amigopod XMLRPC Interface

Unified Visitor Management API

CONFIDENTIAL

Revision	0.9
Date	4 December 2007
	Copyright © 2007 amigopod Pty Ltd
Web	www.amigopod.com

Table of Contents

Introduction

This document describes the XMLRPC interface available to third-party applications that will integrate with the amigopod Visitor Management Appliance.

Audience

- Developers of integrated applications. Some familiarity with HTTP based web services and XMLRPC is assumed.
- System administrators of the amigopod appliance.

System Requirements

- amigopod Guest Manager 1.8.0
- XMLRPC client

More Information

For more details about XMLRPC, or to read the XML-RPC specification, visit www.xmlrpc.org.

To contact amigopod support, visit www.amigopod.com.

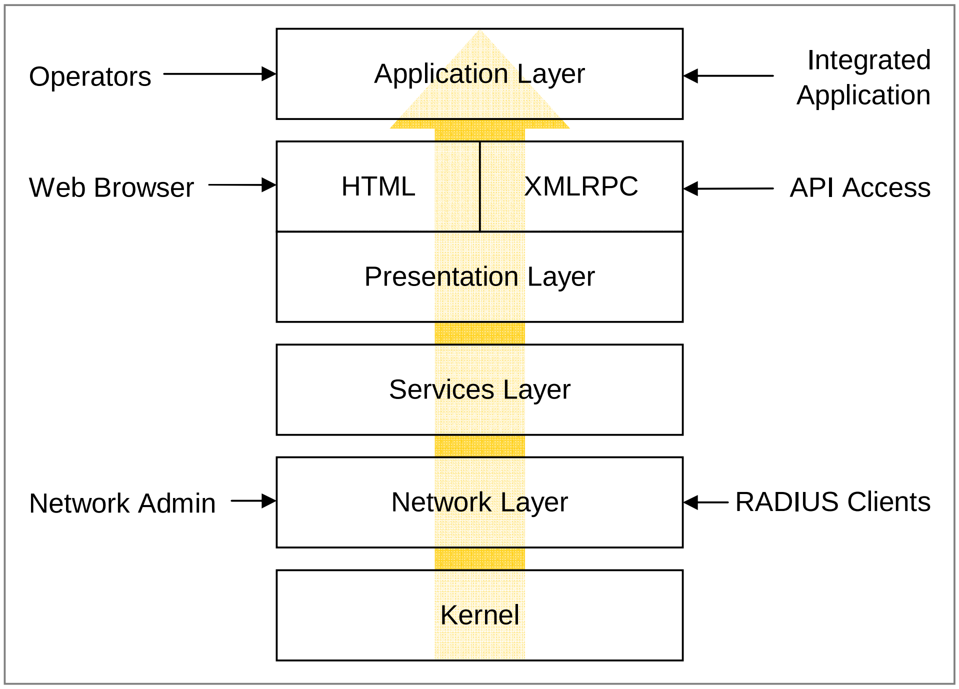
API Overview

About the API

The amigopod XMLRPC API provides direct access to the underlying functionality of the amigopod Visitor Management Appliance. Developers wishing to provide integrated applications can make use of this API to programmatically perform actions that would otherwise require manual operation of the GUI.

Architecture Overview

The amigopod VMA software is built using multiple layers of software.



At the lowest level, the kernel provides basic functions common to the entire system. This includes the web interface framework, appliance operating system, and runtime support services.

The network layer provides critical networking support, including the RADIUS server and the ability for network administrators to manage and control the networking aspects of the VMA.

The services layer provides one or more implementations of application services that are used by the layers above. Examples of these services include managing a user database used for AAA, handling the authentication of operators, and providing translated text resources.

The presentation layer supplies the tools and framework necessary for the VMA software to interact with the outside world. The basic presentation layer services include authentication checks, session establishment, input checking, validation and conversion, and command execution. Both

XMLRPC and HTML presentation methods are supplied, which adapt the underlying basic presentation to appropriate conventions suitable for a machine-to-machine or human-to-machine interaction.

The application layer provides the page templates, business logic, and the concrete features making up visitor management applications, such as Guest Manager or Hotspot Manager. These applications are built using the services provided by the lower layers.

