

# Gallant VPS API Documentation

## VPS Version 3.2

### API Version 1.2.2.2

March 15th 2012

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Change Log . . . . .	3
1.1.1	Version 1.1 . . . . .	3
1.1.2	Version 1.2 . . . . .	3
1.1.3	Version 1.2.1 . . . . .	3
1.1.4	Version 1.2.2 . . . . .	4
1.2	Terms used in this document . . . . .	5
1.3	Authentication . . . . .	5
1.4	Results . . . . .	5
1.4.1	Result Structure . . . . .	6
1.4.2	Result Codes . . . . .	6
<b>2</b>	<b>Services and Methods</b>	<b>6</b>
2.1	Account Service . . . . .	6
2.1.1	Create Account . . . . .	6
2.1.2	Remove account . . . . .	7
2.1.3	Set Password . . . . .	8
2.1.4	Set Account Username . . . . .	8
2.1.5	Set Account Owner Name . . . . .	9
2.1.6	Set Account UI language . . . . .	10
2.2	VPS service . . . . .	10
2.2.1	Activate VPS . . . . .	10
2.2.2	Deactivate VPS . . . . .	11
2.2.3	Set VPS Display Resolution . . . . .	12
2.2.4	Refresh VPS . . . . .	12
2.2.5	Set VPS Note . . . . .	13
2.2.6	Install Expert Advisor . . . . .	13
2.2.7	Apply Package to VPS . . . . .	14
2.2.8	Create a file upload session . . . . .	14

2.2.9	Upload a file chunk into an open file upload session . . . . .	15
2.2.10	Close a File Upload Session and save the file into the VPS . . .	16
2.2.11	Get VPS Reports List . . . . .	17
2.2.12	Get a VPS Report . . . . .	17
2.2.13	Set VPS Remote Access Password . . . . .	18
2.2.14	Set VPS Nickname . . . . .	18
2.3	Status Service . . . . .	19
2.3.1	Available Package List . . . . .	19
2.3.2	Available Expert Advisor List . . . . .	20
2.3.3	Pool Status . . . . .	20
2.3.4	Account Information . . . . .	21
2.3.5	Accounts List . . . . .	21
2.3.6	VPS Information . . . . .	22
2.3.7	VPS List . . . . .	23
2.3.8	Get Labels . . . . .	24
2.3.9	Get Available Languages . . . . .	24
2.4	Customer Service . . . . .	25
2.4.1	Account Information . . . . .	25
2.4.2	VPS List . . . . .	26
2.4.3	VPS Information . . . . .	27
2.4.4	Set Account Password . . . . .	27
2.4.5	Set VPS Display Resolution . . . . .	28
2.4.6	Refresh VPS . . . . .	29
2.4.7	Available Package List . . . . .	29
2.4.8	Available Expert Advisor List . . . . .	30
2.4.9	Install Expert Advisor . . . . .	30
2.4.10	Apply Package to VPS . . . . .	31
2.4.11	Get VPS Reports List . . . . .	31
2.4.12	Get a VPS Report . . . . .	32
2.4.13	Create a file upload session . . . . .	33
2.4.14	Upload a file chunk into an open file upload session . . . . .	33
2.4.15	Close a File Upload Session and save the file into the VPS . . .	34
2.4.16	Set Account Owner Name . . . . .	35
2.4.17	Set VPS Remote Access Password . . . . .	36
2.4.18	Set VPS Nickname . . . . .	36
2.4.19	Get Labels . . . . .	37
2.4.20	Get Available Languages . . . . .	37
2.4.21	Set Account UI language . . . . .	38

## 1 Introduction

This document will describe the detailed API for communication between Partners and the GallantVPS ERP framework. The API, designed based on SOAP specifications, will allow the retailer to perform tasks to manage customer accounts and VPSs.

## 1.1 Change Log

### 1.1.1 Version 1.1

- Spelling and grammar.
- Added *Last Activation Date* to *VPS Information* (2.3.6) and *VPS List* (2.3.7) Methods.
- Corrected name of AccountList method to AccountsList (2.3.5).

### 1.1.2 Version 1.2

- Added *Set Account Username* (2.1.4) method to Account Service
- VPS Note field was added to the *VPS Information* (2.3.6) and *VPS List* (2.3.7) methods of the Status Service.
- VPS VNC Password field was added to the *VPS Information* (2.3.6) and *VPS List* (2.3.7) methods of the Status Service.
- Corrected bug in *VPS List* (2.3.7) method of the Status Service. It used to push all account, now pushing only active accounts.
- *Set VPS Note* ( 2.2.5 on page 13) Method was added to the VPS Service
- Added *Customer Service* (2.4)with following methods:
  - Account Information
  - VPS List
  - VPS Information
  - Set Account Password
  - Refresh VPS
  - Available Package list
  - Set VPS Nickname
- Added VPS password to response of *Activate method* of the VPS service.
- Added File upload calls to the Customer & VPS services.

### 1.1.3 Version 1.2.1

- Added MT4 Report access to VPS, and Customer Services.
- Added VNC Port & VPS Nickname to VPS description in:
  - Status Service VPS List method ( 2.3.7 on page 23)
  - Status Service VPS Information method ( 2.3.6 on page 22).

- Added VPS Nickname to VPS description in:
  - Customer Service VPS List method( 2.4.2 on page 26) .
  - Customer Service VPS Information method( 2.4.3 on page 27) method.
- Added the following methods to the Customers Service:
  - Set VPS Display Resolution
  - Available Expert Advisor List
  - Apply Package to VPS
  - Install Expert Advisor
  - Set Account Owner Name
  - Set VPS Remote Access Password
- Added instruction on how to deal with non URL safe characters in account name.

#### **1.1.4 Version 1.2.2**

- Added the getLabels method to the Customer Service and the Status.
- Deprecated the Pool Status method of the Status service: Once your VPS setup has been upgraded to version 3.2 your account will have been included into our geolocation subsystem. This subsystem will provision new accounts into your pool on demand and will render this method useless. After the upgrade you will always see 0 Inactive accounts in the Output. This method is deprecated and will be removed from the API in version 1.3.
- Add the getAvailableLanguages Method to the Status and Customer service.
- Added the UI Language Field to the:
  - AccountInformation & AccountList Methods of the Status service
  - AccountInformation Method of the Customer service
- Added the setLanguage method to the Account Service and the Customer service.
- Added the account resolution to the output of the vpsInformation and vpsList methods of the Customer service.

