

**PROMEMO**<sup>BIZ</sup>

# **Specifiche API**

## **Gestione utenti multipli**

v1.3.73 Last update: Ludovico 28/12/2009

## Sommario

Descrizione del servizio.....	3
Come diventare partner .....	3
Modalità di accesso al servizio con gestione utenti multipli .....	3
Sicurezza .....	3
Richieste e risposte .....	4
Area di test.....	4
Amministrazione account PARTNER .....	4
Elenco delle funzioni disponibili .....	5
Registrazione utente.....	5
Attivazione abbonamento .....	8
Ricarica crediti.....	11
Imposta messaggio .....	13
Modifica messaggio .....	16
Cancella messaggio impostato.....	18
Stato abbonamento .....	20
Lista messaggi inviati .....	22
Lista messaggi salvati.....	25
Stato messaggio .....	28
Aggiornamento informazioni account .....	31
Generazione del MAC .....	33
S2S per verifica attivazione abbonamento e ricarica crediti .....	34
Conclusioni.....	35
Appendice A: codici e messaggi di errore.....	36

## **Descrizione del servizio**

PROmemo.biz è una applicazione web che permette di inviare SMS ed E-mail (anche in date future) come promemoria per appuntamenti e ogni altro genere di comunicazione verso i vostri clienti e collaboratori.

Come funzionalità esclusiva PROmemo.biz permette di inviare SMS ed E-mail ricorrenti: è sufficiente impostare un messaggio una sola volta specificando frequenza di invio e data di scadenza, ed il messaggio verrà inviato automaticamente.

Oltre all'applicazione web, PROmemo.biz mette a disposizione delle API che permettono di integrare tutte le sue funzionalità all'interno di una applicazione desktop o di una applicazione web senza doversi occupare dell'effettivo invio di messaggi, della gestione delle code di SMS e mantenendo le caratteristiche fondamentali dell'applicazione.

## **Come diventare partner**

Per diventare partner di PROmemo.biz e ricevere le indicazioni per iniziare ad integrare e testare tutte le sue funzionalità è necessario effettuare la registrazione sul nostro sito web all'indirizzo <http://www.promemo.biz/partner/>.

Successivamente provvederemo a contattarvi e a fornirvi un identificativo PARTNER, l'autorizzazione ad accedere all'amministrazione del vostro account e l'accesso ad un ambiente di test.

## **Modalità di accesso al servizio con gestione utenti multipli**

Questa documentazione è specifica per il servizio con la gestione **multi utente**, in cui ogni partner ha la possibilità di dare accesso ai servizi di PROmemo.biz ad infiniti utenti senza preoccuparsi troppo della gestione degli abbonamenti, dei crediti residui, di tenere traccia dello storico dei messaggi inviati, etc.

Infatti tutta la gestione delle scadenze degli abbonamenti, dei crediti residui per ogni utente, dei rapporti sullo stato di invio dei messaggi e sulla corretta assegnazione dei vari messaggi ai vari utenti sarà a carico di PROmemo.biz che provvederà a gestire, per conto del partner, ogni esigenza di questo tipo.

Per effettuare l'integrazione è sufficiente inserire le opportune chiamate HTTPS alle API all'interno della propria applicazione, desktop o web che sia.

## **Sicurezza**

La connessione è sicura (https) con codice di autentica MAC basato su chiave privata.

Verrete notificati via email per ogni operazione eseguita che vi coinvolga come partner (notifica nuova iscrizione, notifica richiesta di attivazione abbonamento e ricarica crediti, notifica abbonamento o crediti in scadenza...)

Inoltre l'attivazione dell'abbonamento e delle ricariche crediti può essere soggetta ad un ulteriore livello di sicurezza grazie ad una chiamata "Server to Server" da PROmemo.biz ad un vostro server per avere conferma dell'autorizzazione ad attivare un abbonamento. Maggiori informazioni nella sezione **"S2S per verifica attivazione abbonamento e ricarica crediti"**

## ***Richieste e risposte***

Le richieste https devono essere inviate all'indirizzo **<https://www.promemo.biz/api.php>**?  
I valori dei parametri devono essere codificati secondo l'RFC 1738 (la specifica è consultabile sul sito <http://www.faqs.org/rfcs/rfc1738>).

Le risposte del server di PROMemo.biz saranno in formato XML.

Esempio di chiamata https per accedere al servizio:

```
https://www.promemo.biz/api.php?function=xxxx&parametro1=yyyy&parametro2=zzzzzz
```

Esempio di risposta XML:

```
<xml>  
  <state>OK</state>  
  <message>message saved</message>  
</xml>
```

## ***Area di test***

Per effettuare tutte le prove per l'integrazione è disponibile un'area di test.

Per poter usare l'area di test è necessario procedere alla registrazione come partner sul sito <http://www.promemo.biz/partner/>.

Provvederemo a contattarvi e ad inviarvi un **identificativo Partner** e l'accesso all'area di test con i relativi **subscription\_id** e **tupup\_id**.

L'area di test è disponibile sul sito <http://www.dev2.intelijam.com/promemo-biz/partner/>.

Tutte le operazioni effettuate nell'area di test avvengono utilizzando il protocollo **http** (e non il protocollo **https**).

Nell'area di test è possibile provare tutte le funzioni delle API, compreso l'invio dei messaggi.

In questa area però l'invio dei messaggi SMS è solo simulato (quindi non riceverete un SMS sul vostro telefono), ma è comunque possibile provare l'invio di messaggi ricorrenti, visualizzare lo storico dei messaggi inviati, la lista dei messaggi impostati e quant'altro.

Attenzione, i messaggi via E-mail invece verranno effettivamente inviati, quindi vi invitiamo a prestare attenzione agli indirizzi dei destinatari.

## ***Amministrazione account PARTNER***

Accedendo alle pagine di amministrazione del vostro account partner sul sito <http://www.promemo.biz/partner/admin/> avrete la possibilità di controllare lo stato del vostro account, modificare i vostri dati, verificare lo stato dei vostri utenti.

Sono anche disponibili funzioni per creare e modificare i vostri utenti, creare e rinnovare abbonamenti e ricariche per i vostri utenti.

Per queste funzioni sono anche disponibili le funzioni API che consentono di automatizzare completamente il processo di creazione utenti/abbonamento/ricarica.

Se non necessario e per maggiore semplicità abbiamo reso queste funzioni disponibili anche nell'area di amministrazione del vostro account.

## Elenco delle funzioni disponibili

### Registrazione utente

*NB. Funzione disponibile anche nel pannello di amministrazione Partner e che è necessario implementare via API solo per rendere il processo completamente automatico.*

Questa funzione permette di registrare un vostro utente. Durante la registrazione dovrete scegliere un identificativo univoco per il vostro cliente (non potete avere 2 vostri utenti distinti con lo stesso identificativo).

Questo identificativo verrà successivamente utilizzato per associare correttamente tutte le operazioni all'utente che le ha effettuate.

Per aumentare la sicurezza e la protezione dei vostri dati e di quelli dei vostri clienti la funzione di registrazione utente e tutte le altre operazioni avvengono utilizzando il protocollo sicuro **https** (esclusa l'area di test).

Nella tabella 1.1 mostriamo l'elenco dei parametri per la funzione di registrazione utente.

TABELLA 1.1 - Parametri per la funzione di registrazione utente

Nome Parametro	Descrizione	Obb?	Formato - Lunghezza	Valore / Esempio
<b>function</b>	Identifica la funzione che si vuole realizzare	SI	Testo - non specificata	Valore: signup
<b>user_id</b>	Identificativo univoco dell'utente. ( <i>nota1</i> )	SI	Email (vedi nota1), max 200 caratteri	Esempio: 6282@nomepartner.promemo.biz
<b>name</b>	Nome dell'utente	SI	Testo: 2-200 caratteri	Esempio: Mario
<b>surname</b>	Cognome dell'utente	SI	Testo: 2-200 caratteri	Esempio: Rossi
<b>password</b>	Password utente. Nelle successive operazioni verrà utilizzata per generare i codici di autenticazione ( <i>nota2</i> )	SI	Testo, 4-200 caratteri	Esempio: *****
<b>partner_email</b>	E-mail del partner alla quale inviare le notifiche relative a questo utente	SI	Email, max 200 caratteri	Esempio: abcde@example.com
<b>message_header</b>	Numero di telefono o testo che apparirà come mittente dei messaggi inviati	SI	Testo, max 11 caratteri	Esempio: 3381234567 oppure RossiS.p.A.
<b>company_name</b>	Ragione sociale azienda dell'utente	NO	Testo, 3-200 caratteri	Esempio: La magnifica Srl
<b>legal_address</b>	Indirizzo dell'utente	NO	Testo max 255 caratteri	Esempio: Via pinco pallo, 34
<b>zip_code</b>	Codice di avviamento postale (CAP) dell'utente	NO	Numerico, 5 caratteri	Esempio: 00100
<b>city</b>	Città dell'utente	NO	Testo 2-50 caratteri	Esempio: Roma
<b>province</b>	Provincia dell'utente	NO	2 caratteri testuali	Esempio: RM
<b>phone</b>	Telefono dell'utente	NO	Numeri 6-50 cifre	Esempio: 06123456
<b>category</b>	Categoria attività dell'utente	NO	Testo 2-200 caratteri	Esempio: Medici

**Nota1** *user\_id*, identificativo univoco del vostro cliente.

È una stringa che ha il formato di una E-mail (es: xxxxxxxx@PARTNER.promemo.biz).

La stringa è composta da una prima parte (prima della @) generata da voi e per voi univoca, dopo la @ inserirete l'identificativo partner (l'identificativo da noi rilasciato in fase di registrazione del partner) e per finire .promemo.biz che identifica il servizio.

