
SWsoft, Inc.

Plesk 8.0 for Linux/Unix Integration Handbook

Plesk XML API



(c) 1999-2006

Contents

| | |
|---|----------|
| About this guide | 3 |
| <hr/> | |
| About Plesk XML API | 4 |
| <hr/> | |
| How Plesk XML API works?..... | 4 |
| Who can use Plesk API protocol? | 4 |
| What versions of Plesk API protocol are in use? | 4 |
| | |
| Creating a user account in Plesk | 5 |
| <hr/> | |
| Request..... | 5 |
| Response | 5 |
| | |
| Adding a domain in Plesk | 7 |
| <hr/> | |
| Request..... | 7 |
| Response | 7 |
| | |
| Creating a session for Plesk user | 8 |
| <hr/> | |
| Request..... | 8 |
| Response | 8 |
| | |
| Customizing Plesk UI appearance | 9 |
| <hr/> | |
| Request..... | 9 |
| Selecting a GUI skin | 9 |
| Setting a locale | 9 |
| Customizing top frame..... | 10 |
| Customizing left frame..... | 10 |
| Adding custom buttons | 11 |

About this guide

This guide's goal is to explain the mechanism of integration of third party software with Plesk by means of Plesk API. It shows how the most common tasks can be performed using an integration with a web-based store as an example. Comprehension of the material presented here requires basic understanding of XML and HTTP.

Examples given here do not fully cover the entire Plesk API. For its complete description please refer to the official Plesk SDK documentation.

About Plesk XML API

In order to provide means of interaction between Plesk and third-party software, Plesk provides a specially designed XML-based API. This interface exposes a set of functions for operating Plesk, such as user account and domain creation and management.

How Plesk XML API works?

Plesk communicates with other applications by means of exchanging specially formed XML packets via HTTPS (Secure HTTP) protocol. All requests in POST format are directed to the handler located at:

```
https://<plesk_server_address>:8443/enterprise/control/agent.php
```

The handler processes the requests and returns the response, also formed as an XML packet, containing the results of the performed operations.

Who can use Plesk API protocol?

The use of Plesk API is allowed for the Plesk Administrator and Plesk Clients. Plesk Administrators are capable of using it by default, Plesk Clients must be granted this option: “CP access” and “Ability to use remote XML interface” permissions must be enabled.

Authentication is provided by HTTP environment. The request must contain the following headers:

- HTTP_AUTH_LOGIN – login name for the user account (“admin” for Plesk Administrator),
- HTTP_AUTH_PASSWD – password for this account.

What versions of Plesk API protocol are in use?

Plesk API protocol evolves in sync with Plesk constantly: if Plesk obtains new features, they are reflected in the protocol right away. The most recent versions of Plesk API protocol for Plesk for Linux/Unix are:

| Plesk version | Protocol version |
|-----------------------------|------------------|
| Plesk 7.5.4 for Linux/Unix | 1.3.5.1 |
| Plesk 8.0.0 for Linux/Unix | 1.4.1.0 |
| Plesk 8.1.0* for Linux/Unix | 1.4.2.1 |

* – this version of Plesk for Linux/Unix has not yet been released.

Note that every version of Plesk supports one current version the API protocol as well as all earlier versions. For instance, Plesk 8.0.0 for Linux/Unix supports version 1.4.1.0 of the protocol, but it will also work with version 1.3.5.1 and earlier.

This guide describes integration based on the version 1.4.1.0 of the protocol.

Creating a user account in Plesk

In order to create a user account in Plesk it is necessary to supply the following personal information about the user: **Contact Name, Login, Password, Phone Number, Fax Number, Email Address, Address, City, State, Postal Code, and Country.**

Request

The following XML packet shows how create a client account for user **John Doe** with login **jdoe** and password **passwd**:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<packet version="1.4.1.0">
  <client>
    <add>
      <gen_info>
        <pname>John Doe</pname>
        <login>jdoe</login>
        <password>passwd</password>
      </gen_info>
      <permissions>
        <remote_access_interface>true</
remote_access_interface>
      </permissions>
    </add>
  </client>
</packet>
```

Example 1: Creating a client account

In addition to the personal information, other parameters can be supplied. Such parameters include limits, permissions, client template, etc. Notice the `<permissions>` section in the example above. This section contains an element `<remote_access_interface>`, which provides the client with the ability to use API RPC. It is set to `true`, so the user will be able to log into Plesk through its web interface.

Response

After processing this packet Plesk will send back a response packet that will contain the result of performing the requested operation and the corresponding data. For example, in case of successful creation of the account, the following packet will be received:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<packet version="1.4.1.0">
  <client>
    <add>
      <result>
        <id>123321</id>
        <status>ok</status>
      </result>
    </add>
```

```
</client>
</packet>
```