## 1.2 Terms used in this document

**Partner** The organization offering VPS services through Gallant VPS.

**Account** A GallantVPS account allows the end user to access and manipulate one or more VPS(s).

**VPS** A unique instance of the MetaTrader platform hosted on the GallantVPS platform. A VPS may be in one of three statuses:

- Active: where the VPS is accessible to an end user through their account.
- Inactive: VPSs that are pooled for later use and are otherwise unavailable.
- Pending: which only occurs in the case of a malfunction during account manipulation.

**VPS Pool** A group of inactive VPSs that have been assigned to the Partner. These may be activated and provisioned at any time through the manager UI or the API. **Once your VPS setup has been upgraded to version 3.2 your account will have been included into our geolocation subsystem. This subsystem will provision new accounts into your pool on demand and will render the pool concept useless.**

**VPS Package** A package refers to a set of files and programs that are installed into a new VPS or re-installed into an existing VPS, including a specific MetaTrader, background image, activation icon used to relaunch the MetaTrader, and optional Expert Advisors.

**Activating a VPS** Implies taking an arbitrary VPS from the inactive VPS pool, applying a VPS Package to it, starting the VPS, and assigning it to an Account.

## 1.3 Authentication

All communication with the API will be done over SSL, and will require a username and password as part of its input. These credentials will be provided once the Partner account is activated. Please note that authentication for the *Customer Service* (2.4) will be done via the end user credentials.

Any communication attempt that does not pass authentication will return an Authentication Fault.

## 1.4 Results

Most methods will return an Array of Strings on the structure defined in section 1.4.1. However the methods provided by the Status Service described (in section 2.3 on page 19) will have specific result structures that will be described in each method separately.

### 1.4.1 Result Structure

```
Array{
  "Result Code : Numeric code from the list below",
  "Result : Result or error description from the list below",
  "Result Description : Detailed result or error description",
}
```

### 1.4.2 Result Codes

- 0 Success
- 1 Authentication Error
- 2 Account/VPS does not exist
- 3 Value does not match request
- 4 Pool size insufficient
- 5 Unauthorized access attempt
- 6 System Error

## 2 Services and Methods

### 2.1 Account Service

**URL** <https://vps3.gallantvps.com/account/vps/>

**WSDL URL** <https://vps3.gallantvps.com/api/account/service.wsdl>

Methods of this service will allow you to manage user accounts.

#### 2.1.1 Create Account

**Method Name** create

**Description** This method will attempt to create a new account and activate one VPS from the pool for it. The system will check the pool size to make sure the Partner has an available VPS to be activated for this account. If there are none it will NOT create the account and return the appropriate result code. The geolocation subsystem will generate new accounts into the pool on demand. Please note that this call will take between forty to seventy seconds to complete depending on the number of EAs you choose to install. Please configure your SOAP client accordingly.

## Input

**username:** String - API authentication token as described in section 1.3.

**password:** String - API authentication token as described in section 1.3.

**accountName:** String - The end users Email Address, which will be used as the Username to access the account. Must be a string in valid email address format<sup>12</sup>.

**firstName:** String - The end user's first name. May be an empty string.

**lastName:** String - The end user's last name. Must contain a value.

**packageId:** Int - The ID of the Package to be installed on the VPS that will be activated for the account. You can find the list of available packages and their ID by contacting Gallant VPS or calling the *Available Package List* method of the *Status Service* as described in section 2.3.1 on page 19. Please note that the list provided by the API is always up to date.

**eaIds:** String - The IDs of Expert Advisors you wish to install on the VPS that will be activated for the account. You may send an empty string or the string "None" if you choose not to install any experts on the VPS. To install more than one Expert please add all the IDs to the string separated by a pipe ("|"). You can find the list of available Experts and their ID by contacting Gallant VPS or calling the *Available Experts List* method of the *Status Service* as described in section 2.3.2 on page 20. Please note that the list provided by the API is always up to date. To add additional Experts to your Library please contact Gallant VPS.

## Output

This method will return its output in the standard structure as described in section 1.4.1 on the preceding page.

If completed successfully the description section of the output will include the new Account Name and Password in this format:

"Account [Account Name] was created successfully. Password was set to : [Password]"

### 2.1.2 Remove account

**Method Name** remove

**Description** This method allows the Partner to remove an account and disable all its VPSs. Please note that all the information belonging to the account and its VPSs will be lost.

---

<sup>1</sup>As defined in <http://tools.ietf.org/html/rfc2142>

<sup>2</sup>Please note that due to SOAPs nature the provided string must be URL safe. To use characters like '+' please consider url encoding this string. (e.g Replace '+' with '%2B'.) Please see [http://www.w3schools.com/tags/ref\\_urlencode.asp](http://www.w3schools.com/tags/ref_urlencode.asp) for details.

### **Input**

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**accountName:** String - Account name is the email address used as the Username for the account.

### **Output**

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully the description section of the output will include the Account Name in this format:

“Account [Account Name] was successfully removed and its VPS(s) have been deactivated and returned to your pool”

#### **2.1.3 Set Password**

**Method Name** password

**Description** This method allows the Partner to reset the password of an active account.

### **Input**

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**accountName:** String - The email address used as the Username for the account.

**newPassword:** String - New password may be a string containing the value to be assigned as the new Account password or an empty string. In which case the system will generate a random password. Password must be at least six characters long.

### **Output**

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully the description section of the output will include the new Account Name and password in this format:

“The password for account [Account Name] was set to : [Password]”

#### **2.1.4 Set Account Username**

**Method Name** accountName

**Description** This method allows the Partner to reset the username (email address) associated with the account.

### Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**accountName:** String - The email address used as the Username for the account<sup>3</sup>.

**newAccountName:** String - The new email address to be used as the Username for the account. The format of the email addresses must comply with in RFC 5322<sup>4</sup>(sections 3.2.3 and 3.4.1) and by RFC 5321<sup>5</sup>.

