# Table Of Contents

## Introduction

The Ubersmith API is a RESTful web service for communicating with an Ubersmith DE software installation.

This document provides documentation on the commands and parameters available in the Ubersmith API.

## API Sections

The API is broken into several sections, each containing commands having related functions. The sections are:

- `uber`    - Functions of a general nature
- `client`  - Functions related to the management of clients, contacts, services, and invoices
- `device`  - Functions related to the management of devices, monitors, and IP assignments
- `order`   - Functions related to orders
- `sales`   - Functions related to sales opportunities
- `support`- Functions related to support departments and tickets

Methods within each section are identified using a dot notation, e.g. `uber.methods_list`.

## API Authentication

The Ubersmith API supports the use of API Tokens as well as passwords for authentication. The use of API Tokens allows generating separate tokens for each external system accessing the API, and optionally setting an expiration date for the token.

API Tokens can be managed from the User Management page in Setup & Admin, by clicking the 'api tokens' link which appears next to any user who has been granted API access. Once generated a token may be used in place of the user's password for API access.

Authentication details are passed to the Ubersmith API via HTTP authentication, if you are using our client class this is taken care of automatically.

## Using the Ubersmith API

For supported languages we recommend using our client (see below). For integrating other languages the Ubersmith API uses standard HTTP requests and returns data in JSON format.

The Ubersmith API is accessed through the URL `https://billing.mycompany.com/api/2.0/`, where 'billing.mycompany.com' is the address of your Ubersmith installation.

API methods are called by passing the parameter `'method'` in the URL, like `'https://billing.mycompany.com/api/2.0/?method=uber.method_list'`, which will return a list of all the available API methods.

Additional parameters can be passed to the API either as an array or as a JSON encoded string.

To use an array, either pass the array as GET parameteres or as a standard HTTP POST request.

To use a JSON encoded string, set the header to `'Content-type: application/json'` and pass the JSON encoded string as a standard HTTP POST request:

```
{
  "brand_id":"1",
  "active":"1",
  "order_by":"name",
  "limit":"10"
}
```

By default, the language used in the API response will be based off the user account's preferred language. To manually specify a language for the output, you can pass a `'lang_id'` parameter in the URL with either the numeric language id or the language locale, e.g. en_US.

Responses are returned in JSON format (with the exception of specialized methods which return raw PDF, Image, XML or HTML data), and include the standard elements:

- `status`    - API call success true/false
- `error_code`    - error code for failed API calls
- `error_message` - error message for failed API calls
- `data`    - results of API call

## Example Output

```
{
  "status":true,
  "error_code":null,
  "error_message":"",
  "data": "<api call output here>"
}
```

# API Clients

## PHP

A PHP API client class and example are included in the `'api/client/2.0/php'` directory of the Ubersmith distribution:

The file `'class.uber_api_client.php'` provides the PHP class `'uber_api_client'`, which can be used to interact with the Ubersmith API.

By default, parameters will be passed to the API as an array. If you set the `'json_req'` option to `'true'` however, using the `'uber_api_client->set_option'` method, the parameters will instead be passed to the API as a JSON encoded string.

The file `'sample.php'` provides an example of how to use the class to retrieve data from the API.

## Other Languages

As other language implementations become available, the will be included in the `'api/client/2.0/'` directory. If you have any questions about creating your own client class, or would like to contribute a client class you have written, please contact Ubersmith Support.

# uber - Ubersmith

Functions of a general nature

## uber.acl_resource_add - Add ACL resource

This method is used to add an ACL resource

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| parent_resource_name | Parent Resource Name | Y | | |
| resource_name | Resource Name | Y | | Can not start with "admin." or "client." |
| label | Resource Label | Y | | |
| actions | Available Actions | | | Comma-separated list of actions for this resource. This overrides default CRUD actions |

**Example Output**

```
{
  "status":true,
  "error_code":null,
  "error_message":"",
  "data":[]
}
```

## uber.acl_resource_delete - Delete ACL resource

This method is used to delete ACL resources

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| resource_name | Resource Name | Y | | |

**Example Output**

```
{
  "status":true,
  "error_code":null,
  "error_message":"",
  "data":[]
}
```

## uber.acl_resource_list - List ACL resources

This method is used to list ACL resources

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| resource_name | Resource Name | | | |