API Symmetry

Because amigopod VMA applications are built using the framework supplied by the presentation layer, there is a direct symmetry between application features provided to operators using a web browser (HTML presentation), and application features provided to external applications using the XMLRPC presentation.

In particular, the following items are shared between HTML and XMLRPC presentations:

- Access control
- Parameter names
- Parameter validation rules
- Customized fields and their rules

Access Control

Page privileges are applied to XMLRPC authenticated sessions in the same way as the HTML user interface. However, XMLRPC access also requires the XMLRPC API privilege to be granted.

Parameter Names

The parameter names passed to the XMLRPC interface are the same as the field names in the HTML user interface.

Parameter Validation

Each field of the forms in the HTML user interface is subject to validation according to the rules defined for that field. The same rules also apply to XMLRPC parameters.

If a required field is missing, or an invalid value for a field is supplied, an error is generated by the presentation layer and returned to the XMLRPC client.

Field Customization

Some forms in the HTML user interface are customizable. New fields may be defined (for example, stored with a guest account) and used as part of the form in the presentation layer. Policy-based processing of fields includes form validation, data conversion and input formatting, and user interface metadata (for example, the list of valid options for a multiple-choice “dropdown” control).

These custom fields are automatically inherited as XMLRPC parameters, and the same policy-based processing is applied to them.

Parameter Types

The XMLRPC specification supports a wide range of data types. The following data types are supported by the XMLRPC presentation layer:

Data Type	Description
Array	Associative array using numeric keys with a 0-based index
Boolean	True or false
Flag	Scalar value of 0 or 1, implicitly Boolean
Integer	Integer, 32-bit range
Number	Numeric value, floating point OK
Scalar	Non-array value
String	String, UTF-8 encoded
Struct	Associative array using string keys May be nested (shown in the syntax as **first.second**, etc.)

Data Representation

Unless otherwise specified, all strings should be considered to be UTF-8 encoded Unicode.

Dates and times are represented in an ISO-8601 compatible format

```
YYYY-MM-DD hh:mm:ss
```

XMLRPC Faults

An XMLRPC Fault is a specific kind of return value indicating that an error has occurred in the presentation layer. The return value is a struct containing three named values:

Name	Type	Description
error	Flag	Set to 1 for an XMLRPC Fault
faultCode	Integer	Status code indicating the cause of the fault
faultString	String	Description of the fault

This type of return might appear as

```
'error' => 1,
'faultCode' => 401,
'faultString' => 'Invalid username or password',
```

These are the predefined XMLRPC Fault codes:

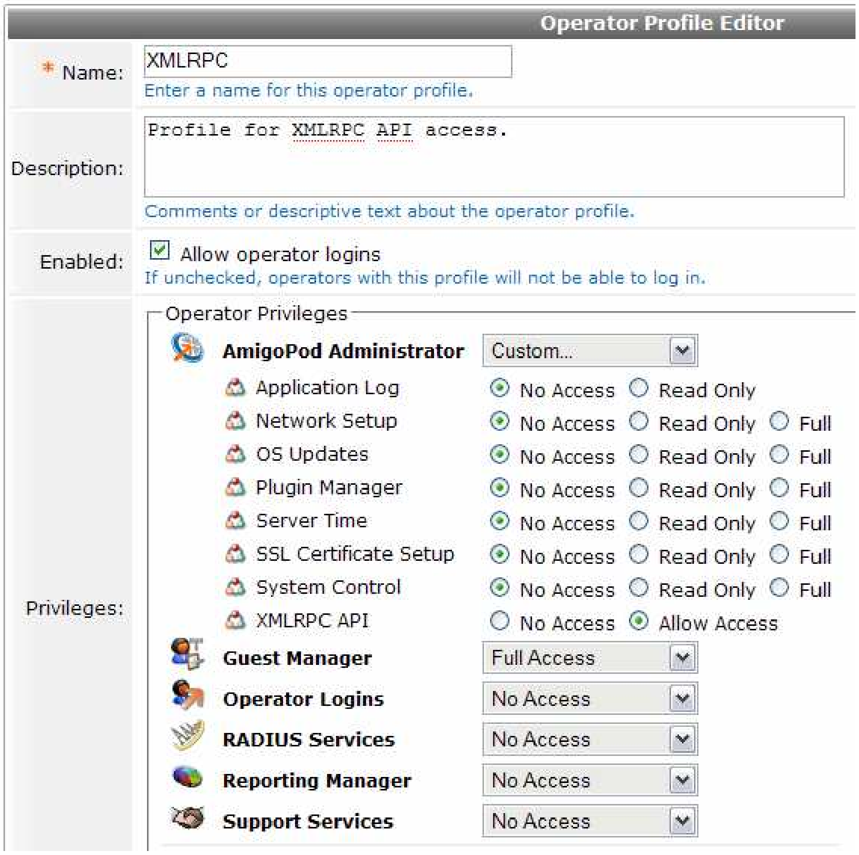
Code	Description
401	Authentication problem – invalid username or password