Example 2: Successful client account creation response

This example shows the response packet that contains a newly created account's ID: **123321**, and the status of operation's completion: **ok**. If the operation should fail, the returning response will state the status of the operation as **error**, and will contain available information about the failure reasons:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<packet version="1.4.1.0">
  <client>
    <add>
      <result>
        <status>error</status>
        <errcode>1007</errcode>
      </result>
    </add>
  </client>
</packet>
```

Example 3: Failed client account creation response

This packet returns the error code in the element `<errcode>`, which can help identify the reason the account creation failed. In this example the returned error code is **1007**, which means such an account already exists.

Adding a domain in Plesk

Using Plesk API one can add and configure domains for a given client account. The following parameters are required: **Domain Name** and **IP Address**.

Request

The following XML packet shows how add a domain **johndoe.example.com**:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<packet version="1.4.1.0">
  <domain>
    <add>
      <gen_setup>
        <name>johndoe.example.com</name>
        <client_id>123321</client_id>
        <ip_address>192.168.62.129</ip_address>
      </gen_setup>
      <prefs>
        <www>>true</www>
      </prefs>
    </add>
  </domain>
</packet>
```

Example 4: Adding a domain

Notice the `<client_id>` parameter, which is set to the account ID for the client **John Doe**. It is necessary here to specify which client account the domain is added for.

Response

In case the domain was added successfully the response packet is quite simple. It delivers the status **ok** and the new domain id (**234432**):

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<packet version="1.4.1.0">
  <domain>
    <add>
      <result>
        <status>ok</status>
        <id>234432</id>
      </result>
    </add>
  </domain>
</packet>
```

Example 5: Successful domain addition response

In case of the failure the response packet is similar to that shown in Example 3, carrying the status of operation completion equal to **error** and the error code.

Creating a session for Plesk user

In order to enable a user to switch between systems (e.g. between web-based store and Plesk CP) a session must be created for the user on the Plesk server. The only required data for it is the user's **Login Name** in Plesk. However, additional parameters can be passed in order to provide more information.

Request

The following packet in addition to the login name **jd**oe passes two more parameters to specify the location the user comes to Plesk CP, they are `<user_ip>` and `<source_server>` – the IP address and the name of the server.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<packet version="1.4.1.0">
  <server>
    <create_session>
      <login>jd</login>
      <data>
        <user_ip>BASE64_IP</user_ip>
        <source_server>BASE64_URL</source_server>
      </data>
    </create_session>
  </server>
</packet>
```

Example 6: Creating a session in Plesk

Please note: all parameters in the `<data>` section of this packet must be base64 encoded. For instance, in PHP this can be done as follows:

```
base64_encode("http://openfusiondemo.plesk.ru/")
```

The result of this operation should be substituted for **BASE64_URL** in the example above.

Response

Successful creation of the session in Plesk returns status of completion **ok** and the creation session ID.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<packet version="1.4.1.0">
  <server>
    <create_session>
      <login>jd</login>
      <result>
        <status>ok</status>
        <id>bbbb20d80ea69a129969e951092dd470</id>
      </result>
    </create_session>
  </server>
</packet>
```

Example 7: Successful session creation response

Customizing Plesk UI appearance

In order to make the integration more complete, Plesk GUI can be customized to look and function more in tune with another product. By means of Plesk API one can change the logo image displayed at the top of the screen, customize elements of the left navigation pane, change entire GUI skin, set the locale, and more. All this is done during the session creation.

Request

Following is the same packet presented in Example 6, only complemented with additional parameters:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<packet version="1.4.1.0">
  <server>
    <create_session>
      <login>jdoe</login>
      <data>
        <user_ip>BASE64_IP</user_ip>
        <starting_url>BASE64_STARTING_URL</starting_url>
        <source_server>BASE64_URL</source_server>
        <navigation>BASE64_NAVIGATION</navigation>
        <logo_img_url>BASE64_LOGO_URL</logo_img_url>
        <logo_img_ref>BASE64_LOGO_REF</logo_img_ref>
        <locale_id>BASE64_LOCALE_ID</locale_id>
        <skin>BASE64_SKIN_NAME</skin>
        <domain_name>BASE64_DOMAIN_NAME</domain_name>
        <back_url>BASE64_BACK_URL</back_url>
      </data>
    </create_session>
  </server>
</packet>
```

Example 8: Creating a session in Plesk with customization parameters

Parameter `<domain_name>` specifies which of the domains is to be accessed when the user accesses the control panel.