**Example Output**

```
{
  "status":true,
  "error_code":null,
  "error_message":"",
  "data":{
    "108": {
      "resource_id": "108",
      "parent_id": "1",
      "label": "Admin View",
      "name": "admin.portal",
      "lft": "214",
      "rgt": "293",
      "actions": {
        "1": "Create",
        "2": "View",
        "3": "Update",
        "4": "Delete"
      },
      "children": {
```

```
            "109": {
                "resource_id": "109",
                "parent_id": "108",
                "label": "Client Manager",
                "name": "clientmgr",
                "lft": "215",
                "rgt": "226",
                "actions": {
                    "1": "Create",
                    "2": "View",
                    "3": "Update",
                    "4": "Delete"
                }
            }
        }
      }
    }
  }
}
```

## uber.acl_resource_update - Update ACL resource

This method is used to update ACL resources

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| resource_name | Resource Name | Y | | |
| label | Resource Label | | | |
| actions | Available Actions | | | Comma-separated list of actions for this resource. This overrides default CRUD actions |

**Example Output**

```
{
  "status":true,
  "error_code":null,
  "error_message":"",
  "data":[]
}
```

## uber.admin_avatar_get - Retrieve an Admin Avatar

This method is used to retrieve an admin user avatar.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| user_id | User ID | | | |
| brand | Brand | | | |
| size | Size | | | Specify 'small' to load a 25 pixel avatar; otherwise a 50 pixel avatar will be returned. |

**Example Output**

```
{
"data": {
"image_base64": [base64-encoded image data],
"image_type":  "image/png"
},
"error_message": "",
"error_code":    null,
"status":        true
}
```

## uber.admin_avatar_set - Set an Admin Avatar

This method is used to set an admin user avatar.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| user_id | Client ID | | | |
| brand | Brand ID | | | |
| avatar | Attachment | | | An image file for the avatar should be specified. When using cURL the value for the avatar parameter should be @ where is the avatar you wish to attach. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.admin_client_relationship_add - Add Admin-Client Relationship

This method adds an admin-client relationship such as a dedicated Account Manager or Salesperson.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| client_id | Client ID | Y | | |
| person_id | Admin ID | Y | | |
| people_client_type _id | Relationship Type ID | Y | | |

**Example Output**

```
{
  "data": "413",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.admin_client_relationship_delete - Remove Admin-Client Relationship

This method removes an existing admin-client relationship.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| people_client_id | Relationship ID | Y | | |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.admin_client_relationship_list - List Admin-Client Relationships

This method lists admin-client relationship filtered by the parameters specified.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| people_client_id | Relationship ID | | | |
| client_id | Client ID | | | |
| person_id | Admin ID | | | |
| people_client_type _id | Relationship Type ID | | | |
| pct_status | Relationship Type Status | | 1 | Options: 1 = Active, 0 = Inactive |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "1": {
      "people_client_id": "1",
      "person_id": "2",
      "client_id": "1009",
      "people_client_type_id": "2",
      "company": "ABC Corp",
      "person_name": "Steph Staff",
      "pct_status": "1",
      "people_client_type_name": "Account Manager"
    },
    "2": {
      "people_client_id": "2",
      "person_id": "2",
      "client_id": "1012",
      "people_client_type_id": "2",
      "company": "MLE dba Emily",
      "person_name": "Steph Staff",
      "pct_status": "1",
      "people_client_type_name": "Account Manager"
    }
  }
}
```

## uber.admin_client_relationship_type_add - Add Admin-Client Relationship Type

This method adds a admin-client relationship type to the system.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| name | Relationship Type Name | | | Label for relationship type |
| status | Status | | 1 | Options: 1 = Active, 0 = Inactive |
| send_notifications | Send Notifications | | 0 | Send a copy of notifications to admin. Options: 1 = Yes, 0 = No |