Code	Description
404	File implementation of XMLRPC method not found
501	XMLRPC implementation not found
502	XMLRPC method registration failed
503	XMLRPC server creation failed
504	Access denied
505	No XMLRPC implementation for this page

Access Control

Accessing the API requires an operator account with a profile that has the XMLRPC API privilege, plus any privileges required for the API calls.

- To create a sample XMLRPC API profile, use the following settings:



Operator Profile Editor

* Name: XMLRPC
Enter a name for this operator profile.

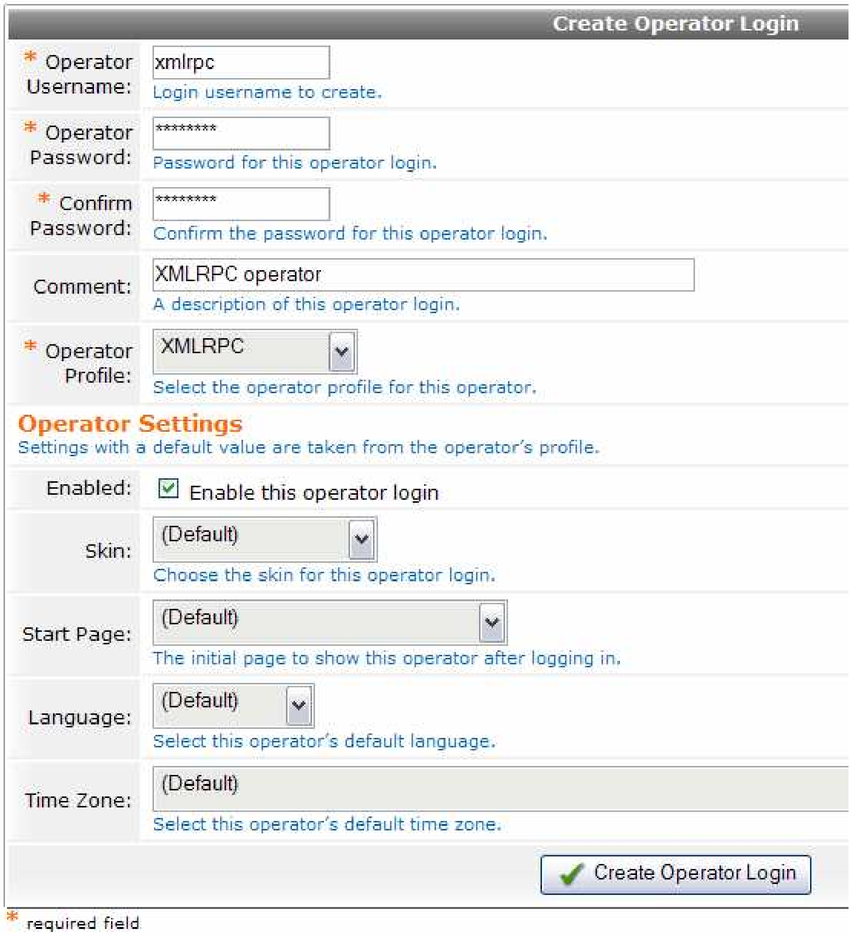
Description: Profile for XMLRPC API access.
Comments or descriptive text about the operator profile.