Consigliamo di utilizzare un identificativo che sia legato all'utente anche in caso di reinstallazione del vostro software (in caso di applicazioni desktop) in modo da non perdere i messaggi, l'abbonamento ed i crediti di quell'utente.

Analogamente, qualora la licenza della vostra applicazione preveda di poter installare l'applicazione su più computer, è consigliabile fare in modo che tutte le installazioni usino lo stesso identificativo, così da poter gestire gli stessi avvisi da ogni postazione.

### **Nota2 Password utente**

La password comunicata con protocollo di comunicazione sicura HTTPS tramite la funzione di registrazione non verrà più richiesta in seguito, ma verrà utilizzata per generare i codici di autenticità delle successive chiamate. Per maggiori info sulla generazione dei codici di autentica fare riferimento al paragrafo "**Generazione del MAC**".

Come per lo user\_id, si consiglia di utilizzare una password legata all'utente per evitare di perderla, ad esempio, a causa di una reinstallazione del vostro software. Questa può essere ad esempio una password scelta dall'utente stesso o una password generata a partire dallo user\_id.

### **Nota3**

Molti dei campi richiesti per la registrazione non sono obbligatori, ma se forniti permetteranno una maggiore personalizzazione delle mail di notifica (che riceverete solo voi) e vi aiuteranno ad identificare il cliente che ha l'abbonamento in scadenza o che ha terminato i crediti.

Inoltre fornirà importanti dati statistici che vi aiuteranno a vendere meglio il prodotto.

Valori consigliati nel campo category:

Medici, ManutenzioneAuto, ManutenzioneDiversa, CommercialistiConsulenti, CRM, Veterinari, CentriEsteticiCuraPersona, Assicurazione, Altro.

### **Esempio di chiamata https alla funzione di registrazione utente**

Fare attenzione, i valori dei parametri devono essere codificati secondo l'[RFC 1738](#) (@=%40, spazio=%20...). Non bisogna codificare le "&" che separano le singole coppie chiave=valore.

```
https://www.promemo.biz/api.php?function=signup&name=pinco&surname=pallo&user_id=userapi5%40PARTNER.promemo.biz&password=jsjes87sa&partner_email=pincopallo%40yahoo.it&message_header=HEADER&company_name=Azienda%20S.p.a&legal_address=via%20roma%2C%201001&zip_code=32333&city=La%20spezia&province=LS&phone=39080232131
```

### **Risposta per la funzione di registrazione utente**

Le risposte sono in formato XML.

Per la funzione di registrazione utente la risposta potrà essere formata dai seguenti parametri:

```
<state>TEXT</state>
```

Indica se l'operazione richiesta è andata a buon fine. Il valore potrà essere OK (successo) oppure KO (fallimento)

```
<error_number>NUMBER</error_number>
```

Indica il codice univoco dell'errore che si è verificato.

```
<message>TEXT</message>
```

Descrive con una frase l'errore che si è verificato.

Contestualmente alla risposta positiva via http riceverete una mail (all'indirizzo specificato nel parametro **partner\_email**) come notifica dell'avvenuta registrazione di un nuovo utente.

### Esempio di risposta positiva

```
<xml>
  <state>OK</state>
  <message>signup done</message>
</xml>
```

### Esempio di risposta negativa

```
<xml>
  <state>KO</state>
  <error_number>0101</error_number>
  <message>User already registered</message>
</xml>
```

### Possibili codici di errore e messaggi di errore

0001 = The requested function doesn't exist  
0002 = Invalid user\_id format  
0003 = User\_id inactive  
0004 = The specified partner doesn't exists or is inactive  
0005 = Invaild MAC  
0006 = Missing required field (*seguono altre info sul campo mancante*)  
0007 = Wrong field format (*seguono altre info sul campo errato*)

0101 = User already registered  
0102 = Unexpected error (*seguono altre info sull'errore*)

## Attivazione abbonamento

*NB. Funzione disponibile anche nel pannello di amministrazione Partner e che è necessario implementare via API solo per rendere il processo completamente automatico.*

Questa funzione permette di creare ed attivare un abbonamento per un vostro cliente. Il cliente è identificato tramite la **user\_id** comunicata in fase di registrazione ed il tipo di abbonamento da attivare è definito tramite il parametro **subscription\_id**.

Le **subscription\_id** utilizzabili vi saranno comunicate in fase di registrazione ed identificano univocamente la durata dell'abbonamento. In genere gli abbonamenti non attribuiscono anche dei crediti per l'invio dei messaggi, si dovrà quindi procedere anche con l'operazione di ricarica crediti.

Questa funzione è utilizzata, oltre che per attivare abbonamenti anche per rinnovare quelli già attivi (in questo caso si faccia attenzione alla *nota4*).

Per aumentare la sicurezza e prevenire l'attivazioni di abbonamenti a vostra insaputa è possibile utilizzare una verifica server-to-server sincrona tra i nostri sistemi ed un vostro server prima di procedere all'effettiva attivazione dell'abbonamento.

In caso di successo il partner riceverà una email di notifica con le informazioni sull'abbonamento attivato.

Nella tabella 1.2 mostriamo l'elenco dei parametri per la funzione di attivazione abbonamento.

TABELLA 1.2 - Parametri per la funzione di attivazione abbonamento

Nome Parametro	Descrizione	Obb?	Formato - Lunghezza	Valore / Esempio
<b>function</b>	Identifica la funzione che si vuole realizzare	SI	Testo - non specificata	Valore: subscription
<b>user_id</b>	Identificativo univoco dell'utente.	SI	Email, max 200 caratteri	Esempio: 6282@nomepartner.promemo.biz
<b>subscription_id</b>	Tipo di abbonamento da associare al vostro utente ( <i>nota4</i> )	SI	Intero	Esempio: 2
<b>S2S_check_code</b>	Codice di attivazione che deve essere sottoposto a verifica S2S ( <i>nota5</i> )	SI se attivata la verifica S2S per il partner	Testo - non specificato	Esempio: SS7J39SY3JS9JS9
<b>mac</b>	Stringa opportunamente costruita ed utilizzata per verificare l'autenticità della funzione richiesta ( <i>nota6</i> )	SI	Testo 32 caratteri	Esempio: a0b1c2d3e4f....

### **Nota4** subscription\_id

*Subscription\_id* identifica il tipo di abbonamento (con durata) che si vuole creare. I codici vi verranno forniti in fase di registrazione dopo che avrete deciso quali tipi di abbonamento vorrete offrire ai vostri clienti.

*Ci sono dei tipi di abbonamento standard, ma in caso di necessità è possibile crearne di nuovi appositamente per voi.*

*Ovviamente al variare delle vostre esigenze potrete chiederci di variare i tipi di abbonamento a vostra disposizione.*

*Con "rinnovo di un abbonamento" si intende prolungare opportunamente la durata di quello attualmente attivo e si realizza effettuando una chiamata con lo stesso *subscription\_id* dell'abbonamento attuale.*

*Nel caso in cui, mentre l'utente ha già un abbonamento attivo, si usa questa funzione ma con un *subscription\_id* diverso da quello attualmente in uso si creerà un nuovo abbonamento che va sostituire quello attuale. Il vecchio abbonamento si interrompe (ed i giorni residui vanno persi) e se ne attiva uno nuovo a partire dal giorno corrente.*



*Gli abbonamenti acquistati, come le ricariche, hanno una scadenza di utilizzo. In genere 1 anno. Non si possono attivare/rinnovare abbonamenti con data di inizio successiva alla data di scadenza degli abbonamenti acquistati dal partner.*

#### **Nota5 S2S\_check\_code**

*Nel caso in cui in fase di registrazione abbiate abilitato la verifica S2S delle richieste di attivazione di un abbonamento questo campo contiene in codice di attivazione che deve essere sottoposto a verifica. Sarà poi compito nostro effettuare una chiamata http verso il vostro server e ricevere l'autorizzazione. Vedete capitolo "S2S per verifica attivazione abbonamento" per ulteriori chiarimenti.*

#### **Nota6 MAC**

*La sicurezza e l'autenticità delle comunicazioni tra client (applicazione desktop o web) e server è garantita dal MAC, una stringa opportunamente costruita che firma la chiamata http. Per maggiori informazioni su come costruire la stringa MAC vedete il capitolo "generazione del MAC".*

### **Esempio di chiamata https alla funzione di attivazione abbonamento**

Fare attenzione, i valori dei parametri devono essere codificati secondo l'[RFC 1738](#) (@=%40, spazio=%20...). Non bisogna codificare le "&" che separano le singole coppie chiave=valore.

```
https://www.promemo.biz/api.php?function=subscription&user_id=userapi5%40PARTNER.promemo.biz&subscription_id=4&S2S_check_code=aa&mac=89c5b44ded98ecac38a1ba6928e18b91
```

### **Risposta per la funzione di attivazione abbonamento**

Le risposte sono in formato XML.