### Output

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully the description section of the output will include the new Account Name and Password in this format:

“The account formerly know as [Old Account Name] is now know as : [New Account Name]”

#### 2.1.5 Set Account Owner Name

**Method Name** setOwnerName

**Description** This method allows the Partner to reset the owner name of with the account.

### Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**accountName:** String - The email address used as the Username for the account.

**firstName:** String - The accounts owner first name.

**lastName:** String - The accounts owner last name.

### Output

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully be the description should be:

“The accounts owner name has been set.”

<sup>3</sup>Please note that due to SOAPs nature the provided string must be URL safe. To use characters like '+' please consider url encoding this string. (e.g Replace '+' with '%2B'.) Please see [http://www.w3schools.com/tags/ref\\_urlencode.asp](http://www.w3schools.com/tags/ref_urlencode.asp) for details.

<sup>4</sup><http://www.ietf.org/rfc/rfc5322.txt>

<sup>5</sup><http://www.ietf.org/rfc/rfc5321.txt>

### 2.1.6 Set Account UI language

**Method Name** setLanguage

**Description** This method allows the retailer to set the accounts UI Language to one of the available values provided by the getAvailableLanguages method ( 2.3.9 on page 24). Please note that the VPS UI language relies on available labels value as provided by the getLables method ( 2.3.8 on page 24). If the label value in the set language is missing the label will default to English.

#### Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**accountName:** String - The email address used as the Username for the account.

**languageCode:** String - The preferred languageCode as provided by getAvailableLanguages method ( 2.3.9 on page 24) .

#### Output

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully be the description should be:

“The account language was set to [Language Name].”

## 2.2 VPS service

**URL** <https://vps3.gallantvps.com/api/vps/>

**WSDL URL** <https://vps3.gallantvps.com/api/vps/service.wsdl>

Methods of this service will allow you to manage the VPSs assigned to you.

### 2.2.1 Activate VPS

**Method Name** activate

**Description** This method will activate one arbitrary VPS from the pool and attach it to an existing account. The system will check your pool size to make sure you have an available VPS to be activated. The geolocation subsystem will generate new accounts into the pool on demaned.

Please note that this call will take between forty to seventy seconds to complete depending the number of EAs selected for installation. Please configure your SOAP client accordingly.

### Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**accountName:** String - The email address used as the Username for the selected Account.

**packageId:** Int - The ID of the Package to be installed on the new VPS. You can find the list of available packages and their IDs by contacting Gallant VPS or calling the *Available Package List* method of the *Status Service* as described in section 2.3.1 on page 19. Please note that the list provided by the API is always up to date.

**eaIds:** String - The IDs of Expert Advisors you wish to install on the new VPS. You may send an empty string or the string "None" if you chose not to install any experts on the VPS. To install more than one Expert please add all the IDs to the string separated by a pipe ("|"). You can find the list of available Experts and their IDs by contacting Gallant VPS or calling the *Available Experts List* method of the *Status Service* as described in section 2.3.2 on page 20. Please note that the list provided by the API is always up to date. To add additional Experts to your Library please contact Gallant VPS.

### Output

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully, the description section of the output will include the Account Name and new VPS ID in this format:

"VPS [VPS ID] was activated and attached to Account [Account Name]."

### 2.2.2 Deactivate VPS

**Method Name** deactivate

**Description** This method will return an Active VPS into the VPS Pool to be recycled. If this VPS is the only VPS assigned to an account the method will return a 3. "Value does not match request" return code as the system does not allow an account with no VPSs. You should remove the account itself by using the Remove Account method described in section 2.1.2 on page 7.

### Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**vpsId:** String - The ID for the VPS you wish to deactivate

## Output

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully, the description section of the output will include the VPS ID in this format:

“VPS [VPS ID] was successfully deactivated and returned to your pool”

### 2.2.3 Set VPS Display Resolution

**Method Name** setDisplay

**Description** This method will set the display resolution for the specified VPS and then restart the VPS to apply the new settings.

## Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**vpsId:** String - The ID for the VPS you wish to modify

**newRes:** String - The new resolution value in “[Width in Pixels]x[height in pixels]” format. The only allowed values are: “800x600”, “1024x768” and “1280x1024”.

## Output

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully, the description section of the output will include the VPS ID in this format:

“VPS [VPS ID] display size was reset successfully”

### 2.2.4 Refresh VPS

**Method Name** refresh

**Description** This method will restart a VPS, much like rebooting a computer, resolving most MetaTrader issues that may arise due to misconfiguration or system errors.

## Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**vpsId:** String - The ID for the VPS you wish to refresh

## Output

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully, the description section of the output will include the VPS ID in this format:

“VPS [VPS ID] was successfully refreshed”

### 2.2.5 Set VPS Note

**Method Name** setNote

**Description** This method will update the internal note associated with the VPS. This note will never be visible to the ens user. Any former value of the VPS Note will be overwritten and cannot be retrieved.

## Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**vpsId:** String - The ID for the VPS you wish to modify.

**note:** String - The note to be attached to the account. Maximum length is 255 characters.

## Output

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully, the description section of the output will be: “VPS note was set successfully”

### 2.2.6 Install Expert Advisor

**Method Name** installEa

**Description** This method will install an EA specified by the EA ID as provide by the *Available Expert Advisor List* method 2.4.8 on page 30 on the specified VPS. After a successful installation the VPS will be restart to apply the changes.

## Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**vpsId:** String - The ID for the VPS you wish to modify.

**eaId:** String - The ID for the Expert Advisor to be installed.

## Output

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully, the description section of the output will include the VPS ID in this format:

“An EA was successfully installed on VPS [VPS ID]”

### 2.2.7 Apply Package to VPS

**Method Name** applyPackage

**Description** This method will apply the selected package identified by the Package ID as provided by the *Package List* method of the status service 2.3.1 on page 19 on the VPS. All the information inside the VPS will be destroyed during this process including account info and any installed files the user may have uploaded.

## Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**vpsId:** String - The ID for the VPS you wish to modify.

