WebsitePanel Integration API

Author: Feodor Fitsner
Last updated: 02/12/2010
Version: 1.0

Table of Contents

Introduction

This document describes basic WebsitePanel Integration API that could be used in 3rd-party applications.

Requirements

- WSE 3.0 or Basic Authentication
- Each API call should have user's credentials provided

Basic Authentication

For interacting with WebsitePanel API you should use Basic Authentication. WebsitePanel recognizes "Authorization" header with the user credentials provided in the following format: username:password.

Basic Web Services

Manage User Accounts

AddUser (UserInfo user, bool sendLetter) – creates a user account and sends confirmation to the user's email.

Endpoint: http://<Enterprise Server URL>/esUsers.asmx

The “user” parameter requires the following fields to be set:

OwnerId – ID of the user that owns account to be created;

RoleId – ID of the user role to be created. The following roles available: 1 – Administrator, 2 – Reseller and 3 – User;

StatusId – ID of the user status to be created. The following statuses available: 1 – Active, 2 – Suspended, 3 – Cancelled and 4 – Pending;

IsDemo – specifies that created account intends for demonstration purposes only. When user is in demo all management functionality becomes inaccessible;

IsPeer – specifies that created account is a peer for the user’s owner;

Username – username of the account to be created;

Password – password of the account to be created;

FirstName – first name of the account to be created;

LastName – last name of the account to be created;

Email – e-mail address of the account to be created;

All other fields of the “user” parameter are optional.

Also you may decide to set “sendLetter” parameter to “true” if you would like to send user registration notification.

Method returns ID of the created account or error code occurred during account creation.

UserExists (string username) – gets a value indicating whether a username already exists.

Endpoint: http://<Enterprise Server URL>/esUsers.asmx

AuthenticateUser (string username, string password, string ip) – gets a value indicating that provided credentials are correct.

Endpoint: http://<Enterprise Server URL>/esAuthentication.asmx

Method returns the following statuses: -109 – wrong username has been specified, -110 – wrong password has been specified, -105 – user account has been cancelled, -103 – user account still in pending status, 0 – user credentials specified are correct;

GetUserById (int userId) – gets user account details by ID specified;

Endpoint: http://<Enterprise Server URL>/esUsers.asmx

GetUserByUsername (string username) – gets user account details by username specified;

Endpoint: http://<Enterprise Server URL>/esUsers.asmx

GetUsers (int ownerId, bool recursive) – gets all users for the specified owner. If you would like to obtain users recursively then set “recursive” parameter to “true”;

Endpoint: http://<Enterprise Server URL>/esUsers.asmx

UpdateUser (UserInfo user) – updates user account details. Returns status 0 if account updated successfully;

Endpoint: http://<Enterprise Server URL>/esUsers.asmx

DeleteUser (int userId) – deletes user account from the database. Returns 0 if account deleted successfully;

Endpoint: http://<Enterprise Server URL>/esUsers.asmx

ChangeUserPassword (int userId, string password) – changes user account password. Returns 0 if password changed successfully;

Endpoint: http://<Enterprise Server URL>/esUsers.asmx

ChangeUserStatus (int userId, UserStatus status) – changes user account current status. The following statuses available: 1 – Active, 2 – Suspended, 3 – Cancelled and 4 – Pending; Returns 0 if status changed successfully;

Endpoint: http://<Enterprise Server URL>/esUsers.asmx

Manage Hosting Spaces

AddPackage (int userId, int planId, string packageName, string packageComments, int statusId, DateTime purchaseDate) – adds a package to the specified user account.