Per la funzione di attivazione abbonamento la risposta potrà essere formata dai seguenti parametri:

```
<state>TEXT</state>
```

Indica se l'operazione richiesta è andata a buon fine. Il valore potrà essere OK (successo) oppure KO (fallimento)

```
<error_number>NUMBER</error_number>
```

Indica il codice univoco dell'errore che si è verificato. Il valore è un numero.

```
<message>TEXT</message>
```

Descrive con una frase la riuscita dell'operazione o l'eventuale errore che si è verificato.

```
<credits_left>NUMBER</credits_left>
```

Numero di crediti residui disponibili al momento della risposta

```
<expiry_date>DATE-TIME</expiry_date>
```

Data di scadenza dell'abbonamento

### **Esempio di risposta positiva**

```
<xml>
  <state>OK</state>
  <message>subscription done</message>
  <credits_left>115</credits_left>
  <expiry_date>2009-11-34 23:34:30</expiry_date>
</xml>
```

### **Esempio di risposta negativa**

```
<xml>
  <state>KO</state>
```

```
<error_number>201</error_number>  
<message>wrong user_id</message>  
</xml>
```

## **Possibili codici di errore e messaggi di errore**

0001 = The requested function doesn't exist  
0002 = Invalid user\_id format  
0003 = User\_id inactive  
0004 = The specified partner doesn't exists or is inactive  
0005 = Invaild MAC  
0006 = Missing required field (*seguono altre info sul campo mancante*)  
0007 = Wrong field format (*seguono altre info sul campo errato*)

0201 = Invalid subscription\_id  
0202 = No more subscriptions available for the partner  
0203 = Invalid S2S answer  
0204 = Unexpected error (*seguono altre info sull'errore*)  
0205 = Can't add more subscriptions to this user

## Ricarica crediti

*NB. Funzione disponibile anche nel pannello di amministrazione Partner e che è necessario implementare via API solo per rendere il processo completamente automatico.*

Questa operazione permette di ricaricare il credito di un vostro cliente.

Il cliente è identificato tramite la **user\_id** comunicata in fase di registrazione ed il tipo di ricarica da attivare è definito tramite la **topup\_id**. Le **topup\_id** utilizzabili vi saranno comunicate in fase di registrazione ed identifica univocamente il numero di crediti (messaggi che si possono inviare) accreditati sull'account dell'utente.

Per aumentare la sicurezza e prevenire la ricarica a vostra insaputa è possibile utilizzare una verifica server-to-server tra i nostri sistemi ed un vostro server prima di procedere all'effettiva ricarica.

In ogni caso riceverete una mail (all'indirizzo specificato in **partner\_email**) come notifica del tentativo di creazione di una ricarica e l'esito (ricarica creata senza verifica S2S, ricarica creata con verifica S2S, ricarica non creata verifica S2S fallita).

Nella tabella 1.3 mostriamo l'elenco dei parametri per la funzione di ricarica crediti.

TABELLA 1.3 - Parametri per la funzione di ricarica crediti

Nome Parametro	Descrizione	Obb?	Formato – Lunghezza	Valore / Esempio
<b>function</b>	Identifica la funzione che si vuole realizzare	SI	Testo – non specificata	Valore: topup_credits
<b>user_id</b>	Identificativo univoco dell'utente.	SI	Email, max 200 caratteri	Esempio: <a href="mailto:6282@nomepartner.promemo.biz">6282@nomepartner.promemo.biz</a>
<b>topup_id</b>	Tipo di ricarica da associare al vostro utente ( <i>nota7</i> )	SI	Intero	Esempio: 2
<b>S2S_check_code</b>	Codice di attivazione che deve essere sottoposto a verifica S2S ( <i>nota5</i> )	SI se attiva la verifica a S2S per il partner	Testo, non specificato	Esempio: SS7J39SY3JS9JS9
<b>mac</b>	Stringa opportunamente costruita ed utilizzata per verificare l'autenticità della funzione richiesta ( <i>nota6</i> )	SI	Testo 32 caratteri	Esempio: a0b1c2d3e4f...

### **Nota7** topup\_id

*Topup\_id* identifica il tipo di ricarica che si vuole creare.

*I codici vi verranno forniti in fase di registrazione dopo che avrete deciso quali tipi di ricarica vorrete offrire ai vostri clienti. Ci sono dei tipi di ricarica standard, ma in caso di necessità è possibile crearne di nuovi appositamente per voi.*

*Ovviamente al variare delle vostre esigenze potrete chiederci di variare i tipi di ricarica a vostra disposizione.*

### **Esempio di chiamata https alla funzione di ricarica crediti**

Fare attenzione, i valori dei parametri devono essere codificati secondo l'[RFC 1738](#) (@=%40, spazio=%20...). Non bisogna codificare le "&" che separano le singole coppie chiave=valore.

`https://www.promemo.biz/api.php?function=topup_credits&user_id=userapi5%40PARTNER.promemo.biz&topup_id=1&S2S_check_code=aa&mac=1432bcaa968b4d69777b1d3223dc8aa9`

### **Risposta per la funzione di ricarica crediti**

Le risposte sono in formato XML.

Nel caso di ricarica crediti la risposta potrà essere formata dai seguenti parametri:

```
<state>STATE</state>
```

Indica se l'operazione richiesta è andata a buon fine. Il valore potrà essere OK (successo) oppure KO (fallimento)

```
<error_number>NUMBER</error_number>
```

Indica il codice univoco dell'errore che si è verificato. Il valore è un numero.

```
<message>MESSAGE</message>
```

Descrive con una frase la riuscita dell'operazione o l'eventuale errore che si è verificato.

```
<credits_left>NUMBER</credits_left>
```

Numero di crediti residui disponibili al momento della risposta

```
<expiry_date>DATE-TIME</expiry_date>
```

Data di scadenza dell'abbonamento

### Esempio di risposta positiva

```
<xml>  
  <state>OK</state>  
  <message>subscription done</message>  
  <credits_left>115</credits_left>  
  <expiry_date>2009-11-34 23:34:30</expiry_date>  
</xml>
```

### Esempio di risposta negativa

```
<xml>  
  <state>KO</state>  
  <error_number>201</error_number>  
  <message>wrong user_id</message>  
</xml>
```

### Possibili codici di errore e messaggi di errore

0001 = The requested function doesn't exist  
0002 = Invalid user\_id format  
0003 = User\_id inactive  
0004 = The specified partner doesn't exists or is inactive  
0005 = Invaild MAC  
0006 = Missing required field (*seguono altre info sul campo mancante*)  
0007 = Wrong field format (*seguono altre info sul campo errato*)  
  
0301 = Invalid topup\_id  
0302 = No more top ups available for the partner  
0303 = Invalid S2S answer  
0304 = Unexpected error (*seguono altre info sull'errore*)

## Imposta messaggio

Questa operazione permette di impostare un messaggio da inviare via SMS e/o E-Mail. Compilando tutti gli opportuni parametri è anche possibile impostare l'invio di messaggi ricorrenti.

Nella tabella 1.4 mostriamo l'elenco dei parametri per la funzione di imposta messaggio

TABELLA 1.4 - Parametri per la funzione di imposta messaggio

Nome Parametro	Descrizione	Obb?	Formato – Lunghezza	Valore / Esempio
<b>function</b>	Identifica la funzione che si vuole realizzare	SI	Testo – non specificata	Valore: save_message
<b>user_id</b>	Identificativo univoco dell'utente.	SI	Email, max 200 caratteri	Esempio: 6282@nomepartner.promemo.biz
<b>message_body</b>	Testo del messaggio da inviare	SI	Testo, max 146-160 caratteri del charset GSM 03.38 ( <i>vedi nota8</i> )	Esempio: Gentile Mario Rossi, le ricordo l'appuntamento per domani alle 12.00
<b>receiver_number</b>	Cellulare del destinatario	+ ( <i>nota9</i> )	Numero con o senza prefisso internazionale, senza spazi o altri segni	Esempio: 3391234567 oppure 00393391234567
<b>receiver_email</b>	E-Mail del destinatario	+ ( <i>nota9</i> )	Email - non specificato	Esempio: mario@rossi.com
<b>date_time_send</b>	Data e ora di invio del messaggio	NO ( <i>nota10</i> )	Date_time in formato standard (YYYY-MM-DD HH:II:SS)	Esempio: 2009-11-29 18:30:00
<b>recurring</b>	Frequenza della ricorrenza	# ( <i>nota11</i> )	Intero	Esempio: 3
<b>recurring_interval</b>	Unità di misura della ricorrenza	# ( <i>nota11</i> )	uno dei seguenti valori: YEAR, MONTH, WEEK, DAY	Esempio: MONTH
<b>recurring_expiry_date</b>	Data scadenza della ricorrenza	# ( <i>nota11</i> )	Date in formato standard (YYYY-MM-DD)	Esempio: 2009-11-29
<b>mac</b>	Stringa opportunamente costruita ed utilizzata per verificare l'autenticità della funzione richiesta ( <i>nota6</i> )	SI	Testo 32 caratteri	Esempio: a0b1c2d3e4f....

### **Nota8** lunghezza messaggio

In base al tipo di accordo sottoscritto si avrà la possibilità di impostare messaggi lunghi 160 caratteri (Partner Gold) oppure di 146 caratteri (Partner Silver).

Il charset GSM 03.38 indica quali caratteri vengono certamente visualizzati correttamente in tutti i dispositivi. Una griglia è consultabile sul sito [http://en.wikipedia.org/wiki/GSM\\_03.38](http://en.wikipedia.org/wiki/GSM_03.38).

In breve fanno parte del charset tutti i caratteri dell'alfabeto latino maiuscolo e minuscolo (comprese le lettere accentate), i numeri e i segni di punteggiatura.

Casi particolari sono rappresentati dai caratteri ^, {, }, \, [, ], ~, |, € che sono solo una estensione a del charset GSM 03.38 e che quindi valgono "doppio" (vengono contati come 2 caratteri).

I messaggi che contengono caratteri "doppi" e che per questo eccedono la lunghezza di 160 caratteri produrranno l'errore 402.