Enabled: ☑ Allow operator logins
If unchecked, operators with this profile will not be able to log in.

Privileges: Operator Privileges

AmigoPod Administrator	Custom...		
Application Log	● No Access	○ Read Only	
Network Setup	● No Access	○ Read Only	○ Full
OS Updates	● No Access	○ Read Only	○ Full
Plugin Manager	● No Access	○ Read Only	○ Full
Server Time	● No Access	○ Read Only	○ Full
SSL Certificate Setup	● No Access	○ Read Only	○ Full
System Control	● No Access	○ Read Only	○ Full
XMLRPC API	○ No Access	● Allow Access	
Guest Manager	Full Access		
Operator Logins	No Access		
RADIUS Services	No Access		
Reporting Manager	No Access		
Support Services	No Access		

- Use the XMLRPC profile when creating a user account for API access:



* Operator Username:	xmlrpc Login username to create.
* Operator Password:	******** Password for this operator login.
* Confirm Password:	******** Confirm the password for this operator login.
Comment:	XMLRPC operator A description of this operator login.
* Operator Profile:	XMLRPC Select the operator profile for this operator.

Operator Settings

Settings with a default value are taken from the operator's profile.

Enabled:	☑ Enable this operator login
Skin:	(Default) Choose the skin for this operator login.
Start Page:	(Default) The initial page to show this operator after logging in.
Language:	(Default) Select this operator's default language.
Time Zone:	(Default) Select this operator's default time zone.

* required field

Invoking the API

An XMLRPC method call consists of:

- An XML document specifying the method name and parameters
- Sent as a HTTP POST with `Content-Type: text/xml`
- Using HTTP Basic user authorization
- At https://amigopod/xmlrpc.php

SSL Security

Different levels of certificate validation checks may be necessary, depending on the SSL certificate that has been installed. This corresponds to the user interface provided by web browsers for certificate trust and verification.

The examples presented in this document assume a self-signed certificate has been installed, and reduce the level of SSL verification accordingly. In a secure environment, make use of the peer verification that SSL provides by either installing an SSL certificate signed by a well-known certificate authority, or issue your own certificates from a network's certificate authority.

Method Summary

These methods are currently available:

Method Name	Synopsis
amigopod.guest.change.expiration	Change the expiration time of a guest account
amigopod.guest.create	Create a new guest account
amigopod.guest.delete	Disable or remove a guest account
amigopod.guest.edit	Change one or more properties of a guest account
amigopod.guest.enable	Enable a guest account
amigopod.guest.get	List one or more guest accounts
amigopod.guest.list	List guest accounts
amigopod.guest.reset.password	Reset a guest account's password

For more details on these methods, refer to the API Documentation.

Sample Code

For sample PHP code that demonstrates how to invoke the amigopod XMLRPC API, refer to Appendix 1.

API Documentation

Method amigopod.guest.change.expiration

Change the expiration time of a guest account.

Parameters

Name	Type	Description
uid	Scalar	ID of the guest account to update
guestaccountexpiry	Scalar	Amount of time in hours before the guest account will expire

Return Values

Name	Type	Description
error	Flag	0 if successful, 1 if an error occurred
message	String	Status message describing the operation
item	Struct	Updated user information record
*_error	String	Field-specific error message
*_error_flag	Flag	Field-specific error flag, set to 1 if present

Access Control

Requires the **change_expiration** privilege (Guest Manager → Change Expiration).

Example Usage

Sample parameters for call:

```
'uid' => 162,
'guestaccountexpiry' => 24,
```

Result returned by a successful operation:

```
'error' => 0,
'message' => 'Changed expiration time of guest account
 Account will expire at 2007-11-28 23:27:00',
'item' => array (
  'id' => 162,
  'do_expire' => 4,
  'expire_time' => 1196256420,
  'username' => '',
)
```

Result returned by a failed operation:

```
'uid' => 162,
'username' => '',
'expiration_time' => '',
'user_enabled' => '',
'guestaccountexpiry_error' => 'Please choose from one of these
options.',
'guestaccountexpiry_error_flag' => 1,
'error' => 1,
```

Method amigopod.guest.create

Create a new guest account.