**Example Output**

```
{
  "data": "6726",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.admin_client_relationship_type_list - List Admin-Client Relationship Types

This method lists existing admin-client relationship types.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| people_client_type_id | Relationship Type ID | | | Unique admin-client relationship type identifier |
| name | Relationship Type Name | | | Label for relationship type |
| status | Status | | 1 | Options: 1 = Active, 0 = Inactive |
| send_notifications | Send Notifications | | | Send a copy of notifications to admin. Options: 1 = Yes, 0 = No |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "1": {
      "people_client_type_id": "1",
      "name": "Salesperson",
      "status": "1",
      "send_notifications": "0"
    },
    "2": {
      "people_client_type_id": "2",
      "name": "Account Manager",
      "status": "1",
      "send_notifications": "1"
    }
  }
}
```

## uber.admin_client_relationship_type_update - Edit Admin-Client Relationship Type

This method edits an existing admin-client relationship type.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| people_client_type_id | Relationship Type ID | Y | | Unique admin-client relationship type identifier |
| name | Relationship Type Name | | | Label for relationship type |
| status | Status | | | Options: 1 = Active, 0 = Inactive |
| send_notifications | Send Notifications | | | Send a copy of notifications to admin. Options: 1 = Yes, 0 = No |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.admin_client_relationship_update - Edit Admin-Client Relationship

This method edits an existing admin-client relationship.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| people_client_id | Relationship ID | Y | | |
| client_id | Client ID | | | |
| person_id | Admin ID | | | |
| people_client_type_id | Relationship Type ID | | | |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.admin_get - User Information

This method is used to return a admin user information based on the user ID or username.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| userid | User ID | | | |
| username | User Name | | | |

**Example Output**

```
{
  "data": {
      "id":"1"
        "name": "Johnny Ubersmith",
        "email": "someone@ubersmith.com",
        "signature": "",
        "username": "juber",
        "password": "{ssha1}1234567890abcdef",
        "password_timeout": "0",
        "password_changed": "0",
          "access": {"devicemgr":"2","supportmgr":"2","ordermgr":"2"}",
"config":"",
        "last_login": "123456789",
        "current_login": "123456789",
        "prefer_lang": "1",
        "brands": {"brand1":"2","brand2":"2","brand3":"2"}",
        "api": "1",
        "active": "1"
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.admin_list - List User Logins

This method is used to returns a list of login names defined.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| active | User Active | | 1 | Display active or deactivated users |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
      "1": {
      "id":"1"
        "name": "Johnny Ubersmith",
        "email": "someone@ubersmith.com",
        "signature": "",
        "username": "juber",
        "password": "{ssha1}1234567890abcdef",
        "password_timeout": "0",
        "password_changed": "0",
          "access": {"devicemgr":"2","supportmgr":"2","ordermgr":"2"}",
"config":"",
        "last_login": "123456789",
        "current_login": "123456789",
        "prefer_lang": "1",
        "active": "1"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.admin_permission_get - Get User Permission

This method is used to get admin user permission.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| userid | User ID | | | |
| username | User Name | | | |
| resource_name | Resource Name | Y | | |
| action | Action | Y | | |

**Example Output**

```
{
  "data": ""
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.admin_permission_list - List User Permissions

This method is used to return a tree/list of user permissions.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| userid | User ID | | | |
| username | User Name | | | |
| resource | Resource Name | | | |
| format | Listing format | | | Option: tree (default), list |
| effective | Include Effective Actions | | 0 | Include the effective (inherited) actions along with excplicit actions |

**Example Output**

```
{
  "data": {
      "1": {
          "resource_id": "1",
          "name": "admin.portal",
          "label": "Admin Portal",
          "parent_id": "0",
          "resource_action_id": "1",
          "action": {
              "create": "1",
              "read": "1",
              "update": "1",
              "delete": "1"
          },
          "effective": {
              "create": "1",
              "read": "1",
              "update": "1",
              "delete": "1"
          },
          "actions": "",
          "children": {
              "2": {
                  "resource_id": "2",
                  "name": "clientmgr",
                  "label": "Client Manager",
                  "parent_id": "1",
                  "resource_action_id": "1",
                  "action": {
                      "create": "1",
                      "read": "1",
                      "update": "1",
                      "delete": "1"
                  },
                  "actions": "",
              }
          }
      }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

::: ubersmith

## uber.admin_permission_set - Set User Permissions

This method is used to set admin user permissions.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| userid | User ID | | | |
| username | User Name | | | |
| resource_name | Resource Name | Y | | |
| action | Action | Y | | |
| type | Permission Type | Y | | Available values: allow, deny, inherit |

**Example Output**