*I messaggi che contengono caratteri che non sono nel charset GSM 03.38 non produrranno nessun errore. I caratteri non accettati verranno eliminati ed il messaggio verrà salvato senza di questi.*

**Nota9** *receiver\_number e receiver\_email*

*+ è obbligatorio che sia presente almeno uno dei 2 valori (receiver\_email, receiver\_number).*

*Se verranno inseriti entrambi verranno inviati sia la E-mail che l'SMS.*

*Se vengono omessi entrambi verrà riportato un errore e il messaggio non verrà impostato.*

**Nota10** *date\_time\_send*

*È possibile impostare messaggi che verranno inviati nel futuro.*

*Se il parametro date\_time\_send viene omesso il messaggio verrà inviato subito.*

*Se date\_time\_send è antecedente di oltre 24 ore rispetto al momento della comunicazione http il messaggio non verrà inviato e verrà segnalato come "scaduto".*

*È quindi importante assicurare una tempestiva sincronizzazione tra l'applicazione desktop e il nostro server.*

**Nota11** *recurring, recurring\_interval e recurring\_expiry\_date*

*# il messaggio ricorrente verrà impostato solo se tutti e tre questi valori vengono forniti correttamente.*

*Un messaggio impostato con date\_time\_send=2009-10-29 18:30:00, recurring=2, recurring\_interval=WEEK, recurring\_expiry\_date=2010-05-12 verrà inviato, a partire dal 2009-10-29, ogni 2 settimane fine al 2010-05-12 compreso.*

*Se vengono inseriti solo 1 o 2 valori per questi 3 parametri, i valori immessi verranno ignorati e il messaggio verrà salvato come un messaggio non ricorrente.*

*Se si desidera impostare un messaggio ricorrente si consiglia di implementare all'interno dell'applicazione un controllo della presenza di tutti e tre i parametri.*

**ATTENZIONE:** nell'area di test i messaggi via E-mail vengono inviati realmente. Per i messaggi via SMS l'invio è simulato, ma è possibile eseguire tutte le altre funzioni normalmente.

**Esempio di chiamata https alla funzione di imposta messaggio**

Fare attenzione, i valori dei parametri devono essere codificati secondo l'[RFC 1738](#) (@=%40, spazio=%20...). Non bisogna codificare le "&" che separano le singole coppie chiave=valore.

```
https://www.promemo.biz/api.php?function=save_message&user_id=userapi5%40PARTNER
.promemo.biz&message_body=test%20invio%20da%20Adadadsadadsadsa%20sdsa%20a%20PI33
&receiver_number=23213213213&receiver_email=&date_time_send=2009-09-
30%2013%3A30%3A30&recurring=2&recurring_interval=DAY&recurring_expiry_date=2009-
11-30&mac=918c683925a86bbf2078168814343558
```

**Risposta per la funzione di imposta messaggio**

Le risposte sono in formato XML.

Nel caso di imposta messaggio la risposta potrà essere formata dai seguenti parametri:

`<state>STATE</state>`

Indica se l'operazione richiesta è andata a buon fine. Il valore potrà essere OK (successo) oppure KO (fallimento)

`<error_number>NUMBER</error_number>`

Indica il codice univoco dell'errore che si è verificato. Il valore è un numero.

`<message>MESSAGE</message>`

Descrive con una frase la riuscita dell'operazione o l'eventuale errore che si è verificato.

`<message_id>NUMBER</message_id>`

Conterrà l'identificativo del messaggio appena impostato e verrà utilizzato in seguito per eventualmente modificare o cancellare il messaggio impostato e per avere informazioni sul rapporto di invio del messaggio.

```
<credit_alert>BOOLEAN</credit_alert>
```

Il parametro credit\_alert (se == 1) indica se il credito residuo non è sufficiente ad inviare tutti i messaggi impostati fino alla scadenza dell'abbonamento corrente. Messaggi impostati in date successive non vengono coinvolti in questo avviso.

### Esempio di risposta positiva

```
<xml>
  <state>OK</state>
  <message>message saved</message>
  <message_id>12345</message_id>
  <credit_alert>1</credit_alert>
</xml>
```

### Esempio di risposta negativa

```
<xml>
  <state>KO</state>
  <error_number>401</error_number>
  <message>wrong user_id</message>
</xml>
```

### Possibili codici di errore e messaggi di errore

0001 = The requested function doesn't exist

0002 = Invalid user\_id format

0003 = User\_id inactive

0004 = The specified partner doesn't exists or is inactive

0005 = Invaild MAC

0006 = Missing required field (*seguono altre info sul campo mancante*)

0007 = Wrong field format (*seguono altre info sul campo errato*)

0401 = An error occorre

0402 = Message too long (remember GSM double char) >>> (vedi nota8)

## Modifica messaggio

Questa operazione permette di modificare un messaggio precedentemente impostato. I parametri richiesti sono del tutto simili a quello necessari in imposta messaggio, con l'aggiunto di **user\_id** che permette di identificare il messaggio che si vuole modificare.

Nella tabella 1.5 mostriamo l'elenco dei parametri per la funzione di modifica messaggio

TABELLA 1.5 - Parametri per la funzione di modifica messaggio

Nome Parametro	Descrizione	Obb?	Formato - Lunghezza	Valore / Esempio
<b>function</b>	Identifica la funzione che si vuole realizzare	SI	Testo - non specificata	Valore: modify_message
<b>user_id</b>	Identificativo univoco dell'utente.	SI	Email, max 200 caratteri	Esempio: 6282@nomepartner.promemo.biz
<b>message_id</b>	Identificativo del messaggio che si vuole modificare	SI	Intero	Esempio: 12
<b>message_body</b>	Testo del messaggio da inviare	SI	Testo, max 146-160 caratteri del charset GSM 03.38 (vedi nota8)	Esempio: Gentile Mario Rossi, le ricordo l'appuntamento per domani alle 12.00
<b>receiver_number</b>	Cellulare del destinatario	+ (nota9)	Numero con o senza prefisso internazionale, senza spazi o altri segni	Esempio: 3391234567 oppure 00393391234567
<b>receiver_email</b>	E-Mail del destinatario	+ (nota9)	Email - non specificato	Esempio: mario@rossi.com
<b>date_time_send</b>	Data e ora di invio del messaggio	NO (nota10)	Date_time in formato standard (YYYY-MM-DD HH:II:SS)	Esempio: 2009-11-29 18:30:00
<b>recurring</b>	Frequenza della ricorrenza	# (nota11)	Intero	Esempio: 3
<b>recurring_interval</b>	Unità di misura della ricorrenza	# (nota11)	uno dei seguenti valori: YEAR, MONTH, WEEK, DAY	Esempio: MONTH
<b>recurring_expiry_date</b>	Data scadenza della ricorrenza	# (nota11)	Date in formato standard (YYYY-MM-DD)	Esempio: 2009-11-29
<b>mac</b>	Stringa opportunamente costruita ed utilizzata per verificare l'autenticità della funzione richiesta (nota6)	SI	Testo 32 caratteri	Esempio: a0b1c2d3e4f...

### Esempio di chiamata https alla funzione di modifica messaggio

Fare attenzione, i valori dei parametri devono essere codificati l'RFC 1738 (@=%40, spazio=%20...). Non bisogna codificare le "&" che separano le singole coppie chiave=valore.

```
https://www.promemo.biz/api.php?function=modify_message&user_id=userapi5%40PARTNER.promemo.biz&message_id=1352&message_body=test%20invio%20da%20API%20MODIFICATO%2022&receiver_number=23213213213&receiver_email=&date_time_send=2009-09-30%2013%3A30%3A30&recurring=2&recurring_interval=DAY&recurring_expiry_date=2009-11-30&mac=a1e80b289691e9392bebc9452a42328e
```



## Risposta per la funzione di modifica messaggio

Le risposte sono in formato XML.

Nel caso di modifica messaggio la risposta potrà essere formata dai seguenti parametri:

`<state>STATE</state>`

Indica se l'operazione richiesta è andata a buon fine. Il valore potrà essere OK (successo) oppure KO (fallimento)

`<error_number>NUMBER</error_number>`

Indica il codice univoco dell'errore che si è verificato. Il valore è un numero.

`<message>MESSAGE</message>`

Descrive con una frase la riuscita dell'operazione o l'eventuale errore che si è verificato.

`<message_id>NUMBER</message_id>`

Conterrà l'identificativo del messaggio appena impostato e verrà utilizzato in seguito per eventualmente modificare o cancellare il messaggio impostato e per avere informazioni sul rapporto di invio del messaggio.

`<credit_alert>BOOLEAN</credit_alert>`

Il parametro `credit_alert` (se == 1) indica se il credito residuo non è sufficiente ad inviare tutti i messaggi impostati fino alla scadenza dell'abbonamento corrente. Messaggi impostati in date successive non vengono coinvolti in questo avviso.

### Esempio di risposta positiva

```
<xml>
  <state>OK</state>
  <message>message saved</message>
  <message_id>12345</message_id>
  <credit_alert>1</credit_alert>
</xml>
```

### Esempio di risposta negativa

```
<xml>
  <state>KO</state>
  <error_number>401</error_number>
  <message>wrong user_id</message>
</xml>
```

### Possibili codici di errore e messaggi di errore

0001 = The requested function doesn't exist  
0002 = Invalid user\_id format  
0003 = User\_id inactive  
0004 = The specified partner doesn't exists or is inactive  
0005 = Invaild MAC  
0006 = Missing required field (*seguono altre info sul campo mancante*)  
0007 = Wrong field format (*seguono altre info sul campo errato*)  
  
0501 = The requested message was not found  
0502 = An error occorre  
0503 = Message too long (remember GSM double char) >>> (vedi nota8)

## Cancella messaggio impostato

Questa operazione permette di cancellare un messaggio precedentemente impostato.