Parameters

Name	Type	Description
sponsor_name	String	Name of the person sponsoring the guest account
visitor_name	String	Name of the visitor
visitor_company	String	Company name of the visitor
email	String	The visitor's email address. This will become their username to log into the network
expire_after	Numeric	Amount of time before the account will expire. Specified in hours
expire_time	String	Optional date and time at which the account will expire
role_id	Integer	RADIUS Role ID to assign to the account
creator_accept_terms	Flag	Set to 1 to indicate acceptance of the service's terms of use
*****	*	Other fields as specified by **create_user** form customization

Return Values

Name	Type	Description
error	Flag	Set to 1 if the account was not created
id	Integer	Set to the ID of the guest account if the account was created
password	String	Set to a randomly generated value (default behaviour only)
_error	String	Field-specific error message
_error_flag	Flag	Field-specific error flag, set to 1 if present

Access Control

Requires the **create_user** privilege (Guest Manager → Create New Guest Account).

Example Usage

Sample parameters for call:

```
'sponsor_name' => 'Sponsor Name',
'visitor_name' => 'Visitor Name',
'visitor_company' => 'Visitor Company',
'email' => 'demo@example.com',
'expire_after' => 4,
'expire_time' => '',
'role_id' => 2,
'visitor_phone' => '0',
'creator_accept_terms' => 1,
```

Result returned by a successful operation:

```
'username' => 'demo@example.com',
'password' => '73067792',
'role_id' => 2,
'role_name' => 'Guest',
'simultaneous_use' => '1',
'do_schedule' => 0,
'enabled' => true,
'expire_time' => 1196769257,
'do_expire' => 4,
'expire_postlogin' => 0,
'sponsor_name' => 'Sponsor Name',
'visitor_name' => 'Visitor Name',
'visitor_company' => 'Visitor Company',
'email' => 'demo@example.com',
'creator_accept_terms' => true,
'id' => 1,
```

Result returned by a failed operation:

```
'password' => '78342029',
'expire_time' => '',
'submit' => '',
'sponsor_name_error' => 'You cannot leave this field blank.',
'sponsor_name_error_flag' => 1,
'visitor_name_error' => 'You cannot leave this field blank.',
'visitor_name_error_flag' => 1,
'visitor_company_error' => 'You cannot leave this field blank.',
'visitor_company_error_flag' => 1,
'email_error' => 'Please enter a valid email address.',
'email_error_flag' => 1,
'expire_after_error' => 'Please choose from one of the available
options.',
'expire_after_error_flag' => 1,
'expire_time_error' => 'Please enter a valid date and time.',
'expire_time_error_flag' => 1,
'role_id_error' => 'Parameter must be provided',
'role_id_error_flag' => 1,
'creator_accept_terms_error' => 'You must accept the terms of use to
continue.',
'creator_accept_terms_error_flag' => 1,
'error' => 1,
```

Method amigopod.guest.delete

Disable or remove a guest account.

Parameters

Name	Type	Description
uid	Integer	ID of the guest account to delete
delete_account	Flag	Set to 0 to disable the account, 1 to delete the account

Return Values

NOTE This function may return a Boolean **false** value if some input parameters are invalid.

Name	Type	Description
error	Flag	Set to 1 if the account was not deleted
message	String	Message describing the success or failure of the operation
item	Struct	User structure containing updated field values
uid	Integer	ID of the guest account
_error	String	Field-specific error message
_error_flag	Flag	Field-specific error flag, set to 1 if present

Access Control

Requires the **remove_account** privilege (GuestManager → Remove Accounts).

Example Usage

Sample parameters for call:

```
'uid' => '162',
'delete_account' => '0',
```

Result returned by a successful operation:

```
'error' => 0,
'message' => 'Disabled guest account ',
'item' =>
array (
  'id' => 162,
  'enabled' => 0,
  'username' => '',
),
```

Result returned by a failed operation:

```
'uid' => 162,
'username' => '',
'expiration_time' => '',
'user_enabled' => '',
'delete_account_error' => 'Please choose from one of these options.',
'delete_account_error_flag' => 1,
'error' => 1,
```

Method amigopod.guest.edit

Change one or more properties of a guest account.