**packageId:** String - The ID for the Package to be applied.

## Output

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully the description section of the output will include the VPS ID in this format:

“A new package was successfully applied to VPS [VPS ID]”

### 2.2.8 Create a file upload session

**Method Name** CreateFileUploadSession

**Description** This method will initiate a file upload session that will allow the upload of files of arbitrary size into the VPS. On success it will return a session token that will be used to upload the file using one or more calls to the uploadFileChunk at 2.2.9 on the next page. A session token is a case sensitive alphanumeric string of 128 characters. Once the session is in place you may upload a file by brack- ing it into chunks and uploading those chunks into the session using the up- loadFileChunk method ( 2.2.9 on the following page). File upload session will survive up to 24 hours or until closed with the close session call ( 2.2.10 on page 16).

### Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**vpsId:** String - The ID for the VPS you wish to modify.

### Output

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully, the description section of the output will include the new File Upload Session Token in the following format : “File Upload session initiated successfully; session token was set to :[SESSION TOKEN]”. Please note that there are no spaces before or after the token.

### 2.2.9 Upload a file chunk into an open file upload session

**Method Name** uploadFileChunk

**Description** Once a new file upload session is created using the CreateFileUploadSession ( 2.2.8 on the previous page) the target file must be broken into reasonable size chunks. Each chunk must be *Modified Base64 For URLs* encoded<sup>6 7</sup> and then feed into this method. For instructions on how to implement *Modified Base64 For URLs* please review foot notes 6 and 7. Chunk size should not exceed 1MB. You can make as many calls of this methods into a sessions as you need.

### Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**vpsId:** String - The ID for the VPS you wish to modify.

**sessionToken:** String - A File Upload session token as provided by the CreateFileUploadSession method ( 2.2.8 on the preceding page).

**chunk:** String - A part of the content of the file that is being uploaded as a *Modified Base64 For URLs* encoded string as described in foot notes 6 and 7.

---

<sup>6</sup>Please note that unlike any other use of base64 encoding described in this document this method requires *Modified Base64 For URLs* encoding. Where the '+' and '/' characters of standard Base64 are respectively replaced by '-' and '\_'. In certain languages like Python the base64 library will have a urls safe encoding and decoding methods, those should work fine. In other languages, like PHP, ASP.net and C#, after encoding the chunk into base64 you must replace all '+' and '/' characters with '-' and '\_' respectively.

<sup>7</sup>[http://en.wikipedia.org/wiki/Base64#URL\\_applications](http://en.wikipedia.org/wiki/Base64#URL_applications)

## Output

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully, the description section of the output will be: “The file chunk was added into the session successfully”. Please note that if any of the calls to this methods fails the session will be close and the information uploaded will be lost. Please create a new session and start again. If these issues persist consider lowering the chunk size.

### 2.2.10 Close a File Upload Session and save the file into the VPS

**Method Name** closeFileUploadSession

**Description** This method will take all the information uploaded into a File Upload Session and deposit it as a file by the provided name in a folder inside the selected VPS MetaTrader folder. The selection of the sub folder is set by the provided File Type. The supported types are:

File Type Value	Destination subfolder inside the MetaTrader installation
EA	experts/
Indicator	experts/indicators/
Include	experts/include/
Preset	experts/presets/
Script	experts/scripts/
Template	templates/
Config	config/
DLL	experts/libraries/
Files	experts/files/
Profiles	profiles/
MetaTrader	The meta trader folder itself

Table 1: Files Types and destinations

## Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**vpsId:** String - The ID for the VPS you wish to modify.

**sessionToken:** String - A File Upload session token as provided by the CreateFileUploadSession method ( 2.2.8 on page 14).

**fileType:** String - The type of file being uploaded, allowed valued are described in table 1.

**fileName:** String - The name of file being uploaded.

### Output

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully, the description section of the output will be: “The file was uploaded successfully”

### 2.2.11 Get VPS Reports List

**Method Name** reports

**Description** This method will return all the approved reports generated by meta or by specific expert advisors. The approval process at this point allows only meta generated html reports and some per-approved CSV files. To add new reports to the per-approved list please contact GallantVPS support.

### Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**vpsId:** String - The ID for the VPS you wish to modify.

### Output

If there are no reports this method will return an empty Array of Strings, other wise it will return an Array of Strings in the following format:

```
Array{
  '[Report URI/Report Name]',
  '[Another Report URI/Another Report Name]',
  ...
}
```

### 2.2.12 Get a VPS Report

**Method Name** report

**Description** This method will return a requested VPS report specified by the Report Name as provided by the *Get VPS Reports List* method 2.2.11. The report will be returned as a base64 encoded string to allow safe transport.

**Input**

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**vpsId:** String - The ID for the VPS you wish to modify.

**reportName:** String - The name and path of the report to be returned.

**Output**

The method will return a String containing the Base 64 encoded content of the report. Note that all dependent files of a report will have already been included into the content.

**2.2.13 Set VPS Remote Access Password**

**Method Name** setRemoteAccessPassword

**Description** This method will change the remote access password to the VPS. Not to be confused with the account password. The new password is expected to be base64 encoded for safe transport.

**Input**

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**vpsId:** String - The ID for the VPS you wish to modify.

**newPass:** String - The new remote access password encoded in base64.

**Output**

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully, the description section of the output will be:  
“The VPS password was set successfully.”

**2.2.14 Set VPS Nickname**

**Method Name** setNickname

**Description** This method will change the nickname assigned to the VPS.

### Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**vpsId:** String - The ID for the VPS you wish to modify.

**newName:** String - The new nickname for the VPS.

### Output

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully, the description section of the output will be:

“Nickname was set successfully.”

## 2.3 Status Service

**URL** <https://vps3.gallantvps.com/api/status/>

**WSDL URL** <https://vps3.gallantvps.com/api/status/service.wsdl>

The methods in this service will provide you with information about the status of your Partner Account and VPS Pool. Each of these methods will return the results in their own format as they return different data from the other services. Please note that all errors will be returned in the standard structure as described in section 1.4.1 on page 6.

### 2.3.1 Available Package List

**Method Name** packages

**Description** This method will return a list of all the packages available to the Partner for installation on their VPSs.

### Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

### Output

Upon successful completion, this method will return an Array of Arrays of strings. Each Array will include the Package ID and Package Name as its only members.

```
Array{
  Array{ '[Package ID]', '[Package Name]' },
  Array{ '[Another Package ID]', '[Another Package Name]' },
  ...
}
```

### 2.3.2 Available Expert Advisor List

**Method Name** experts

**Description** This method will return a list of all the expert advisors available to the Partner for installation on their VPSs from the EA Library. If you want to modify this list please contact Gallant VPS.

#### Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

#### Output

Upon successful completion, this method will return an Array of Arrays of Strings. Each Array will include the Expert ID and Expert Name as its only members.

```
Array{  
  Array{ '[EA ID]', '[EA Name]' },  
  Array{ '[Another EA ID]', '[Another EA Name]' },  
  ...  
}
```

### 2.3.3 Pool Status

**Method Name** pool

**Description** **Once your VPS setup has been upgraded to version 3.2 your account will have been included into our geolocation subsystem. This subsystem will provision new accounts into your pool on demand and will render this method useless. After the upgrade you will always see 0 Inactive accounts in the Output. This method is deprecated and will be removed from the API in version 3.3.** This method will return the number of accounts currently active, the number of Active VPSs assigned to them, and the number of inactive VPSs available in the pool assigned to the Partner.

#### Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

#### Output

On successful completion this method will return an Array of Arrays of Strings in the following structure:

```
Array {  
  Array { "Accounts", "[NUMBER OF ACCOUNTS]" },
```

```
Array {"Active VPSs","[NUMBER OF ACTIVE VPSS]"},
Array {"Inactive VPSs","[NUMBER OF INACTIVE VPSS]"}
}
```

### 2.3.4 Account Information

**Method Name** accountInformation

**Description** This method will return a detailed description of the requested account.

#### Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**accountName:** String - Account name is the email address used as the Username for the account<sup>8</sup>.

#### Output

On successful completion this method will return an Array of Strings in the following structure:

```
Array {
  [Account Name e.g. "john.doe@email.com"],
  [First Name e.g. "John"],
  [Last Name e.g. "Doe"],
  [Time since last log-in in "D days, HH:MM ago" format e.g. "15 Days, 20:47 ago"],
  [Comma Separated List of all the VPS IDs attached to this account e.g. "1195,1132,.."],
  [Language Code as provided by the getAvailableLanguages method ( 2.3.9 on page 24) e.g "en"]
}
```

### 2.3.5 Accounts List

**Method Name** accountsList

**Description** This method will return the detailed description of all of the accounts belonging to the Partner.

#### Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

<sup>8</sup>Please note that due to SOAPs nature the provided string must be URL safe. To use characters like '+' please consider url encoding this string. (e.g Replace '+' with '%2B'.) Please see [http://www.w3schools.com/tags/ref\\_urlencode.asp](http://www.w3schools.com/tags/ref_urlencode.asp) for details.

## Output

On successful completion this method will return an Array of Array of Strings in the following structure:

```
Array{
  Array {
    [Account Name e.g. "john.doe@email.com"],
    [First Name e.g. "John"],
    [Last Name e.g. "Doe"],
    [Time since last log-in in "D days, HH:MM ago" format e.g. "15 Days, 20:47 ago"],
    [Comma Separated List of all the VPS IDs attached to this account e.g. "1195,1132,.."],
    [Language Code as provided by the getAvailableLanguages method ( 2.3.9 on page 24) e.g "en"]
  },
  Array {
    [Account Name e.g. "Jane.doe@email.com"],
    [First Name e.g. "Jane"],
    [Last Name e.g. "Doe"],
    [Time since last log in in "D days, HH:MM ago" format e.g. "15 Days, 20:47 ago"],
    [Comma Separated List of all the VPS IDs attached to this account e.g. "1195,1132,.."],
    [Language Code as provided by the getAvailableLanguages method ( 2.3.9 on page 24) e.g "en"]
  }, ...
}
```

### 2.3.6 VPS Information

**Method Name** vpsInformation

**Description** This method will return a detailed description of the requested VPS.

#### Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**vpsId:** String - The ID of the VPS you are requesting

#### Output

On successful completion this method will return an Array of Strings in the following structure:

```
Array {
  [Account ID e.g. "9999"],
  [Assign to Account Name e.g. "john.doe@email.com"]
}
```

```

    [Package e.g. "White"],
    [Data center where the VPS is located e.g "DFW" for Dallas, TX or "LON" for
London, UK]
    [Last Modified date "D days, HH:MM ago" format e.g. "15 Days, 20:47 ago"],
    [Last Activated date "YYYY-MM-DD" format e.g. "2011-04-17"],
    [VPS Note e.g. "Trading Account 1111111" Or "None" if no note is in place],
    [VPS VNC Password e.g. "SecretPa55W0rD"],
    [VPS VNC Port e.g. "3988"],
    [VPS Nickname]
}

```

### 2.3.7 VPS List

**Method Name** vpsList

**Description** This method will return a detailed description of all of the VPSs belonging to the Partner.

#### Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

#### Output

On successful completion this method will return an Array of Array of Strings in the following structure:

```

Array{
  Array {
    [Account ID e.g. "9999"],
    [Assign to Account Name e.g. "john.doe@email.com"]
    [Package e.g. "White"],
    [Data center where the VPS is located e.g "DFW" for Dallas, TX or "LON" for
London, UK],
    [Last Modified date "D days, HH:MM ago" format e.g. "15 Days, 20:47 ago"],
    [Last Activated date "YYYY-MM-DD" format e.g. "2011-04-17"],
    [VPS Note e.g. "Trading Account 1111111" Or "None" if no note is in place],
    [VPS VNC Password e.g. "SecretPa55W0rD"],
    [VPS VNC Port e.g. "3988"],
    [VPS Nickname]
  },
  Array {
    [Account ID e.g. "9999"],
    [Assign to Account Name e.g. "john.doe@email.com"]
    [Package e.g. "White"],

```

```

    [Data center where the VPS is located e.g "DFW" for Dallas, TX or "LON" for
    London, UK],
    [Last Modified date "D days, HH:MM ago" format e.g. "15 Days, 20:47 ago"],
    [Last Activated date "YYYY-MM-DD" format e.g. "2011-04-17"],
    [VPS Note e.g. "Trading Account 1111111" Or "None" if no note is in place],
    [VPS VNC Password e.g. "SecretPa55W0rD"],
    [VPS Nickname]
  }, ...
}

```

### 2.3.8 Get Labels

**Method Name** getLabels

**Description** This method will return an Array of Arrays of Strings containing the labels as set via Manage Labels UI by the manager of the retailer..

#### Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

#### Output

On successful completion this method will return an Array of Arrays of Strings in the following structure:

```

  Array {
    Array {"[Label Name]","[Label Value]"},
    Array {"[Another Label Name]","[Another Label Value]"},
    ...
  }

```

### 2.3.9 Get Available Languages

**Method Name** getAvailableLanguages

**Description** This method will return an Array of Arrays of Strings containing all the languages supported by the GallantVPS system in Language Code, Language Name pairs.

#### Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

## Output

On successful completion this method will return an Array of Arrays of Strings in the following structure:

```
Array {
  Array {
    Array {"[Language Code e.g en]","[Language Name e.g English]"},
    Array {"[Another Language Code ]","[Another Language Name]"},
    ...
  }
}
```

## 2.4 Customer Service

**URL** <https://vps3.gallantvps.com/account/customer/>

**WSDL URL** <https://vps3.gallantvps.com/api/customer/service.wsdl>

Methods of this service will allow end user level access to the API. They are made available for creation of third party customer application and every interaction with the API will be subject to end user level limitation. As such the user name and password required by this service are the customer's account name (Email address<sup>9</sup>) and password. As our system is made available to many retailers you must first obtain and receive a retailer code (short string) to be able to use this service.

### 2.4.1 Account Information

**Method Name** accountInformation

**Description** This method will return a detailed description of the requesting end user account.

#### Input

**username:** String - The end user's account name (email address).

**password:** String - The end user's password.

**retailerCode:** String - The retailer code assigned to the users retailer.

#### Output

On successful completion this method will return an Array of Strings in the following structure:

```
Array {
  [Account Name e.g. "john.doe@email.com"],
  [First Name e.g. "John"],
}
```

---

<sup>9</sup>Please note that due to SOAPs nature the provided string must be URL safe. To use characters like '+' please consider url encoding this string. (e.g Replace '+' with '%2B'.) Please see [http://www.w3schools.com/tags/ref\\_urlencode.asp](http://www.w3schools.com/tags/ref_urlencode.asp) for details.

```
[Last Name e.g. "Doe"],
[Time since last log-in in "D days, HH:MM ago" format e.g. "15 Days, 20:47
ago"],
[Comma Separated List of all the VPS IDs attached to this account e.g. "1195,1132,.."],
}
```

## 2.4.2 VPS List

**Method Name** vpsList

**Description** This method will return a detailed description of all of the VPSs belonging to the end user.

### Input

**username:** String - The end users account name (email address).

**password:** String - The end users password.

**retailerCode:** String - The retailer code assigned to the users retailer.

### Output

On successful completion this method will return an Array of Array of Strings in the following structure:

```
Array{
  Array {
    [Account ID e.g. "9999"],
    [Assign to Account Name e.g. "john.doe@email.com"]
    [Package e.g. "White"],
    [VPS Server IP Address]
    [Last Modified date "D days, HH:MM ago" format e.g. "15 Days, 20:47 ago"],
    [Last Activated date "YYYY-MM-DD" format e.g. "2011-04-17"],
    [VPS Note e.g. "Trading Account 111111" Or "None" if no note is in place],
    [VPS VNC Password e.g. "SecretPa55W0rD"],
    [VPS VNC Port e.g. "3988"],
    [VNC Resolution]
  },
  Array {
    [Account ID e.g. "9999"],
    [Assign to Account Name e.g. "john.doe@email.com"]
    [Package e.g. "White"],
    [VPS Server IP Address]
    [Last Modified date "D days, HH:MM ago" format e.g. "15 Days, 20:47 ago"],
    [Last Activated date "YYYY-MM-DD" format e.g. "2011-04-17"],
    [VPS Note e.g. "Trading Account 111111" Or "None" if no note is in place],
    [VPS VNC Password e.g. "SecretPa55W0rD"],
```

```
[VPS VNC Port e.g. "3988"],  
[VNC Resolution]  
}, ...  
}
```

### 2.4.3 VPS Information

**Method Name** vpsInformation

**Description** This method will return a detailed description of the requested VPS.

#### Input

**username:** String - The end users account name (email address).

**password:** String - The end users password.

**retailerCode:** String - The retailer code assigned to the users retailer.

**vpsId:** String - The ID of the VPS you are requesting

#### Output

On successful completion this method will return an Array of Strings in the following structure:

```
Array {  
  [Account ID e.g. "9999"],  
  [Assign to Account Name e.g. "john.doe@email.com"]  
  [Package e.g. "White"],  
  [VPS Server IP Address]  
  [Last Modified date "D days, HH:MM ago" format e.g. "15 Days, 20:47 ago"],  
  [Last Activated date "YYYY-MM-DD" format e.g. "2011-04-17"],  
  [VPS Note e.g. "Trading Account 111111" Or "None" if no note is in place],  
  [VPS VNC Password e.g. "SecretPa55W0rD"],  
  [VPS VNC Port e.g. "3899"],  
  [VNC Resolution]  
}
```

### 2.4.4 Set Account Password

**Method Name** password

**Description** This method allows the end user to reset the password of his account.

### Input

**username:** String - The end users account name (email address).

**password:** String - The end users password.

**retailerCode:** String - The retailer code assigned to the users retailer.

**newPassword:** String - New password may be a string containing the value to be assigned as the new Account password or an empty string. In which case the system will generate a random password. Password must be at least six characters long.

### Output

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully the description section of the output will include the new Account Name and password in this format:

“The password for account [Account Name] was set to : [Password]”

## 2.4.5 Set VPS Display Resolution

**Method Name** setDisplay

**Description** This method will set the display resolution for the specified VPS and then restart the VPS to apply the new settings.