Nella tabella 1.6 mostriamo l'elenco dei parametri per la funzione di cancella messaggio impostato

TABELLA 1.6 - Parametri per la funzione di cancella messaggio impostato

Nome Parametro	Descrizione	Obb?	Formato - Lunghezza	Valore / Esempio
<b>function</b>	Identifica la funzione che si vuole realizzare	SI	Testo - non specificata	Valore: delete_message
<b>user_id</b>	Identificativo univoco dell'utente.	SI	Email, max 200 caratteri	Esempio: 6282@nomepartner.promemo.biz
<b>message_id</b>	Identificativo del messaggio che si vuole modificare	SI	Intero	Esempio: 12
<b>mac</b>	Stringa opportunamente costruita ed utilizzata per verificare l'autenticità della funzione richiesta ( <i>nota6</i> )	SI	Testo 32 caratteri	Esempio: a0b1c2d3e4f....

### Esempio di chiamata https alla funzione di cancella messaggio impostato

Fare attenzione, i valori dei parametri devono essere codificati secondo l'[RFC 1738](#) (@=%40, spazio=%20...). Non bisogna codificare le "&" che separano le singole coppie chiave=valore.

```
https://www.promemo.biz/api.php?function=delete_message&user_id=userapi5%40PARTNER.promemo.biz&message_id=1352&mac=4190e6a001b5ac86ea544cd0f585101c
```

### Risposta per la funzione di cancella messaggio impostato

Le risposte sono in formato XML.

Nel caso di cancella messaggio impostato la risposta potrà essere formata dai seguenti parametri:

```
<state>STATE</state>
```

Indica se l'operazione richiesta è andata a buon fine. Il valore potrà essere OK (successo) oppure KO (fallimento)

```
<error_number>NUMBER</error_number>
```

Indica il codice univoco dell'errore che si è verificato. Il valore è un numero.

```
<message>MESSAGE</message>
```

Descrive con una frase la riuscita dell'operazione o l'eventuale errore che si è verificato.

#### Esempio di risposta positiva

```
<xml>  
  <state>OK</state>  
  <message>message deleted</message>  
</xml>
```

#### Esempio di risposta negativa

```
<xml>  
  <state>KO</state>  
  <error_number>401</error_number>  
  <message>wrong user_id</message>  
</xml>
```

## **Possibili codici di errore e messaggi di errore**

0001 = The requested function doesn't exist

0002 = Invalid user\_id format

0003 = User\_id inactive

0004 = The specified partner doesn't exists or is inactive

0005 = Invaild MAC

0006 = Missing required field (*seguono altre info sul campo mancante*)

0007 = Wrong field format (*seguono altre info sul campo errato*)

0601 = The requested message was not found

0602 = An error occurred

## Stato abbonamento

Questa operazione permette di avere informazioni sullo stato dell'abbonamento dell'utente. Riceverete informazioni come la data di scadenza dell'abbonamento, i crediti residui e un alert che indica se il credito residuo è sufficiente ad inviare tutti i messaggi impostati fino alla scadenza dell'abbonamento.

Nella tabella 1.7 mostriamo l'elenco dei parametri per la funzione di stato abbonamento

TABELLA 1.7 - Parametri per la funzione di stato abbonamento

Nome Parametro	Descrizione	Obb?	Formato - Lunghezza	Valore / Esempio
<b>function</b>	Identifica la funzione che si vuole realizzare	SI	Testo - non specificata	Valore: get_subscription_info
<b>user_id</b>	Identificativo univoco dell'utente.	SI	Email, max 200 caratteri	Esempio: 6282@nomepartner.promemo.biz
<b>mac</b>	Stringa opportunamente costruita ed utilizzata per verificare l'autenticità della funzione richiesta ( <i>nota6</i> )	SI	Testo 32 caratteri	Esempio: a0b1c2d3e4f...

### Esempio di chiamata https alla funzione di stato abbonamento

Fare attenzione, i valori dei parametri devono essere codificati secondo l'[RFC 1738](#) (@=%40, spazio=%20...). Non bisogna codificare le "&" che separano le singole coppie chiave=valore.

[https://www.promemo.biz/api.php?function=get\\_subscription\\_info&user\\_id=userapi5%40PARTNER.promemo.biz&mac=8bd371cb34e30de4b0a0d960f955e58b](https://www.promemo.biz/api.php?function=get_subscription_info&user_id=userapi5%40PARTNER.promemo.biz&mac=8bd371cb34e30de4b0a0d960f955e58b)

### Risposta per la funzione di stato abbonamento

Le risposte sono in formato XML.

Nel caso di stato abbonamento la risposta potrà essere formata dai seguenti parametri:

`<state>STATE</state>`

Indica se l'operazione richiesta è andata a buon fine. Il valore potrà essere OK (successo) oppure KO (fallimento). Significa solo che non ci sono stati errori, non che è abbonato.

`<error_number>NUMBER</error_number>`

Indica il codice univoco dell'errore che si è verificato. Il valore è un numero.

`<message>MESSAGE</message>`

Descrive con una frase la riuscita dell'operazione o l'eventuale errore che si è verificato.

`<subscription_active>YES</subscription_active>`

Indica se l'ultimo abbonamento sottoscritto è ancora attivo. Può essere YES oppure NO

`<credits_left>NUMBER</credits_left>`

Numero di crediti residui disponibili al momento della risposta.

`<expiry_date>DATE-TIME</expiry_date>`

Data di scadenza dell'abbonamento.

`<credit_alert>BOOLEAN</credit_alert>`

Il parametro `credit_alert` (se == 1) indica se il credito residuo non è sufficiente ad inviare tutti i messaggi impostati. Conta solo i messaggi impostati con data prevista di invio fino alla scadenza dell'abbonamento.

## Esempio di risposta positiva

```
<xml>
  <state>OK</state>
  <subscription_active>YES</subscription_active>
  <expiry_date>YYYY-MM-DD HH:II:SS</expiry_date>
  <credits_left>NUMBER</credits_left>
  <credit_alert>NUMBER</credit_alert>
</xml>
```

## Esempio di risposta negativa

```
<xml>
  <state>KO</state>
  <error_number>401</error_number>
  <message>wrong user_id</message>
</xml>
```

## Possibili codici di errore e messaggi di errore

0001 = The requested function doesn't exist  
0002 = Invalid user\_id format  
0003 = User\_id inactive  
0004 = The specified partner doesn't exists or is inactive  
0005 = Invaild MAC  
0006 = Missing required field (*seguono altre info sul campo mancante*)  
0007 = Wrong field format (*seguono altre info sul campo errato*)

## Lista messaggi inviati

Questa operazione permette di ottenere la lista dei messaggi inviati dall'utente in un determinato intervallo di tempo.

È anche possibile effettuare una ricerca per E-mail destinatario o per cellulare destinatario per visualizzare esclusivamente lo storico dei messaggi inviati ad un determinato destinatario.

Nella tabella 1.8 mostriamo l'elenco dei parametri per la funzione di lista messaggi inviati

TABELLA 1.8 - Parametri per la funzione di lista messaggi inviati

Nome Parametro	Descrizione	Obb?	Formato - Lunghezza	Valore / Esempio
<b>function</b>	Identifica la funzione che si vuole realizzare	SI	Testo - non specificata	Valore: get_messages_history
<b>user_id</b>	Identificativo univoco dell'utente.	SI	Email, max 200 caratteri	Esempio: 6282@nomepartner.promemo.biz
<b>from_date</b>	Data di inizio dello storico richiesto ( <i>nota12</i> )	NO	YYYY-MM-DD HH:II:SS	Esempio: 2009-09-01 00:00:00
<b>to_date</b>	Data di fine dello storico richiesto ( <i>nota12</i> )	NO	YYYY-MM-DD HH:II:SS	Esempio: 2009-09-30 23:59:59
<b>how_many</b>	Quanti risultati restituire ( <i>nota13</i> )	NO	Intero (quanti risultati) <i>oppure</i> Intero partenza, Intero quanti risultati	Valore di default: 100 Esempio: 50,10 Esempio: 25
<b>search</b>	Restituisce solo i messaggi inviati al numero di cellulare o E-mail indicati	NO	Cellulare o E-mail	Esempio: 3391234567 Esempio: pippo@example.com
<b>status</b>	Indica se mostrare solo i messaggi in un particolare stato di invio	NO	uno dei seguenti valori: error, sent, not_received, received, expired	Esempio: not_received per vedere solo i messaggi non ricevuti
<b>mac</b>	Stringa opportunamente costruita ed utilizzata per verificare l'autenticità della funzione richiesta ( <i>nota6</i> )	SI	Testo 32 caratteri	Esempio: a0b1c2d3e4f...

### **Nota12** from\_date, to\_date

questi 2 parametri non sono obbligatori, e se omessi verranno restituiti gli ultimi "how\_many" messaggi inviati

### **Nota13** how\_many

il valore di default per how\_many è 100, quindi se ad esempio si omettono i campi how\_many, from\_date, to\_date verranno restituiti gli ultimi 100 messaggi inviati.

How\_many può essere nel formato "20, 10" per avere 10 risultati a partire dal 20-esimo. È così possibile dividere in diverse pagine i risultati.

Il numero massimo di risultati che può essere restituito per ogni chiamata è 1000.

### **Esempio di chiamata https alla funzione di lista messaggi inviati**

Fare attenzione, i valori dei parametri devono essere codificati secondo l'[RFC 1738](#) (@=%40, spazio=%20...). Non bisogna codificare le "&" che separano le singole coppie chiave=valore.

```
https://www.promemo.biz/api.php?function=get_messages_history&user_id=userapi5%40PARTNER.promemo.biz&from_date=&to_date=&how_many=&search=&mac=f2202d38b0246571086e87304df145ac
```

### **Risposta per la funzione di lista messaggi inviati**

Le risposte sono in formato XML.

