

Specifiche API

Provisioning Rivenditore

Indice

1	Introduzione Tecnica.....	2
2	Richiesta XML-RPC.....	2
3	Creazione nuovo account Cliente.....	2
4	Modifica account Cliente.....	3
5	Assegnamento rotte all'Account Cliente.....	3
6	Ricarica Account Cliente.....	3
7	Interrogazione credito residuo Cliente.....	4
8	Ricerca ID Account dal Login.....	4
9	Interrogazione elenco utenti.....	4
10	Codici di Errore.....	4
11	Esempi.....	5
11.1	Esempio PHP.....	5
11.2	Esempio in VB .NET.....	6

I Introduzione Tecnica

Le API sono accessibili tramite HTTPS utilizzando il protocollo XML-RPC. Ogni metodo accetta un solo parametro di tipo "struct" contenente l'elenco dei parametri accettati da ogni singola funzione specifica. Quindi per chiamare un qualsiasi metodo qui documentato vanno specificati i parametri di autenticazione e quelli specifici di ogni metodo.

L'encoding di base per l'accesso alla piattaforma è ISO-8859-1 (Latin I), tutti i caratteri vanno quindi indicati con questo charset e anche nell'eventuale encoding in base64 vanno usati caratteri Latin I.

L'autenticazione viene eseguita con i seguenti parametri obbligatori:

Nome Campo	Descrizione	XML Type
authlogin	Login di autenticazione alle API che nel livello di accesso corrente (BRP) corrisponde al login dell'Account Rivenditore	string
authpasswd	Password di autenticazione alle API che come sopra, corrisponde alla password dell'Account Rivenditore	string

In caso di errore fatale l'API ritorna un errore XML tramite un oggetto **Fault**, con codice di errore e descrizione del relativo codice di errore (vedi "Codici di Errore", paragrafo 9).

2 Richiesta XML-RPC

La procedura provisioning.py fornisce la possibilità di gestire gli account relativi ai propri clienti. Tutte le operazioni eseguite con questa procedura avvengono sul Database Operativo e quindi tutti i dati sono interrogati e variati in tempo reale.

L'URL completo della chiamata XML-RPC è:

<https://secure.apisms.it/xmlrpc/BRP/provisioning.py>

3 Creazione nuovo account Cliente

Metodo: create_user

Return: ID Account (long)

Descrizione: Crea un nuovo cliente e restituisce l'ID, il cliente viene creato con credito zero.

Nome Campo	Descrizione	XML Type
login	Login del cliente al quale verrà aggiunto il suffisso rivenditore campo da codificare in base64	string
password	Password per l'accesso al portale cliente campo da codificare in base64	string
ragione_sociale	Ragione sociale o Nome e Cognome, campo da codificare in base64	string
<i>cellulare</i>	<i>Cellulare</i>	<i>string</i>
<i>telefono</i>	<i>Telefono</i>	<i>string</i>
<i>email</i>	<i>E-mail</i>	<i>string</i>
<i>soglia_notifica_credito</i>	<i>Soglia di notifica del credito in esaurimento, , se valorizzato con 0 (zero) la funzione è disattiva.</i>	<i>double</i>
<i>scadenza_credito</i>	<i>Scadenza del credito in formato GG/MM/AAAA e maggiore della data odierna. Se non specificato viene impostato ad un anno dalla data di creazione dell'account.</i>	<i>string</i>
<i>note</i>	<i>Note, campo da codificare in base64</i>	<i>string</i>

I campi in corsivo sono opzionali

4 Modifica account Cliente

Metodo: modify_user

Return: in caso di successo ritorna come stringa "OK" altrimenti restituisce un oggetto Fault

Descrizione: modifica i campi del cliente, vanno specificati nella richiesta solo i campi che si intende modificare. Quelli non specificati rimarranno invariati.

