:mm (ore:minuti)

per invii immediati <date> deve valere 0000-00-00 e <time> 00:00.

Di seguito alcuni esempi di valorizzazione del parametro SENDDATA

esempio 1: invio di un messaggio ad un destinatario:

1\tristiano\t+393471234567\tprova invio sms\t0000-00-00\t00:00

esempio 2: invio di un messaggio a 2 destinatari

1\tristiano\t+393471234567,+393357654321\tprova invio sms\t0000-00-00\t00:00

esempio 3: invio di piu' gruppi di messaggi

1\tristiano\t+393471234567,+393357654321\tprova invio sms test \t0000-00-00\t00:00\r\n2\trmittente\t+393471234567,+393357654321\ttest di invio 2\t2005-07-19\t15:00

Notare con in quest'ultimo esempio il <cid> sia diverso per ogni gruppo di messaggio

3.5.6 Parametro VCODE (facoltativo)

Questo parametro deve essere utilizzato per chi ha aderito al programma VAP (Vola Affiliate Program). Deve essere valorizzato con il codice alfanumerico (VAP CODE) assegnato nel momento della registrazione. Per maggiori informazioni inviare una mail a supporto@vola.it

3.5.7 Risposta positiva del gateway

Se i dati inviati al server reativi al parametro SENDDATA sono **sintatticamente corretti** la risposta inviata al client è la seguente:

01<space><response>\r\n

<response> contiene il risultato dell'invio del gruppo o gruppi di messaggi SMS inviati ed e' composta da una lista separata dal carattere di tabulazione <tab>.

<grp_resp>\t<grp_resp>

Ogni gruppo di risposa <grp_resp> e' composto da una serie di parametri separati dal carattere ";", (punto e virgola) :

<cid>;<orderid_list>;<to_err_list>

dove:

<cid> e' il codice identificativo del gruppo di invio del client
<orderid_list> e' la lista degli orderid attribuita dal gateway al gruppo di invio. Ogni order id e' separato dal carattere “,” (virgola):

<orderid>,<ordeid>,<orderid>

E' possibile che il gateway restituisca più di un <orderid> a fronte di un gruppo di invio, nel caso in cui il testo del messaggio inviato supera i 160 caratteri ed il gateway di VolaSMS supporta il delivery di sms con testo maggiore di 160 caratteri. Se il testo del messaggio non supera i 160 caratteri allora la lista di orderid sarà costituita da un solo elemento.

Se il testo del messaggio supera i 160 caratteri il gateway di VolaSMS effettua uno “split” del testo in modo che ad ogni destinatario vengano inviati N messaggi ciascuno lungo al massimo 160 caratteri. Il numero massimo di caratteri che può contenere il campo messaggio e' 640 caratteri.

<to_err_list>: e' la lista dei numeri di cellulare che il gateway non ha accettato a seguito dei controlli interni effettuati e per i quali non e' stato inviato all'operatore di telefonia il relativo messaggio SMS. La presenza di numeri di telefono in questa lista sta ad indicare un errato formato del numero di destinazione

Il parametro e' valorizzato a con la stringa “NULL” nel caso in cui tutti i destinatari sono stati accettati dal gateway.

Esempio di risposta:

risposta relativa ad un solo gruppo di invio <cid>=12345, il paramentro <to_err_list> vale null perche' tutti i destinatari sono stati accettati

01 12345;334455;null\r\n

risposta relativa ad un solo gruppo di invio <cid>=12345, con un destinatario errato. L'order id vale 0 perche' il destintario era unico.

01 12345;0;393471234567\r\n

risposta relativa ad un solo gruppo di invio <cid>=12345, con un destinatario errato. L'order id e' diverso da 0 perché i destinatari erano più di uno

01 12345;334455;393471234567\r\n

riposta relativa ad un solo gruppo di invio <cid>=12345, con generazione di due orderid.

Si e' trattato di uno splitting del messaggio, ovvero verranno consegnati due messaggi (pari al numero di orderid restituiti) ad ogni destinatario.

01 12345;334455,334456;null\r\n

risposta relativa a 2 gruppi di invio, tutti i destinatari di ogni gruppo di invio sono stati accettati dal server

01 12345;334455;null\t12346;334456;null\r\n

Nel caso in cui i dati contenuti all'interno del parametro SENDATA non sono corretti (sintassi errata) il server risponderà con uno dei codici di ERRORE del gateway descritti precedentemente. Questo significa che l'itero invio non e' stato effettuato.

3.6 Invio messaggi SendOptimized con richiesta di conferma (CMD = 44)

Per la documentazione dell'invio con richiesta di conferma fare riferimento alla documentazione del comando **14**.

La differenza tra i due diversi comandi sta nel fatto che, mentre con il comando **14** gli SMS vengono inviati immediatamente, con il comando **44** gli SMS vengono parcheggiati sul gateway di Vola in attesa di ricevere una conferma di invio (vedi 3.7).

NB: si consiglia l'utilizzo dell'invio con conferma solo quando si devono inviare piu' messaggi all'interno della stessa richiesta. Se viene inviato 1 solo messaggio utilizzare il comando 14.

3.7 Conferma di invio messaggi (CMD 45)

Per i messaggi che sono stati inviati al Gateway di Vola tramite il comando **44** è necessario inviare una conferma al gateway per far sì che vengano effettivamente inviati.