Nel caso di lista messaggi inviati la risposta potrà essere formata dai seguenti parametri:

`<state>STATE</state>`

Indica se l'operazione richiesta è andata a buon fine. Il valore potrà essere OK (successo) oppure KO (fallimento). Significa solo che non ci sono stati errori, non che è abbonato.

`<error_number>NUMBER</error_number>`

Indica il codice univoco dell'errore che si è verificato. Il valore è un numero.

`<message>MESSAGE</message>`

Descrive con una frase la riuscita dell'operazione o l'eventuale errore che si è verificato.

`<total_number_messages>NUMBER</total_number_messages>`

Numero totale di messaggi inviati nel periodo scelto (from, to).

Se from\_date e to\_date non sono stati specificati indica il numero totale di messaggi inviati per l'utente. Questo valore può essere utile per stabilire se dividere i risultati in più pagine o no.

`<messages_list>`

Tag che racchiude (l'array de)i dettagli dei messaggi inviati.

`<message_details>`

Tag che racchiude i dettagli del singolo messaggio inviato. All'interno di message\_list ci possono essere più tag message\_details.

All'interno del tag message\_details troviamo i seguenti tag:

`<message_id>NUMBER</message_id>`

Indica l'identificativo del messaggio.

`<message_header>TEXT</message_header>`

L'intestazione del messaggio inviato.

`<message_body>TEXT</message_body>`

Contiene il testo del messaggio.

`<receiver_number>TEXT</receiver_number>`

Numero di telefono al quale è stato inviato il messaggio.

`<receiver_email>TEXT</receiver_email>`

E-Mail alla quale è stato inviato il messaggio.

`<recurring>BOOL</recurring>`

Indica se il messaggio inviato è di tipo ricorrente o no.

`<date_time_send>YYYY-MM-DD HH:II:SS</date_time_send>`

Indica la data e l'ora di effettivo invio del messaggio.

`<status>TEXT</status>`

Indica il rapporto di invio del messaggio. Può essere Inviato (messaggio inviato ma non ancora ricevuto); Ricevuto (messaggio effettivamente ricevuto); Non Ricevuto (messaggio inviato ma non è stato ricevuto dal terminale perché spento o non raggiungibile, dopo 24 ore non vengono più fatti tentativi di invio); Scaduto (messaggio non inviato perché l'utente non è abbonato oppure ha credito insufficiente); Errore (messaggio non inviato per errore interno al sistema o perché contiene caratteri non ammessi).

Per l'elenco dei caratteri ammessi fare riferimento alla nota8 e consultare il sito [http://en.wikipedia.org/wiki/GSM\\_03.38](http://en.wikipedia.org/wiki/GSM_03.38).

</message\_details> e </messages\_list>

Tag di chiusura dell'elenco dei messaggi e dei relativi dettagli.

### Esempio di risposta positiva

```
<xml>
  <state>OK</state>
  <total_number_messages>221</total_number_messages>
  <messages_list>
    <message_details>
      <message_id>2</message_id>
      <message_body>bla bla bla...</message_body>
      <receiver_number>3398237645</receiver_number>
      <receiver_email>pinco@pallo.com</receiver_email>
      <recurring>0</recurring>
      <date_time_send>2009-11-21 15:00:00</date_time_send>
      <status>Ricevuto</status>
    </message_details>
    <message_details>
      <message_id>4</message_id>
      <message_body>bla bla bla...</message_body>
      <receiver_number></receiver_number>
      <receiver_email>pallo@pinco.it</receiver_email>
      <recurring>1</recurring>
      <date_time_send>2009-10-30 12:15:00</date_time_send>
      <status>Scaduto</status>
    </message_details>
    <message_details>
      <message_id>12</message_id>
      <message_body>bla bla bla </message_body>
      <receiver_number>3341234567</receiver_number>
      <receiver_email></receiver_email>
      <recurring>1</recurring>
      <date_time_send>2009-12-25 12:00:00</date_time_send>
      <status>Inviato</status>
    </message_details>
  </messages_list>
</xml>
```

### Esempio di risposta negativa

```
<xml>
  <state>KO</state>
  <error_number>0007</error_number>
  <message>Wrong field format. From_date is in wrong format.</message>
</xml>
```

### Possibili codici di errore e messaggi di errore

0001 = The requested function doesn't exist  
0002 = Invalid user\_id format  
0003 = User\_id inactive  
0004 = The specified partner doesn't exists or is inactive  
0005 = Invaild MAC  
0006 = Missing required field (*seguono altre info sul campo mancante*)  
0007 = Wrong field format (*seguono altre info sul campo errato*)



## Lista messaggi salvati

Questa operazione permette di ottenere la lista dei messaggi salvati e non ancora inviati dall'utente in un determinato intervallo di tempo.

Se non si specifica un intervallo di tempo verranno mostrati gli ultimi messaggi salvati.

È anche possibile effettuare una ricerca per E-mail destinatario o per cellulare destinatario per visualizzare lo storico dei messaggi salvati da quel cliente.

Il significato dei parametri ed il loro uso è identico a quelli della funzione "lista messaggi inviati".

Nella tabella 1.9 mostriamo l'elenco dei parametri per la funzione di lista messaggi salvati

TABELLA 1.9 - Parametri per la funzione di lista messaggi salvati

Nome Parametro	Descrizione	Obb?	Formato - Lunghezza	Valore / Esempio
<b>function</b>	Identifica la funzione che si vuole realizzare	SI	Testo - non specificata	Valore: get_saved_messages
<b>user_id</b>	Identificativo univoco dell'utente.	SI	Email, max 200 caratteri	Esempio: 6282@nomepartner.promemo.biz
<b>from_date</b>	Data di inizio dello storico richiesto (nota12)	NO	YYYY-MM-DD HH:II:SS	Esempio: 2009-09-01 00:00:00
<b>to_date</b>	Data di fine dello storico richiesto (nota12)	NO	YYYY-MM-DD HH:II:SS	Esempio: 2009-09-30 23:59:59
<b>how_many</b>	Quanti risultati restituire (nota13)	NO	Intero (quanti risultati) oppure Intero partenza, Intero quanti risultati	Valore di default: 100 Esempio: 50,10 Esempio: 25
<b>search</b>	Restituisce solo i messaggi inviati al numero di cellulare o E-mail indicati	NO	Cellulare o E-mail	Esempio: 3391234567 Esempio: pippo@example.com
<b>mac</b>	Stringa opportunamente costruita ed utilizzata per verificare l'autenticità della funzione richiesta (nota6)	SI	Testo 32 caratteri	Esempio: a0b1c2d3e4f...

### Esempio di chiamata https alla funzione di lista messaggi salvati

Fare attenzione, i valori dei parametri devono essere codificati secondo l'[RFC 1738](#) (@=%40, spazio=%20...). Non bisogna codificare le "&" che separano le singole coppie chiave=valore.

```
https://www.promemo.biz/api.php?function=get_saved_messages&user_id=userapi5%40P
ARTNER.promemo.biz&from_date=&to_date=&how_many=&search=&mac=2c89ce5edaaea39153a
cb26f1f17e469
```

### Risposta per la funzione di lista messaggi salvati

Le risposte sono in formato XML.

Nel caso di lista messaggi salvati la risposta potrà essere formata dai seguenti parametri:

```
<state>STATE</state>
```

Indica se l'operazione richiesta è andata a buon fine. Il valore potrà essere OK (successo) oppure KO (fallimento). Significa solo che non ci sono stati errori, non che è abbonato.

```
<error_number>NUMBER</error_number>
```

Indica il codice univoco dell'errore che si è verificato. Il valore è un numero.

```
<message>MESSAGE</message>
```

Descrive con una frase la riuscita dell'operazione o l'eventuale errore che si è verificato.

```
<total_number_messages>NUMBER</total_number_messages>
```

Numero totale di messaggi inviati nel periodo scelto (from, to).  
Se from\_date e to\_date non sono stati specificati indica il numero totale di messaggi inviati per l'utente. Questo valore può essere utile per stabilire se dividere i risultati in più pagine o no.

```
<messages_list>
```

Tag che racchiude (l'array de)i dettagli dei messaggi inviati.

```
<message_details>
```

Tag che racchiude i dettagli del singolo messaggio inviato. All'interno di message\_list ci possono essere più tag message\_details.

All'interno del tag message\_details troviamo i seguenti tag:

```
<message_id>NUMBER</message_id>
```

Indica l'identificativo del messaggio.

```
<message_body>TEXT</message_body>
```

Contiene il testo del messaggio.

```
<receiver_number>TEXT</receiver_number>
```

Numero di telefono al quale è stato inviato il messaggio.

```
<receiver_email>TEXT</receiver_email>
```

E-Mail alla quale è stato inviato il messaggio.

```
<recurring>BOOL</recurring>
```

Indica se il messaggio inviato è di tipo ricorrente o no.

```
<date_time_send>YYYY-MM-DD HH:II:SS</date_time_send>
```

Indica la data e l'ora di effettivo invio del messaggio.

```
</message_details> e </messages_list>
```

Tag di chiusura dell'elenco dei messaggi e dei relativi dettagli.