The following sections explain the rest of the parameters that appeared and how they can be used.

Selecting a GUI skin

Parameter `<skin>` is used for specifying the GUI skin to be used for the user's CP. If the requested skin is not installed, the default skin will be used.

Setting a locale

Parameter `<locale_id>` allows selecting the locale to be used by Plesk CP. For example, passing **en-US** as the locale ID will result in user's Plesk CP being localized in English for US. If the requested locale is not installed in Plesk, the default locale will be used.

Customizing top frame

Plesk CP can be customized so as to propagate the branding of the provider. The logotype image at the top left corner can be replaced with that of the provider, so that it is at all times visible to the user. Parameter `<logo_img_url>` is responsible for passing the URL of the new logo image to Plesk.

Passing the following as **BASE64_LOGO_REF** will result in the default Plesk logo being replaced with the one the URL points to:

```
base64_encode("http://openfusiondemo.plesk.ru/logo.gif")
```

Additionally, it is possible to provide make the logo image a link and specify any desired location. To do that, the URL must be supplied as the parameter `<logo_img_url>`.

Customizing left frame

Plesk CP left frame – the navigation pane – can be customized to include sections with links in addition to the regular set. In order to do that, first, the current navigation pane contents must be obtained from Plesk:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<packet version="1.4.1.0">
  <ui>
    <get-navigation/>
  </ui>
</packet>
```

Example 9: Requesting current navigation pane contents

Important note: due to differences in available functionality, different types of Plesk users have different navigation panes. So, in order to get the contents for the client's navigation pane, the type of user must be communicated to Plesk. This is done by means of HTTP headers `HTTP_AUTH_LOGIN` and `HTTP_AUTH_PASSWD`, which in this case must contain the account login and password of the specific client.

Plesk will return the packet containing the navigation pane contents:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<packet version="1.4.1.0">
  <ui>
    <get-navigation>
      <result>
        <status>ok</status>
        <navigation>BASE64_NAVIGATION</navigation>
      </result>
    </get-navigation>
  </ui>
</packet>
```

Example 10: Receiving current navigation pane contents

Once obtained, the navigation can be customized as necessary. That done, the new custom navigation must be communicated back to Plesk. This is also done within the “create session” packet with the parameter `<navigation>` (see Example 8). Please note, only when a session in Plesk is created with this specially formed request will the custom navigation pane be available for the user. If the user logs in to Plesk directly, only the default navigation pane will be available.

Additionally, it is possible to specify the URL that the user must be taken to once he or she logs out of the Plesk CP. It is handled by means of the parameter `<back_url>`. This URL must be base64 encoded and sent with the “create session” packet as well.

Adding custom buttons

It is possible to add custom buttons at specific locations in Plesk CP thus enabling users to quickly access various resources or applications. This also can be done using Plesk XML API.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<packet version="1.4.1.0">
  <ui>
    <create-custombutton>
      <owner>
        <client-id>123321</client-id>
      </owner>
      <properties>
        <place>client</place>
        <url>URL</url>
        <text>CAPTION_TEXT</text>
      </properties>
    </create-custombutton>
  </ui>
</packet>
```

Example 11: Creating a custom button

This example shows how to form a request to add a custom button for the client with ID 123321. It can also be added for any other type of Plesk user by specifying the type of user within the `<owner>` section and supplying the user ID.

Section `<properties>` contains the necessary data for the new custom button. Element `<place>` specifies the location the custom button will be placed at (one can choose from `client`, `domain`, and `navigation`). Element `<url>` describes the URL the user will be taken to after clicking the custom button. Element `<text>` contains the custom button caption text as it should appear in GUI.

The response packet will contain the status of the creation operation, and the custom button ID in case of success or the error code in case of failure.