3.7.1 CMD (obbligatorio)

E' il parametro che indica il tipo di operazione ed ha il valore **45**

3.7.2 UID (obbligatorio)

E' il parametro che contraddistingue l'account del cliente. Il valore deve essere nel formato MD5

3.7.3 PWD (obbligatorio)

E' il parametro che contraddistingue la password dell'account del cliente. Il valore deve essere nel formato MD5

3.7.4 SERIAL (obbligatorio)

Numero seriale che va valorizzato obbligatoriamente con **TR45GDLBO730HDIUEQJ5**.

3.7.5 ORDERID (obbligatorio)

E' la lista degli orderid ricevuta come risposta dal comando **44**.

3.7.6 Risposta positiva del gateway

Se l'esecuzione del comando e' avvenuta correttamente il gateway risponde al client nel seguente modo:

01\r\n

3.8 Richiesta notifiche (CMD = 10)

Permette di conoscere lo status di invio di tutti gli sms nel caso in cui l'account utilizzata abbia abilitata la funzionalità di invio con notifica. In caso contrario lo stato di notifica restituito si riferisce alla sola presa in consegna del messaggio lato gateway VolaSMS.

Se il comando va a buon fine restituisce una lista contenente lo stato informativo di ricezione del messaggio. La lista ritornata può essere vuota. Questo sta ad indicare che lo status non è cambiato rispetto a quello ultimo ottenuto.

Questo tipo di comando permette di inviare al gateway più richieste di notifica contemporaneamente.

3.8.1 CMD (obbligatorio)

È il parametro che indica il tipo di operazione ed ha il valore **10**

3.8.2 UID (obbligatorio)

È il parametro che contraddistingue l'account del cliente. Il valore deve essere nel formato MD5

3.8.3 PWD (obbligatorio)

È il parametro che contraddistingue la password dell'account del cliente. Il valore deve essere nel formato MD5

3.8.4 SERIAL (obbligatorio)

Numero seriale che va valorizzato obbligatoriamente con **TR45GDLBO730HDIUEQJ5**.

3.8.5 QUERYDATA (obbligatorio)

È il parametro che contiene la lista degli orderid e del relativo MSISDN per cui si richiede di conoscere lo status di notifica.

Il parametro ha la seguente struttura:

<orderid_list>

dove <orderid_list> è la lista delle coppie orderid e msisdn per cui si richiede lo status. Ogni coppia è separata dal carattere “,” (punto e virgola)

<orderid_grp>;<orderid_grp>

<orderid_grp> è composto dalla coppia (separata dal carattere “:”) <orderid>:<msisdn>

<msisdn> può essere non specificato ed in tal caso deve assumere il valore “null”

esempio 1: richiesta di notifica di tutti i messaggi con identico orderid

123456:null

esempio: richiesta di notifica del messaggio con orderid e msisdn specifico

1234567:+393477654321

esempio: richiesta con l'indicazione di più orderid

1234567:null;1234568:null;1234569:null;

3.8.6 Risposta positiva del gateway

Se l'esecuzione del comando è avvenuta correttamente il gateway risponde al client nel seguente modo:

01<space><lista_notifiche>\r\n

dove :

<lista_notifiche> vale "NULL" se non e' stato trovato nessuna notifica associata ai parametri specificati nella richiesta. Questo può succedere se non viene trovato corrispondenza tra orderid e /o numero destinatario.

oppure

<lista_notifiche> contiene la lista delle notifiche raggruppate per orderid. Ogni elemento della lista e' separato dal carattere <cr_lf> (\r\n)
<notifica_grp>\r\n<notifica_grp>...

<notifica_grp> ha il seguente formato
<orderid><tab><status_list>

<orderid> e' il numero di ordine relativo ad un gruppo di invio. Se non specificato o se alfanumerico il gateway risponderà con 01 NULL.

<status_list> e' la lista delle notifiche riferite allo stesso <orderid> ed ha la seguente struttura:
<status>;<status>.....<status>

Ogni elemento <status> ha la seguente struttura:

<date>,<time>,<msisdn>,<status>,<reason_code>

<date>: ha il formato DD-MM-YYYY (data a cui si riferisce lo status di notifica)

<time>: ha il formato HH:mm:ss (ora a cui si riferisce lo status di notifica)

<msisdn>: e' il numero di cellulare a cui si riferisce la notifica e ha il seguente formato
<country_code><number> preceduto dal "+" iniziale: es. +393471234567

<status>: e' un valore numerico indicante lo status di notifica e può assumere i seguenti valori:

0 - SMS ha raggiunto il gateway di VolaSMS e verrà inoltrato verso l'SMSC dell'operatore in base alla data e ora d'invio impostata. Ad esempio nel caso differito.

2 - SMS ha raggiunto l'SMS-C dell'operatore. La data ora è quella data dall'operatore. Il messaggio sta per essere inoltrato alla rete GSM da parte dell'operatore.

3 - SMS **è stato** consegnato al destinatario. La data ora è quella data dall'operatore

4 - SMS **non è stato** consegnato da parte dell'operatore. Maggiori

<reason_code>: e' un valore numerico che indica il motivo per cui il messaggio non è stato consegnato sul terminale del destinatario. Il valore deve essere preso in considerazione solo nel caso in cui il campo <status> ha valore '4'.

Di seguito la tabella che mostra i valori possibili degli stati di notifica che può restituire il gateway di VolaSMS.