## Esempio di risposta positiva

```
<xml>
  <state>OK</state>
  <total_number_messages>221</total_number_messages>
  <messages_list>
    <message_details>
      <message_id>2</message_id>
      <message_body>bla bla bla...</message_body>
      <receiver_number>3398237645</receiver_number>
      <receiver_email>pinco@pallo.com</receiver_email>
      <recurring>0</recurring>
      <date_time_send>2009-11-21 15:00:00</date_time_send>
    </message_details>
    <message_details>
      <message_id>4</message_id>
      <message_body>bla bla bla...</message_body>
      <receiver_number></receiver_number>
      <receiver_email>pallo@pinco.it</receiver_email>
      <recurring>1</recurring>
      <date_time_send>2009-10-30 12:15:00</date_time_send>
    </message_details>
    <message_details>
      <message_id>12</message_id>
      <message_body>bla bla bla </message_body>
      <receiver_number>3341234567</receiver_number>

```

```
<receiver_email></receiver_email>
<recurring>1</recurring>
<date_time_send>2009-12-25 12:00:00</date_time_send>
</message_details>
</messages_list>
</xml>
```

## Esempio di risposta negativa

```
<xml>
  <state>KO</state>
  <error_number>0007</error_number>
  <message>Wrong field format. From_date is in wrong format.</message>
</xml>
```

## Possibili codici di errore e messaggi di errore

0001 = The requested function doesn't exist

0002 = Invalid user\_id format

0003 = User\_id inactive

0004 = The specified partner doesn't exists or is inactive

0005 = Invaield MAC

0006 = Missing required field *(seguono altre info sul campo mancante)*

0007 = Wrong field format *(seguono altre info sul campo errato)*

## Stato messaggio

Questa funzione permette di recuperare le informazioni relative ad uno specifico messaggio. Ad esempio richiedendo informazioni su un messaggio ricorrente vedrete quante volte è stato inviato, quali sono i parametri della ricorrenza e quando è previsto il prossimo invio, oltre a tutti gli altri dati relativi al messaggio.

Nella tabella 1.10 mostriamo l'elenco dei parametri per la funzione di stato messaggio

TABELLA 1.10 - Parametri per la funzione di stato messaggio

Nome Parametro	Descrizione	Obb?	Formato – Lunghezza	Valore / Esempio
<b>function</b>	Identifica la funzione che si vuole realizzare	SI	Testo – non specificata	Valore: get_message_info
<b>user_id</b>	Identificativo univoco dell'utente.	SI	Email, max 200 caratteri	Esempio: 6282@nomepartner.promemo.biz
<b>message_id</b>	Identificativo del messaggio che si vuole modificare	SI	Intero	Esempio: 12
<b>mac</b>	Stringa opportunamente costruita ed utilizzata per verificare l'autenticità della funzione richiesta ( <i>nota6</i> )	SI	Testo 32 caratteri	Esempio: a0b1c2d3e4f...

### Esempio di chiamata https alla funzione di stato messaggio

Fare attenzione, i valori dei parametri devono essere codificati secondo l'[RFC 1738](#) (@=%40, spazio=%20...). Non bisogna codificare le "&" che separano le singole coppie chiave=valore.

[https://www.promemo.biz/api.php?function=get\\_message\\_info&user\\_id=userapi5%40PARTNER.promemo.biz&message\\_id=1354&mac=1f86c6c7c21bd3ba7451b323429ba064](https://www.promemo.biz/api.php?function=get_message_info&user_id=userapi5%40PARTNER.promemo.biz&message_id=1354&mac=1f86c6c7c21bd3ba7451b323429ba064)

### Risposta per la funzione di stato messaggio

Le risposte sono in formato XML.

Nel caso di stato messaggio la risposta potrà essere formata dai seguenti parametri:

`<state>STATE</state>`

Indica se l'operazione richiesta è andata a buon fine. Il valore potrà essere OK (successo) oppure KO (fallimento). Significa solo che non ci sono stati errori, non che è abbonato.

`<error_number>NUMBER</error_number>`

Indica il codice univoco dell'errore che si è verificato. Il valore è un numero.

`<message>MESSAGE</message>`

Descrive con una frase la riuscita dell'operazione o l'eventuale errore che si è verificato.

`<message_id>NUMBER</message_id>`

Indica l'identificativo del messaggio.

`<message_body>TEXT</message_body>`

Contiene il testo del messaggio.

`<receiver_number>TEXT</receiver_number>`

Numero di telefono al quale è stato inviato il messaggio.

`<receiver_email>TEXT</receiver_email>`

E-Mail alla quale è stato inviato il messaggio.

`<recurring>NUMBER</recurring>`

Parametri valorizzati se il messaggio è ricorrente. Indica la frequenza della ricorrenza.

`<recurring_interval>SET(YEAR, MONTH, WEEK, DAY)</recurring_interval>`

Parametri valorizzati se il messaggio è ricorrente. Indica l'unità di misura della ricorrenza.

`<recurring_expiry_date> YYYY-MM-DD HH:II:SS </recurring_expiry_date>`

Parametri valorizzati se il messaggio è ricorrente. Indica la data di scadenza della ricorrenza.

`<date_time_send>YYYY-MM-DD HH:II:SS</date_time_send>`

Indica la data dell'invio del messaggio. Ha lo stesso significato sia se è stato già inviato sia se verrà inviato in futuro.

`<delivery>`

Tag che racchiude l'insieme dei report relativi ai messaggi inviati, se il messaggio è ricorrente, o relativo al messaggio inviato, se messaggio singolo.

`<report>`

Tag che racchiude il rapporto sull'invio del messaggio.

All'interno del tag report troviamo i seguenti tag:

`<date_time_send>YYYY-MM-DD HH:II:SS</date_time_send>`

Indica la data in cui il messaggio è stato inviato.

`<status>TEXT</status>`

Indica il rapporto di invio del messaggio. Può essere Inviato (messaggio inviato ma non ancora ricevuto dal terminale); Ricevuto (messaggio effettivamente ricevuto dal terminale); Non Ricevuto (Il terminale risulta spento o non raggiungibile per oltre 24 ore); Scaduto (messaggio non inviato perché l'utente non è abbonato oppure ha credito insufficiente); Errore (messaggio non inviato per errore di connessione al server o perché contiene caratteri non accettati).

Per l'elenco dei caratteri accettati fare riferimento alla nota8 e consultare il sito [http://en.wikipedia.org/wiki/GSM\\_03.38](http://en.wikipedia.org/wiki/GSM_03.38).

## Esempio di risposta positiva

`<xml>`

`<state>OK</state>`

`<message_id>15</message_id>`

`<message_body>Testo del messaggio da inviare</message_body>`

`<receiver_number>3391234567</receiver_number>`

`<receiver_email>Mario@rossi.it</receiver_email>`

`<recurring>4</recurring>`

`<recurring_interval>DAY</recurring_interval>`

`<recurring_expiry_date>2009-12-31</recurring_expiry_date>`

`<date_time_send>2009-10-02</date_time_send>`

`<delivery>`

`<report>`

`<date_time_send>2009-09-28</date_time_send>`

`<status>Ricevuto</status>`

`</report>`

`<report>`

`<date_time_send>2009-09-24</date_time_send>`

`<status>Ricevuto</status>`

`</report>`

`<report>`

`<date_time_send>2009-09-20</date_time_send>`

`<status>Ricevuto</status>`

`</report>`

`</delivery>`

`</xml>`

## Esempio di risposta negativa

```
<xml>  
  <state>KO</state>  
  <error_number>1001</error_number>  
  <message> The requested message was not found </message>  
</xml>
```

## Possibili codici di errore e messaggi di errore

0001 = The requested function doesn't exist  
0002 = Invalid user\_id format  
0003 = User\_id inactive  
0004 = The specified partner doesn't exists or is inactive  
0005 = Invaild MAC  
0006 = Missing required field (*seguono altre info sul campo mancante*)  
0007 = Wrong field format (*seguono altre info sul campo errato*)  
  
1001 = The requested message was not found

## Aggiornamento informazioni account

Questa operazione permette di aggiornare i dati del vostro utente.

Nella tabella 1.11 mostriamo l'elenco dei parametri per la funzione di aggiornamento informazioni account

TABELLA 1.11 - Parametri per la funzione di aggiornamento informazioni account

Nome Parametro	Descrizione	Obb?	Formato - Lunghezza	Valore / Esempio
<b>function</b>	Identifica la funzione che si vuole realizzare	SI	Testo - non specificata	Valore: update_account_info
<b>user_id</b>	Identificativo univoco dell'utente. (nota1)	SI	Email (vedi nota1), max 200 caratteri	Esempio: 6282@nomepartner.promemo.biz
<b>name</b>	Nome dell'utente	SI	Testo: 2-200 caratteri	Esempio: Mario
<b>surname</b>	Cognome dell'utente	SI	Testo: 2-200 caratteri	Esempio: Rossi
<b>password</b>	Password utente. Nelle successive operazioni verrà utilizzata per generare i codici di autenticazione (nota2)	SI	Testo, 4-200 caratteri	Esempio: *****
<b>partner_email</b>	E-mail del partner alla quale inviare le notifiche relative a questo utente	SI	Email, max 200 caratteri	Esempio: abcde@example.com
<b>message_header</b>	Numero di telefono o testo che apparirà come mittente dei messaggi inviati	SI	Testo, max 11 caratteri	Esempio: 3381234567 oppure RossiS.p.A.
<b>company_name</b>	Ragione sociale azienda dell'utente	NO	Testo, 3-200 caratteri	Esempio: La magnifica Srl
<b>legal_address</b>	Indirizzo dell'utente	NO	Testo max 255 caratteri	Esempio: Via pinco pallo, 34
<b>zip_code</b>	Codice di avviamento postale (CAP) dell'utente	NO	Numerico, 5 caratteri	Esempio: 00100
<b>city</b>	Città dell'utente	NO	Testo 2-50 caratteri	Esempio: Roma
<b>province</b>	Provincia dell'utente	NO	2 caratteri testuali	Esempio: RM
<b>phone</b>	Telefono dell'utente	NO	Numeri 6-50 cifre	Esempio: 06123456
<b>category</b>	Categoria attività dell'utente	NO	Testo 2-200 caratteri	Esempio: Medici

### Esempio di chiamata https alla funzione di aggiornamento informazioni account

Fare attenzione, i valori dei parametri devono essere codificati secondo l'[RFC 1738](#) (@=%40, spazio=%20...). Non bisogna codificare le "&" che separano le singole coppie chiave=valore.

```
https://www.promemo.biz/api.php?function=update_account_info&user_id=userapi4%40PARTNER.promemo.biz&name=mario&surname=rossi&password=aaaa&partner_email=mario.rossi%40yahoo.it&message_header=header&company_name=Azienda%20S.p.a&legal_address=&zip_code=&city=&province=&phone=&mac=80acef65742dc440eb1b1850c471af23
```

### Risposta per la funzione di aggiornamento informazioni account

Le risposte sono in formato XML.