### Input

**username:** String - The end users account name (email address).

**password:** String - The end users password.

**retailerCode:** String - The retailer code assigned to the users retailer.

**vpsId:** String - The ID for the VPS you wish to modify.

**newRes:** String - The new resolution value in “[Width in Pixels]x[height in pixels]” format. The only allowed values are: “800x600”, “1024x768” and “1280x1024”.

### Output

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully, the description section of the output will include the VPS ID in this format:

“VPS [VPS ID] display size was reset successfully”

## 2.4.6 Refresh VPS

**Method Name** refresh

**Description** This method will restart a VPS, much like rebooting a computer, resolving most MetaTrader issues that may arise due to misconfiguration or system errors.

### Input

**username:** String - The end users account name (email address).

**password:** String - The end users password.

**retailerCode:** String - The retailer code assigned to the users retailer.

**vpsId:** String - The ID for the VPS you wish to refresh

### Output

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully, the description section of the output will include the VPS ID in this format:

“VPS [VPS ID] was successfully refreshed”

## 2.4.7 Available Package List

**Method Name** packages

**Description** This method will return a list of all the packages available to the end user.

### Input

**username:** String - The end users account name (email address).

**password:** String - The end users password.

**retailerCode:** String - The retailer code assigned to the users retailer.

### Output

Upon successful completion, this method will return an Array of Arrays of strings. Each Array will include the Package ID and Package Name as its only members.

```
Array{
  Array{
    Array{ '[Package ID]', '[Package Name]' },
    Array{ '[Another Package ID]', '[Another Package Name]' },
    ...
  }
}
```

## 2.4.8 Available Expert Advisor List

**Method Name** experts

**Description** This method will return a list of all the expert advisors available to the Partner for installation on their VPSs from the EA Library. If you want to modify this list please contact Gallant VPS.

### Input

**username:** String - The end users account name (email address).

**password:** String - The end users password.

**retailerCode:** String - The retailer code assigned to the users retailer.

### Output

Upon successful completion, this method will return an Array of Arrays of Strings. Each Array will include the Expert ID and Expert Name as its only members.

```
Array{  
  Array{ '[EA ID]', '[EA Name]' },  
  Array{ '[Another EA ID]', '[Another EA Name]' },  
  ...  
}
```

## 2.4.9 Install Expert Advisor

**Method Name** installEa

**Description** This method will install an EA specified by the EA ID as provide by the *Available Expert Advisor List* method 2.4.8 on the specified VPS. After a successful installation the VPS will be restart to apply the changes.

### Input

**username:** String - The end users account name (email address).

**password:** String - The end users password.

**retailerCode:** String - The retailer code assigned to the users retailer.

**vpsId:** String - The ID for the VPS you wish to refresh.

**eaId:** String - The ID for the Expert Advisor to be installed.

## Output

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully, the description section of the output will include the VPS ID in this format:

“An EA was successfully installed on VPS [VPS ID]”

### 2.4.10 Apply Package to VPS

**Method Name** applyPackage

**Description** This method will apply the selected package identified by the Package ID as provided by the *Package List* method 2.4.7 on page 29 onto the VPS. All the information inside the VPS will be destroyed during this process including account info and any installed files the user may have uploaded.

## Input

**username:** String - The end users account name (email address).

**password:** String - The end users password.

**retailerCode:** String - The retailer code assigned to the users retailer.

**vpsId:** String - The ID for the VPS you wish to refresh.

**packageId:** String - The ID for the Package to be applied.

## Output

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully the description section of the output will include the VPS ID in this format:

“A new package was successfully applied to VPS [VPS ID]”

### 2.4.11 Get VPS Reports List

**Method Name** reports

**Description** This method will return all the approved reports generated by meta or by specific expert advisors. The approval process at this point allows only meta generated html reports and some pre approved CSV files. To add new reports to the pre approved list please contact GallantVPS support.

### Input

**username:** String - The end users account name (email address).

**password:** String - The end users password.

**retailerCode:** String - The retailer code assigned to the users retailer.

**vpsId:** String - The ID for the VPS you wish to modify.

### Output

If there are no reports this method will return an empty Array of Strings, other wise it will return an Array of Strings in the following format:

```
Array{
  '[Report URI/Report Name]',
  '[Another Report URI/Another Report Name]',
  ...
}
```

#### 2.4.12 Get a VPS Report

**Method Name** report

**Description** This method will return a requested VPS report specified by the Report Name as provided by the *Get VPS Reports List* method 2.4.11 on the previous page. The report will be returned as a base64 encoded string to allow safe transport.

### Input

**username:** String - The end users account name (email address).

**password:** String - The end users password.

**retailerCode:** String - The retailer code assigned to the users retailer.

**vpsId:** String - The ID for the VPS you wish to modify.

**reportName:** String - The name and path of the report to be returned.

### Output

The method will return a String containing the Base 64 encoded content of the report. Note that all dependent files of a report will have already been included into the content.

### 2.4.13 Create a file upload session

**Method Name** CreateFileUploadSession

**Description** This method will initiate a file upload session that will allow the upload of files of arbitrary size into the VPS. On success it will return a session token that will be used to upload the file using one or more calls to the uploadFileChunk at 2.4.14. A session token is a case sensitive alphanumeric string of 128 characters. File upload session will survive up to 24 hours or until closed with the close session call ( 2.4.15 on the next page).

#### Input

**username:** String - The end users account name (email address).

**password:** String - The end users password.

**retailerCode:** String - The retailer code assigned to the users retailer.

**vpsId:** String - The ID for the VPS you wish to modify.

#### Output

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully, the description section of the output will include the new File Upload Session Token in the following format : “File Upload session initiated successfully; session token was set to :[SESSION TOKEN]”. Please note that there are no spaces before or after the token.

### 2.4.14 Upload a file chunk into an open file upload session

**Method Name** uploadFileChunk

**Description** Once a new file upload session is created using the CreateFileUpload-Session ( 2.4.13) the target file must be broken into reasonable size chunks. Each chunk must be *Modified Base64 For URLs* encoded<sup>10 11</sup> and then feed into this method. For instructions on how to implement *Modified Base64 For URLs* please review foot notes 10 and 11. Chunk size should not exceed 1MB. You can make as many calls of this methods into a sessions as you need.