STATUS CODE	REASON CODE	DESCRIPTION
0	000	SMS ha raggiunto il gateway di VolaSMS e verrà inoltrato verso l'SMSC dell'operatore in base alla data e ora d'invio impostata. Ad esempio nel caso differito.
2	000	SMS ha raggiunto l'SMS-C dell'operatore. La data ora è quella data dall'operatore. Il messaggio sta per essere inoltrato alla rete GSM da parte dell'operatore. In attesa di ricevere lo stato di notifica da parte dell'operatore sull'esito dell'invio.
3	000	SMS è stato consegnato sul terminale del destinatario. La data ora è quella data dall'operatore
4	000	SMS non consegnato. Il numero del destinatario risulta essere sconosciuto da parte dell'operatore.
4	101	SMS non consegnato da parte dell'operatore dopo il periodo di tentativi previsti (max 48 ore), per impossibilità di consegna. Ad esempio il terminale del destinatario è rimasto spento o fuori copertura della rete GSM per un periodo superiore alle 48 ore. Il valore può variare in base alle impostazioni dipendenti dall'operatore stesso che si occupa di veicolare il messaggio.
4	103	SMS non consegnato. Il numero del destinatario risulta essere stato bloccato da parte dell'operatore.
4	108	SMS non consegnato da parte dell'operatore dopo il periodo di tentativi previsti (max 48 ore), per impossibilità di consegna. Ad esempio il terminale del destinatario è rimasto spento o fuori copertura della rete GSM per un periodo superiore alle 48 ore. Il valore può variare in base alle impostazioni dipendenti dall'operatore stesso che si occupa di veicolare il messaggio.
4	111	SMS non consegnato. Il terminale del destinatario è stato configurato per bloccare il flusso in entrata voce e/o dati.

esempio 1: nessuna notifica trovata
01 null

esempio 2: 1 notifica trovata (stesso orderid)
01 334455\t10-12-2003,05:10:00,+3923471234567,3

il Reasoncode 3 indica che il messaggio è stato consegnato sul terminale del destinatario.

esempio 3: 2 notifica trovata (stesso orderid)
01 334455\t10-12-2003,05:10:00,+3923471234567,2;10-12-2003,05:15:00,+3923471234568,3

Il primo messaggio è stato inoltrato alla rete GSM (reasoncode = 2), il secondo messaggio è stato consegnato sul terminale del destinatario (reasoncode = 3).

esempio 3: 2 notifiche con 2 orderid
01 334455\t10-12-2003,05:10:00,+3923471234567,3\r\n334456\t10-12-2003,05:15:00,+3923471234568,3

Entrambi i messaggi sono stati consegnato sul terminale dei destinatari.

3.9 Get autoresponder (CMD = 16)

3.9.1 CMD (obbligatorio)

E' il parametro che indica il tipo di operazione ed ha il valore **16**

3.9.2 UID (obbligatorio)

E' il parametro che contraddistingue l'account del cliente. Il valore deve essere nel formato MD5

3.9.3 PWD (obbligatorio)

E' il parametro che contraddistingue la password dell'account del cliente. Il valore deve essere nel formato MD5

3.9.4 SERIAL (obbligatorio)

Numero seriale che va valorizzato obbligatoriamente con **TR45GDLBO730HDIUEQJ5**.

3.9.5 KEY (obbligatorio)

3.9.6 Risposta positiva del gateway

Se l'esecuzione del comando e' avvenuta correttamente il gateway risponde al client nel seguente modo:

01<space><testo_autoresponder>\r\n

dove <testo_autoresponder> è il testo attivo per l'autoresponder.

3.10 Set autoresponder (CMD = 17)

3.10.1 CMD (obbligatorio)

E' il parametro che indica il tipo di operazione ed ha il valore **17**

3.10.2 UID (obbligatorio)

E' il parametro che contraddistingue l'account del cliente. Il valore deve essere nel formato MD5

3.10.3 PWD (obbligatorio)

E' il parametro che contraddistingue la password dell'account del cliente. Il valore deve essere nel formato MD5

3.10.4 SERIAL (obbligatorio)

Numero seriale che va valorizzato obbligatoriamente con **TR45GDLBO730HDIUEQJ5**.

3.10.5 AUTORESPONDER (facoltativo)

Testo del messaggio automatico da impostare. Se non specificato viene disattivata la funzionalità.

3.10.6 KEY (obbligatorio)

E' il valore della chiave associata agli MO.

3.10.7 Risposta positiva del gateway

Se l'esecuzione del comando e' avvenuta correttamente il gateway risponde al client nel seguente modo:

01\r\n

3.11 Inserimento di un nuovo mittente

3.11.1 CMD (obbligatorio)

E' il parametro che indica il tipo di operazione ed ha il valore **30**.

3.11.2 UID (obbligatorio)

E' il parametro che contraddistingue l'account del cliente. Il valore deve essere nel formato MD5.

3.11.3 PWD (obbligatorio)

E' il parametro che contraddistingue la password dell'account del cliente. Il valore deve essere nel formato MD5.

3.11.4 SERIAL (obbligatorio)

Numero seriale che va valorizzato obbligatoriamente con **TR45GDLBO730HDIUEQJ5**.

3.11.5 SENDER (obbligatorio)

Il nuovo mittente che si vuole inserire.

3.11.6 Risposta positiva del gateway

Se l'esecuzione del comando e' avvenuta correttamente il gateway risponde al client nel seguente modo:

01\r\n

3.11.7 Codici di errore

Se l'esecuzione del comando ha generato un errore, il gateway risponde al client nel seguente modo:

XX\r\n

Dove XX può assumere i valori:

- 40, è stato raggiunto il limite massimo di mittenti impostabili per l'account
- 41, il mittente richiesto è un numero di cellulare e richiede di essere autenticato (vedi 3.10)
- 42, il pincode è stato inviato sul numero di cellulare da autenticare
- 43, si è verificato un problema per cui non è stato possibile inviare il pincode sul numero di cellulare da autenticare
- 44, il mittente richiesto esiste già nella lista dei mittenti impostati per l'account
- 45, il mittente richiesto non è utilizzabile
- 46, il mittente richiesto non ha una lunghezza valida (11 caratteri alfanumerici oppure 16 numerici)

3.12 Richiesta pincode per autenticare un numero di cellulare

L'autenticazione è necessaria solo per i mittenti che sono riconosciuti come numeri di cellulare.