Nel caso di aggiornamento informazioni account la risposta potrà essere formata dai seguenti parametri:

`<state>TEXT</state>`

Indica se l'operazione richiesta è andata a buon fine. Il valore potrà essere OK (successo) oppure KO (fallimento)

`<error number>NUMBER</error number>`

Indica il codice univoco dell'errore che si è verificato. Il valore è un numero.

`<message>TEXT</message>`

Descrive con una frase l'errore che si è verificato.

Contestualmente alla risposta positiva via http riceverete una mail (all'indirizzo specificato nel parametro **partner\_email**) come notifica dell'avvenuta registrazione di un nuovo utente.

### Esempio di risposta positiva

```
<xml>
  <state>OK</state>
  <message>signup donne</message>
</xml>
```

### Esempio di risposta negativa

```
<xml>
  <state>KO</state>
  <error_number>1101</error_number>
  <message>client_email missed or with wrong format</message>
</xml>
```

### Possibili codici di errore e messaggi di errore

0001 = The requested function doesn't exist  
0003 = User\_id inactive  
0005 = Invalid MAC  
0006 = Missing required field (*seguono altre info sul campo mancante*)  
0007 = Wrong field format (*seguono altre info sul campo errato*)  
  
1101 = Invalid user\_id



## Generazione del MAC

Come preannunciato la sicurezza e l'autenticità delle comunicazioni tra client (applicazione desktop o web) e server è garantita dalla stringa di MAC, una stringa opportunamente costruita che firma la chiamata https.

Per generare il codice MAC, oltre alla stringa che si vuole inviare, viene usata anche la password, comunicata in fase di registrazione.

La generazione della stringa di MAC usa il seguente algoritmo:

MAC = md5(Stringa\_Richiesta + md5(PASSWORD))

STRINGA RICHIESTA = 'nome=xxxx&cognome=xxxxx&...'

STRINGA RICHIESTA CON MAC = STRINGA RICHIESTA + '&mac=' + MAC

NB. Il simbolo "+" indica la concatenazione tra 2 stringhe (in php sarebbe il punto ".") e non un carattere vero e proprio.

es: index.php?nome=xxxx&cognome=xxxxx&.....&mac=HGFDASJGDJAS

Nella STRINGA RICHIESTA devono essere presenti solo parametri richiesti dalla funzione che si sta utilizzando, ed ogni coppia "chiave=valore" deve essere separata da una & commerciale.

Es:

parametro1=abc

parametro2=cde

parametro3=xyz

STRINGA RICHIESTA: parametro1=abc&parametro2=cde&parametro3=xyz

## ERRORI COMUNI

- NELLA FORMAZIONE DELLA "STRINGA RICHIESTA"
  - & alla fine (parametro1=abc&parametro2=cde&parametro3=xyz&)
  - & ripetute (parametro1=abc&&parametro2=cde&parametro3=xyz)
  - nomi di campi non corretti (maiuscole/minuscole)
  - valore dei parametri non codificati correttamente (secondo l'RFC 1738)
  - chiavi non richieste dal metodo (es: passare subscription\_id durante la registrazione)
- NEL CALCOLO DEL MAC
  - concatenazione della password in chiaro, anziché del suo md5  
es ...&parametro3=xyz**PASS IN CHIARO**
  - aggiunta di una & prima della concatenazione  
es ...&parametro3=xyz&PASS\_MD5
  - aggiunta della chiave "password" non richiesta  
es ...&parametro3=xyz**&password=**PASS\_MD5
- NELLA FORMAZIONE DELLA "STRINGA RICHIESTA CON MAC"
  - Parametro mac in maiuscolo (...&parametro3=xyz&**MAC**=ab1c3d4e..)

## ***S2S per verifica attivazione abbonamento e ricarica crediti***

Per la massima sicurezza, vostra e dei vostri clienti, è possibile (e consigliato) abilitare la verifica Server-to-Server. Questa elimina la possibilità di erronea (o fraudolenta) attivazione di abbonamenti o di ricarica crediti.

Il funzionamento è molto semplice:

durante la richiesta di attivazione di un abbonamento o di una ricarica crediti inviate insieme agli altri dati il parametro **S2S\_check\_code**. A questo punto PROMemo effettua una richiesta all'indirizzo prestabilito (es <http://www.vostrosito.com/checkAbb.php>) inviando il codice S2S\_check\_code unitamente alle informazioni sulla sottoscrizione o sulla ricarica richiesta.

A seconda della risposta ricevuta dal vostro server procederemo o meno con l'attivazione dell'abbonamento o della ricarica.

Normalmente quindi il S2S\_check\_code sarà un identificativo dell'abbonamento dal vostro punto di vista, come ad esempio un codice della licenza della vostra applicazione, che potrete verificare in qualsiasi momento.

La risposta S2S deve essere formata in questo modo:

In caso di attivazione abbonamento:

S2S\_check\_code-subscription\_id-MAC\_P

In caso di ricarica crediti

S2S\_check\_code-topup\_id-MAC\_P

Il MAC\_P si calcola facendo l'md5 della concatenazione tra la richiesta che PROMemo invia al vostro script di verifica con l'md5 della password del partner (stabilita in fase di registrazione partner)

Esempio

Arriva a PROMemo la richiesta da un utente di attivare un abbonamento. Il S2S\_check\_code è ABCDEF, il subscription\_id è 2. Supponiamo il partner abbia come url di verifica S2S <http://www.example.com/check.php> e come password partner "disney"

PROMemo effettua la richiesta al vostro server, quindi

[http://www.example.com/check.php?S2S\\_check\\_code=ABCDEF&subscription\\_id=2](http://www.example.com/check.php?S2S_check_code=ABCDEF&subscription_id=2)

Il vostro script verifica internamente la validità della richiesta e risponde affermativamente:

ABCDEF-2-772091d338fc94dee58c4361c0395b69

Il MAC\_P calcolato è infatti

`md5('S2S_check_code=ABCDEF&subscription_id=2'.md5('disney'))`

ovvero

`md5('S2S_check_code=ABCDEF&subscription_id=2f240a4a08ef4d49a9b643168779d8491')`

quindi

772091d338fc94dee58c4361c0395b69

In modo simile per le ricariche, solamente usando topup\_id al posto di subscription\_id

Qualsiasi altra risposta sarà ritenuta negativa e l'abbonamento/ricarica non verrà effettuato.

## ***Conclusioni***

Speriamo di essere stati esaurienti ed aver illustrato tutte le funzionalità messe a disposizione mediante chiari esempi.

Ad ogni modo restiamo a disposizione per eventuali feedback sulla presente documentazione, segnalazione di errori ortografici o di contenuto e per richiesta di supporto all'integrazione di PROmemo.biz nella vostra applicazione.

Per tutto questo non esitate a contattarci all'indirizzo [partner@promemo.biz](mailto:partner@promemo.biz).

Grazie e buon lavoro

## ***Appendice A: codici e messaggi di errore***

Per facilitare la consultazione riportiamo qui i messaggi di errore che si possono verificare durante l'utilizzo delle API.

### **Messaggi di errore generici che si possono verificare in tutte le funzioni:**

0001 = The requested function doesn't exist

0002 = Invalid user\_id format

0003 = User\_id inactive

0004 = The specified partner doesn't exists or is inactive

0005 = Invalid MAC

0006 = Missing required field (seguono altre info sul campo mancante)

0007 = Wrong field format (seguono altre info sul campo errato)

### **Messaggi di errore specifici per Registrazione utente:**

0101 = User already registered

0102 = Unexpected error (seguono altre info sull'errore)

### **Messaggi di errore specifici per Attivazione abbonamento:**

0201 = Invalid subscription\_id

0202 = No more subscriptions available for the partner

0203 = Invalid S2S answer

0204 = Unexpected error (seguono altre info sull'errore)

0205 = Can't add more subscriptions to this user

### **Messaggi di errore specifici per Ricarica crediti:**

0301 = Invalid topup\_id

0302 = No more top ups available for the partner

0303 = Invalid S2S answer

0304 = Unexpected error (seguono altre info sull'errore)

### **Messaggi di errore specifici per Imposta messaggio:**

0401 = An error occorre

0402 = Message too long (remember GSM double char)

### **Messaggi di errore specifici per Modifica messaggio:**

0501 = The requested message was not found

0502 = An error occurred

0503 = Message too long (remember GSM double char)

### **Messaggi di errore specifici per Cancella messaggio impostato:**

0601 = The requested message was not found

0602 = An error occurred

### **Messaggi di errore specifici per Stato messaggio:**

1001 = The requested message was not found

### **Messaggi di errore specifici per Aggiornamento informazioni account:**

1101 = Invalid user\_id