Parameters

Name	Type	Description
uid	Integer	ID of the guest account to edit

Name	Type	Description
username	String	Name of the guest account
password	String	May be: ▪ **random_password** to indicate the account's password should be set to a random password ▪ **password_value** to indicate the account's password should be set to the value in the password_value field ▪ the empty string to leave the password unmodified
password_value	String	Optional password to set the guest account's password (if the **password** field is **password_value**)
role_id	Integer	RADIUS Role ID to assign to the guest account
enabled	Flag	Boolean value indicating whether the account is enabled
simultaneous_use	Integer	Number of simultaneous sessions allowed by the account
do_schedule	Flag	Flag indicating if the account should be enabled at schedule_time
schedule_time	String	Date and time at which the account will be enabled
do_expire	Integer	Action to take when the expire_time is reached
expire_time	String	Time at which the account will expire
expire_postlogin	Integer	Time period for which the account will be valid after the first login, or 0 for indefinitely

Return Values

Name	Type	Description
error	Flag	Set to 1 if the account was not modified
message	String	Message describing the success or failure of the operation
item	Struct	User structure containing updated field values
uid	Integer	ID of the guest account
_error	String	Field-specific error message
_error_flag	Flag	Field-specific error flag, set to 1 if present

Access Control

Requires the **full-user-control** privilege (Guest Manager → Full User Control).

Example Usage

Sample parameters for call:

```
'uid' => 162,
'username' => 'demo@example.com',
'password' => 'password_value',
'password_value' => 'password',
'role_id' => 2,
'enabled' => 1,
'simultaneous_use' => 1,
'do_schedule' => 0,
'schedule_time' => '',
'do_expire' => 4,
'expire_time' => '2007-12-01 00:00:00',
'expire_postlogin' => 0,
```

Sample successful call:

```
'error' => 0,
'message' => 'Edited properties of guest account demo@example.com',
'item' =>
array (
  'id' => 162,
  'username' => 'demo@example.com',
  'role_id' => 2,
  'enabled' => true,
  'simultaneous_use' => 1,
  'do_schedule' => 0,
  'do_expire' => 4,
  'expire_postlogin' => 0,
  'role_name' => 'Guest',
  'expire_time' => 1196431200,
),
```

Sample failed call:

```
'uid' => 162,
'random_password' => '59447116',
'password_value' => '',
'schedule_time' => '',
'expire_time' => '',
'user_enabled' => '',
'username_error' => 'You cannot leave this field blank.',
'username_error_flag' => 1,
'password_error' => 'Please choose from one of the available options',
'password_error_flag' => 1,
'role_id_error' => 'Please choose from one of the available options',
'role_id_error_flag' => 1,
'enabled_error' => 'Parameter must be provided',
'enabled_error_flag' => 1,
'simultaneous_use_error' => 'Please enter a non-negative integer
value.',
'simultaneous_use_error_flag' => 1,
'do_schedule_error' => 'Please choose from one of the available
options',
'do_schedule_error_flag' => 1,
'schedule_time_error' => 'Parameter must be a string',
'schedule_time_error_flag' => 1,
'do_expire_error' => 'Please choose from one of the available
options',
'do_expire_error_flag' => 1,
'expire_time_error' => 'Parameter must be a string',
'expire_time_error_flag' => 1,
'expire_postlogin_error' => 'Please choose from one of the available
options',
'expire_postlogin_error_flag' => 1,
'error' => 1,
```

Method amigopod.guest.enable

Enable a guest account.

Parameters

Name	Type	Description
uid	Integer	ID of the guest account to enable

Return Values

NOTE This function may return a Boolean **false** value if some input parameters are invalid.

Name	Type	Description
error	Flag	Set to 1 if the account was not enabled
message	String	Message describing the success or failure of the operation
item	Struct	User structure containing updated field values
uid	Integer	ID of the guest account
_error	String	Field-specific error message
_error_flag	Flag	Field-specific error flag, set to 1 if present

Access Control

Requires the **remove_account** privilege (Guest Manager → Remove Accounts).