NB: L'esecuzione del comando genera l'invio di un SMS contenente un pincode sul numero di cellulare da autenticare.

3.12.1 CMD (obbligatorio)

E' il parametro che indica il tipo di operazione ed ha il valore **34**.

3.12.2 UID (obbligatorio)

E' il parametro che contraddistingue l'account del cliente. Il valore deve essere nel formato MD5.

3.12.3 PWD (obbligatorio)

E' il parametro che contraddistingue la password dell'account del cliente. Il valore deve essere nel formato MD5.

3.12.4 SERIAL (obbligatorio)

Numero seriale che va valorizzato obbligatoriamente con **TR45GDLBO730HDIUEQJ5**.

3.12.5 SENDER (obbligatorio)

Il mittente (numero di cellulare) da autenticare tramite l'invio del pincode.

3.12.6 Risposta positiva del gateway

Se l'esecuzione del comando e' avvenuta correttamente il gateway risponde al client nel seguente modo:

42\r\n

3.13 Validazione del pincode per l'autenticazione di un mittente (numero di cellulare)

3.13.1 CMD (obbligatorio)

E' il parametro che indica il tipo di operazione ed ha il valore **33**.

3.13.2 UID (obbligatorio)

E' il parametro che contraddistingue l'account del cliente. Il valore deve essere nel formato MD5.

3.13.3 PWD (obbligatorio)

E' il parametro che contraddistingue la password dell'account del cliente. Il valore deve essere nel formato MD5.

3.13.4 SERIAL (obbligatorio)

Numero seriale che va valorizzato obbligatoriamente con **TR45GDLBO730HDIUEQJ5**.

3.13.5 SENDER (obbligatorio)

Il mittente da cancellare.

3.13.6 PINCODE (obbligatorio)

Il codice alfanumerico ricevuto via SMS sul numero di cellulare da autenticare come mittente.

3.13.7 Risposta positiva del gateway

Se l'esecuzione del comando e' avvenuta correttamente il gateway risponde al client nel seguente modo:

01\r\n

3.14 Elenco mittenti impostati

3.14.1 CMD (obbligatorio)

E' il parametro che indica il tipo di operazione ed ha il valore **32**.

3.14.2 UID (obbligatorio)

E' il parametro che contraddistingue l'account del cliente. Il valore deve essere nel formato MD5.

3.14.3 PWD (obbligatorio)

E' il parametro che contraddistingue la password dell'account del cliente. Il valore deve essere nel formato MD5.

3.14.4 SERIAL (obbligatorio)

Numero seriale che va valorizzato obbligatoriamente con **TR45GDLBO730HDIUEQJ5**.

3.14.5 Risposta positiva del gateway

Se l'esecuzione del comando e' avvenuta correttamente il gateway risponde al client nel seguente modo:

01<space><list>\r\n

dove:

<list> è una stringa contenente l'elenco dei mittenti impostati ed ha il seguente formato
<num_mitt>:<status> dove i mittenti sono separati dal carattere ','.

<status> può assumere i seguenti significati:

- P: mittente predefinito
- V: mittente Vola
- A: mittente autenticato
- T: mittente da autenticare

3.15 Cancellazione di un mittente

3.15.1 CMD (obbligatorio)

E' il parametro che indica il tipo di operazione ed ha il valore **31**.

3.15.2 UID (obbligatorio)

E' il parametro che contraddistingue l'account del cliente. Il valore deve essere nel formato MD5.

3.15.3 PWD (obbligatorio)

E' il parametro che contraddistingue la password dell'account del cliente. Il valore deve essere nel formato MD5.

3.15.4 SERIAL (obbligatorio)

Numero seriale che va valorizzato obbligatoriamente con **TR45GDLBO730HDIUEQJ5**.

3.15.4 SENDER (obbligatorio)

Il mittente da cancellare.

3.15.5 Risposta positiva del gateway

Se l'esecuzione del comando e' avvenuta correttamente il gateway risponde al client nel seguente modo:

01\r\n

4 Esempi script PHP

Non gestendo in maniera completa tutti gli aspetti legati al funzionamento del gateway di VolaSMS, gli script riportati di seguito devono essere presi come uno spunto per la realizzazione delle proprie applicazioni.

Per qualsiasi informazione o spiegazione in merito agli esempi si prega di inviare una mail a supporto@vola.it.

4.1 Richiesta credito residuo SMS

```
<?php

$res=getCredit("xxxxx", "yyyyy", "TR45GDLBO730HDUIEQJ5");

print_r($res);

exit;

#
# Funzione che effettua la richiesta del credito.
# Viene restituito un array associativo:
# 'CODE_RESPONSE'=> codice esito comando,
# 'CREDIT'=> credito residuo dell'account se il comando ha avuto esito positivo
#
function getCredit($volauser, $volapwd,$serial) {
    $sito_web="sms.vola.it";
    $porta=80;
    $script="/cgi/volasms_gw_plus2.php";
    $ok=false;
    $fs=fsockopen($sito_web,$porta);

    if($fs) {
        $header = "";
        $header = $header . "POST $script HTTP/1.1\r\n";
        $header = $header . "Host: $sito_web\r\n";
        $header = $header . "Content-type: application/x-www-form-urlencoded\r\n";
        $volauser = md5($volauser);
        $volapwd = md5($volapwd);
        $senddata = rawurlencode($senddata);
        $str_data = "UID=".$volauser."&PWD=".$volapwd."&CMD=2&SERIAL=".$serial;
        $length = strlen($str_data);
        $header = $header."Content-length: $length\r\n\r\n";
        $header = $header.$str_data;

        fputs($fs,$header);
        $continue=true;
        while($continue) {
            $response = fgets($fs);
            if(ereg("[0-9]", $response)) {
                $continue=false;
                $code=substr($response,0,2);
                $tmp=split(" ", $response);
                $credit=$tmp[1];
                $res = array ('CODE_RESPONSE'=>$code,
                             'CREDIT'=>$credit);
            }
        }
        return $res;
    }
}

?>
```