---

<sup>10</sup>Please note that unlike any other use of base64 encoding described in this document this method requires *Modified Base64 For URLs* encoding. Where the '+' and '/' characters of standard Base64 are respectively replaced by '-' and '\_'. In certain languages like Python the base64 library will have a urls safe encoding and decoding methods, those should work fine. In other languages, like PHP, ASP.net and C#, after encoding the chunk into base64 you must replace all '+' and '/' characters with '-' and '\_' respectively.

<sup>11</sup>[http://en.wikipedia.org/wiki/Base64#URL\\_applications](http://en.wikipedia.org/wiki/Base64#URL_applications)

### **Input**

**username:** String - The end users account name (email address).

**password:** String - The end users password.

**retailerCode:** String - The retailer code assigned to the users retailer.

**vpsId:** String - The ID for the VPS you wish to modify.

**sessionToken:** String - A File Upload session token as provided by the CreateFileUploadSession method ( 2.4.13 on the preceding page).

**chunk:** String - A part of the content of the file that is being uploaded as a *Modified Base64 For URLs* encoded string as described in foot notes 10 on the previous page and 11 on the preceding page.

### **Output**

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully, the description section of the output will be: “The file chunk was added into the session successfully”. Please note that if any of the calls to this methods fails the session will be close and the information uploaded will be lost. Please create a new session and start again. If these issues persist consider lowering the chunk size.

#### **2.4.15 Close a File Upload Session and save the file into the VPS**

**Method Name** closeFileUploadSession

**Description** This method will take all the information uploaded into a File Upload Session and deposit it as a file by the provided name in a folder inside the selected VPS MetaTrader folder. The selection of the sub folder is set by the provided File Type. The supported types are:

File Type Value	Destination subfolder inside the MetaTrader installation
EA	experts/
Indicator	experts/indicators/
Include	experts/include/
Preset	experts/presets/
Script	experts/scripts/
Template	templates/
Config	config/
DLL	experts/libraries/
Files	experts/files/
Profiles	profiles/
MetaTrader	The meta trader folder itself

Table 2: Files Types and destinations

### Input

**username:** String - The end users account name (email address).

**password:** String - The end users password.

**retailerCode:** String - The retailer code assigned to the users retailer.

**vpsId:** String - The ID for the VPS you wish to modify.

**sessionToken:** String - A File Upload session token as provided by the CreateFileUploadSession method ( 2.4.13 on page 33).

**fileType:** String - The type of file being uploaded, allowed valued are described in table 2.

**fileName:** String - The name of file being uploaded.

### Output

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully, the description section of the output will be: “The file was uploaded successfully”

#### 2.4.16 Set Account Owner Name

**Method Name** setOwnerName

**Description** This method allows reset the owner name of the account.

### **Input**

**username:** String - The end users account name (email address).

**password:** String - The end users password.

**retailerCode:** String - The retailer code assigned to the users retailer.

**firstName:** String - The accounts owner first name.

**lastName:** String - The accounts owner last name.

### **Output**

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully the description should be:  
“The accounts owner name has been set.”

#### **2.4.17 Set VPS Remote Access Password**

**Method Name** setRemoteAccessPassword

**Description** This method will change the remote access password to the VPS. Not to be confused with the account password. The new password is expected to be base64 encoded for safe transport.

### **Input**

**username:** String - The end users account name (email address).

**password:** String - The end users password.

**retailerCode:** String - The retailer code assigned to the users retailer.

**vpsId:** String - The ID for the VPS you wish to modify.

**newPass:** String - The new remote access password encoded in base64.

### **Output**

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully, the description section of the output will be:  
“The VPS password was set successfully.”

#### **2.4.18 Set VPS Nickname**

**Method Name** setNickname

**Description** This method will change the nickname assigned to the VPS.

### **Input**

**username:** String - The end users account name (email address).

**password:** String - The end users password.

**retailerCode:** String - The retailer code assigned to the users retailer.

**vpsId:** String - The ID for the VPS you wish to modify.

**newName:** String - The new nickname for the VPS.

### **Output**

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully, the description section of the output will be:

“Nickname was set successfully.”

#### **2.4.19 Get Labels**

**Method Name** getLabels

**Description** This method will return an Array of Arrays of Strings containing the labels as set via Manage Labels UI by the manager of the retailer..

### **Input**

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**retailerCode:** String - The retailer code assigned to the users retailer.

### **Output**

On successful completion this method will return an Array of Arrays of Strings in the following structure:

```
Array {  
  Array {  
    Array {“[Label Name]”, “[Label Value]”},  
    Array {“[Another Label Name]”, “[Another Label Value]”},  
    ...  
  }  
}
```

#### **2.4.20 Get Available Languages**

**Method Name** getAvailableLanguages

**Description** This method will return an Array of Arrays of Strings containing all the languages supported by the GallantVPS system in Language Code, Language Name pairs.

### Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**retailerCode:** String - The retailer code assigned to the users retailer.

### Output

On successful completion this method will return an Array of Arrays of Strings in the following structure:

```
Array {  
  Array {  
    Array {"[Language Code e.g en]","[Language Name e.g English]"},  
    Array {"[Another Language Code ]","[Another Language Name]"},  
    ...  
  }  
}
```

#### 2.4.21 Set Account UI language

**Method Name** setLanguage

**Description** This method allows the retailer to set the accounts UI Language to one of the available values provided by the getAvailableLanguages method ( 2.4.20 on the preceding page). Please note that the VPS UI language relies on available labels value as provided by the getLables method ( 2.4.19 on the previous page). If the label value in the set language is missing the label will default to English.

### Input

**username:** String - API authentication token as described in section 1.3 on page 5.

**password:** String - API authentication token as described in section 1.3 on page 5.

**retailerCode:** String - The retailer code assigned to the users retailer.

**languageCode:** String - The preferred languageCode as provided by getAvailableLanguages method ( 2.4.20 on the preceding page) .

### Output

This method will return its output in the standard structure as described in section 1.4.1 on page 6.

If completed successfully be the description should be:

“The account language was set to [Language Name].”