Nome Campo	Descrizione	XML Type
id_cliente	ID account del cliente	int
login	Login del cliente al quale verrà aggiunto il suffisso rivenditore campo da codificare in base64	string
password	Password per l'accesso al portale cliente campo da codificare in base64	string
ragione_sociale	Ragione sociale o Nome e Cognome campo da codificare in base64	string
cellulare	Cellulare	string
telefono	Telefono	string
email	E-mail	string
soglia_notifica_credito	Soglia di notifica del credito in esaurimento	double
scadenza_credito	Scadenza del credito in formato GG/MM/AAAA e maggiore della data odierna.	string
attivo	Se è 0 (zero) viene disattivato, se è diverso da zero viene attivato	integer
note	Note, campo da codificare in base64	string

I campi in corsivo sono opzionali

5 Assegnamento rotte all'Account Cliente

Metodo: set_route

Return: array dei listini che non sono stati accettati, nel caso di esito positivo per tutti gli ID API, viene ritornato un array vuoto.

Descrizione: assegna al cliente le rotte con cui poi potrà inviare SMS.

I parametri accettati da questo metodo sono:

Nome Campo	Descrizione	XML Type
id_cliente	ID account del cliente	int
rotte	Lista degli ID API dei listini cliente da associare, gli ID per essere validi devono essere di tipo Integer	array

6 Ricarica Account Cliente

Metodo: add_credit

Return: credito residuo dopo la ricarica (double)

Descrizione: esegue una ricarica di credito all'account cliente e ne restituisce il credito residuo dopo la ricarica.

I parametri accettati da questo metodo sono:

Nome Campo	Descrizione	XML Type
id_cliente	ID account del cliente	int
credito	Importo del credito per l'operazione di ricarica, può essere anche negativo	double
note	Note campo da codificare in base64	string

7 Interrogazione credito residuo Cliente

Metodo: `get_credit`

Return: il credito residuo del cliente

Descrizione: restituisce il credito residuo del cliente

I parametri accettati da questo metodo sono:

Nome Campo	Descrizione	XML Type
<code>id_cliente</code>	ID account del cliente	int

8 Ricerca ID Account dal Login

Metodo: `get_user_id`

Return: l'ID dell'account cliente (int)

Descrizione: restituisce l'ID del Login del cliente specificato.

I parametri accettati da questo metodo sono:

Nome Campo	Descrizione	XML Type
<code>login</code>	Login (comprensivo di suffisso rivenditore) del cliente di cui si sta cercando l'ID campo da codificare in base64	string

9 Interrogazione elenco utenti

Metodo: `get_users_list`

Return: array/struct (uno per ogni utente) composto da: ID, Login, RagioneSociale, Cellulare, Telefono, Email, NoteInterne, ScadenzaCredito, CreditoResiduo, Attivo (1 se attivo, 0 se non attivo)

I campi Login, RagioneSociale, Email, NoteInterne sono codificati in base64.

Descrizione: restituisce l'elenco degli utenti del rivenditore, nel caso non abbia rivenditori, la lista sarà vuota.

I parametri accettati da questo metodo sono: Nessun parametro

10 Codici di Errore

Tutti gli oggetti *Fault* restituiti dalle API XML-RPC descritte, sono elencati di seguito:

Error Code	Description	Causa
3	Access denied	La login/password per l'accesso non è corretta o il tipo di risorsa richiesta non è accessibile dall'account con cui si è tentato di eseguire l'accesso.
5	Parameter %s not specified	Il parametro fornito al metodo XML-RPC non è stato specificato
6	Date '%s' incorrect	La data specificata non è valida
12	Invalid id_cliente	Si è tentato di operare su un cliente non esistente o non si dispongono dei permessi necessari per l'operazione
13	%s invalid parameter type	Il tipo del parametro inserito non è corretto. (Es: il codice doveva essere Integer invece è stato passato come Stringa)
14	login already exists	Il login specificato è già esistente
15	login does not exist	Il login in specificato non esiste
30	method %s not supported	Il metodo XML-RPC chiamato non esiste
31	%s arguments expected	Il numero di argomenti non è conforme alla richiesta.
32	Internal server error, try again later	La richiesta inserita è stata annullata in quanto si sono verificati problemi tecnici, l'errore è tracciato e va verificato con il reparto tecnico.

33	Wrong xml request: %s	La richiesta xml non è valida, viene riportato il messaggio di errore del parser.
34	Malformed request	La richiesta al metodo non era composta da un'array associativo(struct).

II Esempi

II.1 Esempio PHP

La libreria su cui consigliamo di lavorare in PHP è scaricabile all'indirizzo: <http://phpxmlrpc.sourceforge.net/>
La versione su cui sono stati eseguiti i test è la 2.2.