4.2 Ricezione messaggi

Si ricorda che per ricevere i messaggi SMS MO (Mobile Originated) attraverso il gateway di vola.it e' necessario avere associata all'account utente almeno una chiave di ricezione.

```
<?php

$user="xxxx";
$pwd="yyy";
$key="valore";

$msgs= getIncomingMessages($user,$pwd,$key);
if (count($msgs)>0) {
    print_r($msgs);
    # confermo l'avvenuta ricezione dei messaggi al server in modo
    # che alla successiva chiamata di getIncomingMessage
    # non vengano restituiti gli stessi messaggi
    ackIncomingMessages($user,$pwd);
}

exit;

#
# Funzione conferma ricezione di SMS MO in modo da non restituirli alla successiva
# chiamata di getIncomingMessage
#
function ackIncomingMessages($volouser, $volapwd) {
    // configurazione
    $sito_web="sms.vola.it";
    $porta = 80;
    $script = "/cgi/volasms_gw_plus2.php";
    $fs = fsockopen($sito_web,$porta);
    if($fs) {
        $header = "";
        $header = $header . "POST $script HTTP/1.0\r\n";
        $header = $header . "Host: $sito_web\r\n";
        $header = $header . "Content-type: application/x-www-form-urlencoded\r\n";

        //impostazione stringa parametri e sua lunghezza
        $string = "UID=".md5($volouser)."&PWD=".md5($volapwd)
            . "&CMD=5&SERIAL=TR45GDLB0730HDUIEQJ5";
        $length = strlen($string);

        $header = $header . "Content-length: $length \r\n\r\n";
        $header = $header . $string;

        // invio i dati al socket
        fputs($fs,$header);

        $line = TRUE;

        // ricerca del risultato dell'invio
        while($line) {
            $line = fgets($fs);
            if (eregi("<body>", $line)) {
                while ($line) {
                    $line = fgets($fs);
                    if (eregi("</body>", $line)) $line=FALSE;
                    else if ($line) $response.=$line;
                }
            }
        }

        $a_datas = explode(" ", $response);
        $code=$a_datas[0];
        switch($code) {
            case "01":
                $res=TRUE;
                break;

            default:
                break;
        }
    }

    return $res;
}

#
```

```
# Funzione che effettua la richiesta al gateway di vola.it e restituisce un array
# con l'elenco dei messaggi sms MO.
# Se non vi sono messaggi, oppure il comando fallisce (username e pwd errati) l'array sara' vuoto.
#
function getIncomingMessages($volauser, $volapwd, $key) {
    // configurazione
    $sito_web="sms.vola.it";
    $porta = 80;
    $script = "/cgi/volasms_gw_plus2.php";
    $a_msg= array();
    $fs = fsockopen($sito_web,$porta);
    if($fs) {
        $header = "";
        $header = $header . "POST $script HTTP/1.0\r\n";
        $header = $header . "Host: $sito_web\r\n";
        $header = $header . "Content-type: application/x-www-form-urlencoded\r\n";

        //impostazione stringa parametri e sua lunghezza
        $string = "UID=".md5($volauser)."&PWD=".md5($volapwd)
            . "&CMD=4&KEY=$key&SERIAL=TR45GDLBO730HDIUEQJ5";
        $length = strlen($string);

        $header = $header . "Content-length: $length \r\n\r\n";
        $header = $header . $string;

        // invio i dati al socket
        fputs($fs,$header);

        $line = TRUE;

        // ricerca del risultato dell'invio
        while($line) {
            $line = fgets($fs);
            if (eregi("<body>", $line)) {
                while ($line) {
                    $line = fgets($fs);
                    if (eregi("</body>", $line)) $line=FALSE;
                    else if ($line) $response.=$line;
                }
            }
        }

        $a_datas = explode(" ", $response);
        $code=$a_datas[0];
        switch($code) {
            case "01":
                if ($a_datas[1]>0) {
                    $a_msg = explode("\x1F", substr($response, 3+strlen($a_datas[1])+1));
                }
                break;
            default:
                break;
        }
    }

    return $a_msg;
}
?>
```