Example Usage

Sample parameters for call:

```
'uid' => '162',
```

Sample successful call:

```
'error' => 0,
'message' => 'Guest account  has been re-enabled',
'item' =>
array (
  'id' => 162,
  'enabled' => 1,
  'username' => '',
),
```

Sample failed call:

```
'error' => 1,
'message' => 'Account not found: ID 162',
```

Method amigopod.guest.get

List one or more guest accounts.

Parameters

Name	Type	Description
id	Integer Array	Retrieve a single guest account by ID (integer parameter), or multiple guest accounts by ID (array parameter)

Return Values

Name	Type	Description
id	Mixed	ID or IDs of the guest accounts being returned
users	Array	▪ If a single ID was requested, **users** contains the guest account requested (or an **error** field if an error occurred) ▪ If multiple IDs were requested, **users** contains an array of results

Access Control

Requires the **guest_users** privilege (Guest Manager → List Guest Accounts).

Example Usage

Sample parameters:

```
'id' => array(150, 162)
```

Sample successful call:

```
'id' =>
array (
  0 => 150,
  1 => 162,
),
'users' =>
array (
  0 =>
  array (
    'id' => '150',
    'username' => '44454318',
    'enabled' => '1',
    'role_id' => '2',
    'email' => '',
    'notes' => 'GuestManager account 22 of 30 created by root from
192.168.2.3',
    'do_expire' => '0',
    'expire_time' => '',
    'simultaneous_use' => '1',
    'expire_postlogin' => '0',
    'do_schedule' => '0',
    'schedule_time' => '',
    'ip_address' => '',
    'netmask' => '',
  ),
  1 =>
  array (
    'id' => '162',
    'username' => 'demo@example.com',
    'enabled' => '1',
    'role_id' => '2',
    'email' => 'demo@example.com',
    'notes' => '',
    'do_expire' => '4',
    'expire_time' => '1196253480',
    'simultaneous_use' => '1',
    'expire_postlogin' => '0',
    'do_schedule' => '0',
    'schedule_time' => '',
    'ip_address' => '',
    'netmask' => '',
    'auto_send_sms' => '',
    'creator_accept_terms' => '1',
    'role_name' => 'Guest',
    'sponsor_name' => 'Sponsor Name',
    'visitor_company' => 'Visitor Company',
    'visitor_name' => 'Visitor Name',
    'visitor_phone' => '0',
  ),
),
```

Sample failed call:

```
'id' => 162,
'users' =>
array (
  'error' => 1,
  'message' => 'Account not found: ID 162',
),
```

Method amigopod.guest.list

List guest accounts.

Parameters

Name	Type	Description
details	Flag	Optional parameter; if set to 1 then full details of all guest accounts are returned, otherwise only the IDs are returned

Return Values

Name	Type	Description
ids	Array	Array of guest account IDs (if **details** was 0)
users	Array	Array of guest account structures (if **details** was 1)

Access Control

Requires the **guest_users** privilege (Guest Manager → List Guest Accounts).

Example Usage

Sample parameters

```
'details' => 0,
```

Sample successful call:

```
'ids' =>
array (
  0 => '37',
  1 => '141',
  2 => '40',
  ...
),
```

Method amigopod.guest.reset.password

Reset a guest account's password to a random value.

Parameters

Name	Type	Description
uid	Integer	ID of the guest account to reset the password for

Return Values

Name	Type	Description
error	Flag	Set to 1 if the password was not reset
message	String	Message describing the success or failure of the operation
item	Struct	User structure containing updated field values
_error	String	Field-specific error message
_error_flag	Flag	Field-specific error flag, set to 1 if present

Access Control

Requires the **reset_password** privilege (Reset Password).

Example Usage

Sample parameters for call:

```
'uid' => 162,
```

Sample successful call:

```
'error' => 0,
'message' => 'Guest account password reset for
 Password changed to 37172833',
'item' =>
array (
  'id' => 162,
  'password' => '37172833',
  'username' => '',
),
```

Sample failed call:

```
'error' => 1,
'message' => 'Account not found: ID 162',
```