E' disponibile un KIT che implementa tutte le API Rivenditore in PHP come esempi al seguente indirizzo:

https://secure.apisms.it/documentazione_api/XMLRPC_BRP_KIT.zip

L'esempio che segue prende in considerazione la creazione di un Account tramite la funzione "create_user".

```
<?
include("xmlrpc.inc");
$xmlrpc_client = new xmlrpc_client('https://secure.apisms.it/xmlrpc/BRP/provisioning.py');

# parametri obbligatori (vedi paragrafo 3)
$parametri = array(
    'authlogin' => 'loginrivenditore',
    'authpasswd' => 'passwordrivenditore',
    'login' => base64_encode('logindacreare'),
    'password' => base64_encode('lapassword'),
    'ragione_sociale'=>base64_encode('la ragione sociale')
);
$xmlrpc_msg = new xmlrpcmsg('create_user', array(php_xmlrpc_encode($parametri)));
$xmlrpc_resp = $xmlrpc_client->send($xmlrpc_msg);
if($xmlrpc_resp->errno != 0) {
    # Qui si può gestire un controllo di errore
    die(sprintf("ERRORE %s", $xmlrpc_resp->errstr));
}
$decode = php_xmlrpc_decode($xmlrpc_resp->value());
# come output la struttura dati ritornata che poi si può integrare nei Vs. sistemi
print_r($decode);
?>
```

11.2 Esempio in VB .NET

Di seguito riportiamo il medesimo esempio riportato nel paragrafo precedente in PHP equivalente in VB .NET.

La libreria che consigliamo è scaricabile dall'indirizzo: <http://www.xml-rpc.net/>

```
Imports CookComputing.XmlRpc
Imports System.Text
Public Class SMSConnector
    Private _authlogin As String = ""
    Private _authpasswd As String = ""
    Private _proxyURL As String = ""

    Private _proxy As ISMSproxy
    Public Sub New(ByVal proxyurl As String, ByVal authlogin As String, ByVal authpasswd As String)
        _proxyURL = proxyurl
        _authlogin = authlogin
        _authpasswd = authpasswd
        _proxy = XmlRpcProxyGen.Create(GetType(ISMSproxy))
    End Sub
    Public Function createUser(ByVal login As String, ByVal password As String, ByVal ragione_sociale As String)
        Dim params As New createUserParam
        params.authlogin = _authlogin
        params.authpasswd = _authpasswd
        params.login = convertToBase64String(login)
        params.password = convertToBase64String(password)
        params.ragione_sociale = convertToBase64String(ragione_sociale)
        Try
            Return _proxy.create_user(params)
        Catch ex As Exception
            Throw ex
        End Try
    End Function
    Private Function convertToBase64String(ByVal value As String) As String
        Return System.Convert.ToBase64String(Encoding.Default.GetBytes(value))
    End Function
End Class
<XmlRpcUrl("https://secure.apisms.it/xmlrpc/BRP/provisioning.py")> _
Public Interface ISMSproxy
    <XmlRpcMethod("create_user")> _
    Function create_user(ByVal params As createUserParam) As Integer
End Interface
Public Class baseParams
    Public authlogin As String
    Public authpasswd As String
End Class
Public Class createUserParam
    Inherits baseParams
    Public login As String
    Public password As String
    Public ragione_sociale As String
    ' --- parametri opzionali.. ---
    'Public cellulare As String
    'Public telefono As String
    'Public email As String
    'Public soglia_notifica_credito As Decimal
    'Public scadenza_credito As Date
    'Public note As String
End Class
```

II Changes

09/04/2008 – Versione I

06/05/2008 – Aggiunto encoding base64 su tutti i campi testo che possono contenere caratteri estesi., Cambiato URL base.

Classificazione documento

Revisione	R 81 - 11/07/16 23.08
Versione	I

Classificazione di sicurezza

Protocollo trasmissione	HTTPS
Porta TCP/IP	443
Protocollo applicativo	XML-RPC
URL base di accesso	https://secure.apisms.it/
Livello di accesso	BRP – Provisioning Rivenditore
Procedure autorizzate	provisioning.py [create_user(), set_route(), add_credit(), get_credit(), modify_user(), get_user_id(), get_users_list()]