4.3 Invio SMS

```
<?php
// Il numero di cellulare del destinatario deve contenere il "+" iniziale
$messaggio = "Prova invio messaggio";
$senddata = getSendData("mittente", "+39347xxxxxx", $messaggio);

$res=sendSMS("uuu", "xxxxx", $senddata, "TR45GDLB0730HDUIEQJ5");

print_r($res);
exit;

#
# Funzione che restituisce il contenuto del parametro SENDATA opportunamente
# valorizzato.
#
function getSendData($from, $to, $msg) {
    $res = "0\t$from\t$to\t$msg\t0000-00-00\t00:00\r\n";
    return $res;
}

#
# Funzione che effettua l'invio dei dati a gateway.
# Viene restituito un array associativo:
# 'CODE_RESPONSE'=> codice esito comando,
# 'CID'=> identificativo invio lato cliente,
# 'ORDERID'=> numero ordine relativo all'invio,
# 'TO_ERRORO_LIST' => elenco numeri destinatari non accettati dal gateway
# Esempio:
# Array
# (
#     [CODE_RESPONSE] => 01
#     [CID] => 0
#     [ORDERID] => Array
#         (
#             [0] => 17549187
#         )
#     [TO_ERRORO_LIST] => Array
#         (
#             [0] =>
#         )
# )
#)
function sendSMS($volauser, $volapwd, $senddata, $serial, $test=0) {
    $sito_web="sms.vola.it";
    $porta=80;
    $script="/cgi/volasms_gw_plus2.php";
    $ok=false;
    $fs=fsockopen($sito_web,$porta);

    if($fs) {
        $header = "";
        $header = $header . "POST $script HTTP/1.1\r\n";
        $header = $header . "Host: $sito_web\r\n";
        $header = $header . "Content-type: application/x-www-form-urlencoded\r\n";
        $volauser = md5($volauser);
        $volapwd = md5($volapwd);
        $senddata = rawurlencode($senddata);
        $str_data = "TEST=".$test."&UID=".$volauser."&PWD=".$volapwd.
            "&CMD=14&SENDATA=".$senddata."&SERIAL=".$serial;
        $length = strlen($str_data);
        $header = $header."Content-length: $length\r\n\r\n";
        $header = $header.$str_data;

        fputs($fs,$header);
        $continue=true;
        while($continue) {
            $response = fgets($fs);
            if(ereg("[0-9]", $response)) {
                $continue=false;
                $code=substr($response,0,2);
                $tmp=split(" ", $response);
                $params=$tmp[1];
                $tmp = split(";", $params);
                $cid= $tmp[0];
                $lst_orderid = split(",", $tmp[1]);
                $lst_toerror = split(",", $tmp[2]);
                $res = array ('CODE_RESPONSE'=>$code,
                    'CID'=>$cid,
                    'ORDERID'=>$lst_orderid,
```

```

    }
    return $res;
}

?> 'TO_ERRORO_LIST'=>{$!st_toerror});
    }
}

```

4.4 Richiesta notifiche

```
<?php
```

```
#
# Esempio di richiesta per solo ORDERID
#
$querydata = getQueryData("17424970", "null");
$res=getNotifiche("xxxxx", "yyyyy", $querydata, "TR45GDLBO730HDUIEQJ5");
print_r($res);
```

```
#
# Esempio di richiesta per ORDERID e numero di cellulare
#
$querydata = getQueryData("17424970", "+39347xxxxxx");
$res=getNotifiche("xxxxx", "yyyyy", $querydata, "TR45GDLBO730HDUIEQJ5");
print_r($res);
```

```
exit;
```

```
function getQueryData($orderid, $msisdn) {
    $res = $orderid.":".$msisdn;
    return $res;
}
```

```
#
# La funzione restituisce un array associativo di questo tipo:
```

```

# La funzione restituisce un array associativo
#[CODE_RES] => 01
# [ORDERID] => 17424970
# [NOTIFY_LIST] => Array
# (
# [0] => Array
# (
# [DATE] => 2006-07-05
# [TIME] => 13:55:11
# [MSISDN] => +39335xxxxxx
# [STATUS] => 3
# [REASON] => 000
# )
# [1] => Array
# (
# [DATE] => 2006-07-05
# [TIME] => 12:42:00
# [MSISDN] => +39335xxxxxx
# [STATUS] => 3
# [REASON] => 000
# )
# )
# )

```

```
function getNotifiche($volauser, $volapwd, $querydata, $serial){
    // configurazione
    $sito_web="sms.vola.it";
    $porta = 80;
    $script = "/cgi/volasms_gw_plus2.php";
    $ok = false;
    $fs = fsockopen($sito_web,$porta);
    $res = array();
    if($fs) {
        $header = "";
        $header = $header . "POST $script HTTP/1.1\r\n";
        $header = $header . "Host: $sito_web\r\n";
        $header = $header . "Content-type: application/x-www-form-urlencoded\r\n";
        $volauser = md5($volauser); // user per vola
        $volapwd = md5($volapwd); // password per vola
        $querydata = rawurlencode($querydata);
        // impostazione stringa parametri e sua lunghezza
        $string = "UID=$volauser&PWD=$volapwd"
    }
}
```

```

        ."&CMD=10&SERIAL=$serial&QUERYDATA=$querydata";
        $length = strlen($string);
        $header = $header . "Content-length: $length \r\n\r\n";
        $header = $header . $string;
        // invio i dati al socket
        fputs($fs,$header);
        $continue = true;
        // ricerca del risultato dell'invio
        while($continue) {
            $response = fgets($fs);
            if(ereg("[0-9]", $response)) {
                $continue = false;
                $tmp = split(" ", $response);
                if ($tmp[1]!='NULL') {
                    $tmp2 = split("\t", $tmp[1]);
                    $tmp3 = split(";", $tmp2[1]);
                    for ($i=0; $i<count($tmp3); $i++) {
                        $n=split(" ", $tmp3[$i]);
                        $a_notifiche[]=array('DATE'=>$n[0],
                                                'TIME'=>$n[1],
                                                'MSISDN'=>$n[2],
                                                'STATUS'=>$n[3],
                                                'REASON'=>$n[4]);
                    }
                }
            }
            else $notify = array();
            $res = array('CODE_RES'=>$tmp[0],
                        'ORDERID'=>$tmp2[0],
                        'NOTIFY_LIST'=>$a_notifiche);
        }
    }
    return $res;
}
?>