Endpoint: http://<Enterprise Server URL>/esPackages.asmx

As you can see the most of parameters are self-explainable. The only two parameters require additional explanation:

statusId – status of the user's package to be created. The following statuses are available: 1 – Active, 2 - Suspended, 3 – Cancelled, 4 – New;

purchaseDate – package purchase date (required);

Methods returns PackageResult object that contains created user package ID or error code and exceeding quotas if any;

UpdatePackage (PackageInfo package) – updates a hosting package. Returns PackageResult object that has 0 status code if package updated successfully or error code and exceeding quotas if any;

Endpoint: http://<Enterprise Server URL>/esPackages.asmx

DeletePackage (int packageId) – deletes a user's hosting package. Returns 0 if package deleted successfully;

Endpoint: http://<Enterprise Server URL>/esPackages.asmx

ChangePackageStatus (int packageId, PackageStatus status) – changes user's hosting package current status. The following statuses are available: 1 – Active, 2 - Suspended, 3 – Cancelled, 4 – New. Returns 0 if package status changed successfully;

Endpoint: http://<Enterprise Server URL>/esPackages.asmx

UpdatePackageName (int packageId, string packageName, string packageComments) – updates user's package name and comments. Returns 0 if package name and comments changed successfully.

Endpoint: http://<Enterprise Server URL>/esPackages.asmx

Upgrade/Downgrade User's Packages

AddPackageAddon (PackageAddonInfo addon) – adds a hosting addon to the user's hosting package.

Endpoint: http://<Enterprise Server URL>/esPackages.asmx

The "addon" parameter requires the following fields to be set:

PackageId – ID of the package to be upgraded;

PlanId – ID of the hosting addon you wish to add to the hosting package;

Quantity – quantity of the hosting addon. The minimum value is 1;

StatusId – ID of the hosting addon status. The following statuses are available: 1 – Active, 2 - Suspended, 3 – Cancelled, 4 – New;

PurchaseDate – hosting addon purchased date;

Returns PackageResult object that contains created hosting package addon ID or error code and exceeding quotas if any.

UpdatePackageAddon (PackageAddonInfo addon) – updates a hosting package addon details. Returns PackageResult object that contains 0 status code if hosting package addon updated successfully or error code and exceeding quotas if any.

Endpoint: http://<Enterprise Server URL>/esPackages.asmx

DeletePackageAddon (int packageAddonId) – deletes addon from the hosting package assigned. Returns 0 if addon deleted from hosting package successfully.

Endpoint: http://<Enterprise Server URL>/esPackages.asmx

GetPackageAddon (int packageAddonId) – gets a hosting package addon details.

Endpoint: http://<Enterprise Server URL>/esPackages.asmx

Calling WebsitePanel Web Service from PHP

The following sample code adds a new user to WebsitePanel database. NuSOAP library is used to work with web services from PHP:

```
<?php
require_once ("nusoap/lib/nusoap.php");

/* Set address of Enterprise Server web service */
$WSP = new soapclient('http://127.0.0.1:9002/esUsers.asmx?WSDL', true);
$WSP->soap_defencoding = 'utf-8';

/* Set credentials - this is WebsitePanel login*/
$WSP->setCredentials("serveradmin", "serveradmin");

/* specify method parameters */
$param = array("username" => "serveradmin");

/* call web method */
$result = $WSP->call('UserExists', array('parameters' => $param),
      'http://smbsaas/WebsitePanel/enterpriseserver',
      'http://smbsaas/WebsitePanel/enterpriseserver/UserExists');

/* process results */
if ($WSP->fault) {
    echo '<h2>Fault </h2><pre>'; print_r($result); echo '</pre>';
} else {
    $err = $WSP->getError();
    if ($err) {
        echo '<h2>Error</h2><pre>' . $err . '</pre>';
    } else {
        echo '<h2>Result</h2><pre>'; print_r($result); echo '</pre>';
    }
}

echo '<h2>Request</h2><pre>' . htmlspecialchars($WSP->request, ENT_QUOTES) .
'</pre>';
echo '<h2>Response</h2><pre>' . htmlspecialchars($WSP->response, ENT_QUOTES)
. '</pre>';
echo '<h2>Debug</h2><pre>' . htmlspecialchars($WSP->debug_str, ENT_QUOTES) .
'</pre>';

?>
```

Document History

Version	Date Released	Author	Changes
1.0	09/14/2009	Feodor Fitsner	Initial document version in downloadable format.