```

4.5 Inserimento di un nuovo mittente

```

<?php
// Impostazioni account vola
$volauser = "xxxxx";
$volapswd = "yyyyyyyy";
$sender = "TESTSENDER";
// Inserimento di un mittente
$result = insSender($volauser, $volapswd, $sender);
echo $result;
exit;

function insSender($volauser, $volapswd, $SENDER) {
    $ip = $_SERVER["REMOTE_ADDR"];

    $CMD = 30;
    if(preg_match('^\\+\\d+', $SENDER))
        $mittente = "00".substr($SENDER, 1);
    else if(preg_match('^393\\d+', $SENDER) && strlen($SENDER)>10 )
        $mittente = "00".$SENDER;
    else
        $mittente = $SENDER;
    $UID = md5($volauser);
    $PWD = md5($volapswd);
    $str_data = "UID=" . rawurlencode($UID)
        . "&PWD=" . rawurlencode($PWD)
        . "&SERIAL=" . rawurlencode("TR45GDLBO730HDIUEQJ5")
        . "&CMD=" . $CMD
        . "&SENDER=" . rawurlencode($mittente)
        . "&PINCODE=" . rawurlencode($PINCODE);
    $response = sendCommandToGateway($str_data, $CMD);
    return $response;
}

function sendCommandToGateway($params, $cmd){
    $url = "http://sms.vola.it/cgi/volasms_gw_plus2.php";
    $port = "80";
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_HTTPHEADER, array("Host: sms.vola.it"));
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $params);
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_PORT, $port);
    curl_setopt($ch, CURLOPT_TIMEOUT, 60);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    $result=curl_exec($ch);
    if (!empty($result)){
        curl_close ($ch);
        return $result;
    }else
        return null;
}
?>

```


4.6 Inserimento di un numero di cellulare come mittente

```

<?php
// Impostazioni account Vola
$volouser = "xxxxxxx";
$volapswd = "yyyyyyy";
$sender = "339xxxxxxx";

// Inserimento del mittente
insSender($volouser, $volapswd, $sender);

// Autenticazione del mittente attraverso il pincode ricevuto via SMS
$PINCODE = 'xxxxxx';

$result = confirmPincode($volouser, $volapswd, $sender, $PINCODE);

echo $result;
exit;

function insSender($volouser, $volapswd, $SENDER) {
    $ip = $_SERVER["REMOTE_ADDR"];
    $CMD = 34;
    if(preg_match('/^\+\d+', $SENDER))
        $mittente = "00".substr($SENDER, 1);
    else if(preg_match('/^\+393\d+', $SENDER) && strlen($SENDER)>10 )
        $mittente = "00".$SENDER;
    else
        $mittente = $SENDER;
    $UID = md5($volouser);
    $PWD = md5($volapswd);
    $str_data = "UID=" . rawurlencode($UID)
        . "&PWD=" . rawurlencode($PWD)
        . "&SERIAL=" . rawurlencode("TR45GDLBO730HDIUEQJ5")
        . "&CMD=" . $CMD
        . "&SENDER=" . rawurlencode($mittente)
        . "&PINCODE=" . rawurlencode($PINCODE);
    $response = sendCommandToGateway($str_data);
    return $response;
}

function confirmPincode($volouser, $volapswd, $SENDER, $PINCODE) {
    if(preg_match('/^\+\d+', $SENDER))
        $mittente = "00".substr($SENDER, 1);
    else
        $mittente = $SENDER;

    $UID = md5($volouser);
    $PWD = md5($volapswd);
    $str_data = "UID=" . rawurlencode($UID)
        . "&PWD=" . rawurlencode($PWD)
        . "&SERIAL=" . rawurlencode("TR45GDLBO730HDIUEQJ5")
        . "&CMD=33"
        . "&SENDER=" . rawurlencode($mittente)
        . "&PINCODE=" . rawurlencode($PINCODE);
    $response = sendCommandToGateway($str_data);

    return $response;
}

function sendCommandToGateway($params){
    $url = "http://sms.vola.it/cgi/volasms_gw_plus2.php";
    $port = "80";
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_HTTPHEADER, array("Host: sms.vola.it"));
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $params);
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_PORT, $port);
    curl_setopt($ch, CURLOPT_TIMEOUT, 60);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    $result=curl_exec($ch);
    if (!empty($result)){
        curl_close ($ch);
        return $result;
    }else
        return null;
}
?>

```

4.7 Invio SMS con metodo di conferma

```

<?php
// Impostazioni messaggio
$message = "Prova invio messaggio";
$mittente = "TESTSENDER";
$destinatario = "+39xxxxxxxxx";

// Impostazioni account vola
$user = "xxxxx";
$password = "yyyyy";
$serial = "TR45GDLBO730HDUIEQJ5";

$senddata = getSendData($mittente, $destinatario, $message);

// Invio messaggio al server
$result = sendSMS($user, $password, $senddata, $serial);

// Conferma di invio
list($esito, $info) = explode(" ", $result);

if($esito == "01") {
    list($cid, $lst_orderid) = explode(";", $info);

    $order_id = explode(",", $lst_orderid);

    for($i=0; $i<count($order_id); $i++) {
        if($order_id[$i] != "") {
            $resc = confirmOrderid($user, $password, $order_id[$i], $serial);

            echo $order_id[$i] . ": " . $resc . "\r\n";
        }
    }
} else {
    echo "Errore nell'invio dei messaggi al server: " . $esito . "\r\n";
}

exit;

#
# Funzione che restituisce il contenuto del parametro SENDATA opportunamente
# valorizzato.
#
function getSendData($from, $to, $msg) {
    $res = "0\t$from\t$to\t$msg\t2009-01-01\t00:00\r\n";
    return $res;
}

#
# Funzione che inoltra gli SMS da inviare al Gateway di Vola
#
function sendSMS($volouser, $volapswd, $senddata, $serial) {
    $CMD = 44;

    $UID = md5($volouser);
    $PWD = md5($volapswd);

    $str_data = "UID=" . rawurlencode($UID)
        . "&PWD=" . rawurlencode($PWD)
        . "&SERIAL=" . rawurlencode($serial)
        . "&CMD=" . $CMD
        . "&SENDATA=" . rawurlencode($senddata);

    $response = sendCommandToGateway($str_data);

    resCleaner($response);

    return $response;
}

#
# Funzione che tramite l'orderid conferma al Gateway di Vola l'invio degli SMS
#
function confirmOrderid($volouser, $volapswd, $orderid, $serial) {
    $CMD = 45;

    $UID = md5($volouser);
    $PWD = md5($volapswd);

    $str_data = "UID=" . rawurlencode($UID)

```

```

        . "&PWD=" . rawurlencode($PWD)
        . "&SERIAL=" . rawurlencode($serial)
        . "&CMD=" . $CMD
        . "&ORDERID=" . rawurlencode($orderid);

    $response = sendCommandToGateway($str_data);
    resCleaner($response);
    return $response;
}

#
# Funzione che ripulisce la risposta del server dai tag HTML
#
function resCleaner(&$p_data) {
    $p_data = str_replace("\r\n", "", $p_data);
    $p_data = str_replace("<html>", "", $p_data);
    $p_data = str_replace("</html>", "", $p_data);
    $p_data = str_replace("<body>", "", $p_data);
    $p_data = str_replace("</body>", "", $p_data);
}

#
# Funzione che invia un comando al Gateway di Vola attraverso l'utilizzo delle librerie CURL
#
function sendCommandToGateway($params){
    $url = "http://sms.vola.it/cgi/volasms_gw_plus2.php";
    $port = "80";

    $ch = curl_init();

    curl_setopt($ch, CURLOPT_HTTPHEADER, array("Host: sms.vola.it"));
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $params);
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_PORT, $port);
    curl_setopt($ch, CURLOPT_TIMEOUT, 60);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    $result=curl_exec($ch);
    if (!empty($result)){
        curl_close ($ch);
        return $result;
    }else
        return null;
}
?>

```

5. Principali problemi

5.1 Problemi generici

Codice Errore	Possibili cause
88 GW_NULL_LICENZE	<ul style="list-style-type: none"> Verificare di aver valorizzato in maniera opportuna il parametro SERIAL. Il valore che deve assumere è: TR45GDLBO730HDIUEQJ5
89 GW_CMD_ERROR	<ul style="list-style-type: none"> Verificare di aver valorizzato tutti i parametri indicati come obbligatori Se si sta tentando di inviare un SMS (CMD 14 o 44) verificare di aver valorizzato date e ora di invio con dei valori validi (nel caso di invio immediato valorizzare come data=0000-00-00 ed ora=00:00).
97 GW_CREDIT_ERROR	<ul style="list-style-type: none"> Se si sta tentando di impostare o di recuperare il valore di un autoresponder verificare di aver specificato il parametro KEY. In fase di invio di uno o più SMS indica il caso in cui non si disponga di un credito sufficiente per completare l'invio. Il controllo del credito tiene conto anche se si è richiesto lo stato di notifica.
99 GW_AUTH_ERROR	<ul style="list-style-type: none"> Verificare di aver indicato correttamente i dati relativi al proprio account su VolaSMS. I dati vanno passati nel formato MD5.

5.2 Inserimento di un nuovo mittente (CMD 30)

Codice Errore	Possibili cause
40 GW_SENDERS_LIMIT_REACHED	<ul style="list-style-type: none"> Verificare che non sia stato raggiunto il numero massimo di mittenti impostabili
41 GW_SENDERS_TO_AUTH	<ul style="list-style-type: none"> Il mittente che si è chiesto di impostare è un numero di cellulare e necessita quindi dell'autenticazione tramite PINCODE per poter essere utilizzato
45 GW_SENDER_LIST_KO	<ul style="list-style-type: none"> Il mittente richiesto non è utilizzabile (non è possibile utilizzare mittenti che inizino con prefissi come 899, 144, 12 etc.)
46 GW_FIXED_SENDERS_BLOCK	<ul style="list-style-type: none"> L'account non è abilitato alla personalizzazione del mittente
47 GW_SENDER_INVALID	<ul style="list-style-type: none"> Se il mittente richiesto è alfanumerico la sua lunghezza è maggiore agli 11 caratteri consentiti. Se invece il mittente richiesto è alfanumerico la sua lunghezza è maggiore dei 16 caratteri consentiti

5.4 Richiesta di invio PINCODE per autenticazione numero di cellulare come mittente (CMD 34)

Codice Errore	Possibili cause
43 GW_SENDER_PINCODE_ KO	<ul style="list-style-type: none">• Si è verificato un problema con l'invio del PINCODE sul numero di cellulare da autenticare. Verificare che il credito residuo non sia esaurito
44 GW_SENDER_AUTH_ALR EADY_EXIST	
	<ul style="list-style-type: none">• Nella lista mittenti dell'utente esiste già un numero di cellulare per cui non è stata portata a termine l'operazione di autenticazione