```
{
  "data": ""
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.admin_resource_list - List available admin ACL resources

This method is used to list ACL resources that may be set for an admin

**Example Output**

```
{
  "status":true,
  "error_code":null,
  "error_message":"",
  "data":{
    "108": {
        "resource_id": "108",
        "parent_id": "1",
        "label": "Admin View",
        "name": "admin.portal",
        "lft": "214",
        "rgt": "293",
        "actions": {
            "1": "Create",
            "2": "View",
            "3": "Update",
            "4": "Delete"
        },
        "children": {
            "109": {
                "resource_id": "109",
                "parent_id": "108",
                "label": "Client Manager",
                "name": "clientmgr",
                "lft": "215",
                "rgt": "226",
                "actions": {
                    "1": "Create",
                    "2": "View",
                    "3": "Update",
                    "4": "Delete"
                }
            }
        }
    }
  }
}
```

## uber.api_export - Export Data

This method is used to export data from Ubersmith in CSV format.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| table | Table to export | Y | | Available tables: client, packages, services, invoices, events, tickets, plans, stored_mail, contacts, credits, mail, payment_records |
| gzip | Gzip Flag | | | If set to 1, output will be gzipped |
| order_by | Order By | | | Field to order output by |

**Example Output**

```
Data from the specified table in CSV format.
```

## uber.appliance_list - List Ubersmith Appliances

This method lists Ubersmith Appliances.

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "1": {
      "appliance_id": "1",
      "name": "Appliance",
      "active": "1",
      "address": "http:\/\/appliance1.ubersmith.com\/"
    },
    "2": {
      "appliance_id": "2",
      "name": "Another Appliance",
      "active": "1",
      "address": "http:\/\/appliance2.ubersmith.com\/"
    }
  }
}
```

## uber.attachment_get - Get an attachment

This method is used to get a specified attachment.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| attach_type | Attachment Type | Y | | Possible types: ticket, comment, doc, email, order_ticket |
| attach_id | Attachment ID | Y | | |

**Example Output**

```
This method will output file data for the specified attachment.
```

## uber.attachment_list - List Attachments

This method is used to output a list of attachments

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| attach_type | Attachment type | Y | | Possible types: ticket, comment, doc, email, order_ticket, quote |
| attach_id | Attachment ID | | | |
| id | ID of ticket, comment, etc. | | | |

**Example Output**

```
{
  "data": {
    "1946": {
      "attach_id": "1946",
      "conttype": "application/pdf",
      "filename": "Invoice-1251-411.pdf"
      "size": "10276",
      "id": "20",
      "attach_type": "ticket",
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.billing_period_list - List Billing Periods

This method is used to get a list of billing periods.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| brand_id | Brand ID | Y | | |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "0": {
      "brand": "-1",
      "period": "0",
      "type": "m",
      "s_name": "One Time",
      "p_name": "One Time Fee",
      "api_label": "one_time",
      "s_name_en_US": "One Time",
      "p_name_en_US": "One Time Fee",
      "s_name_fr_CA": "One Time",
      "p_name_fr_CA": "Frais uniques"
    }
  }
}
```

## uber.brand_list - List Brands

This method is used to returns a list of brands.

**Example Output**

```
{
"data": {
"1": {
"class_id"  :"1"
"class_name": "default",
"class_abbr": "dflt",
"serverurl" : "http://www.ubersmith.com",
"class_name_en_US": "default",
"class_name_fr_CA": "default"
}
},
"error_message": "",
"error_code": null,
"status": true
}
```

## uber.check_login - Verify a login and password

This method is used to check the specified login and password.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| login | Login Name | Y | | |
| pass | Password | Y | | |

**Example Output**

```
{
  "data": {
    "last_login": "1312293922",
    "password_expired": "0",
    "fullname": "Administrator",
    "contact_id": null,
    "password_changed": "1269009589",
    "id": "1",
    "type": "admin",
    "class_id": 1,
    "client_id": null,
    "password_timeout": "0",
    "access": {
      "client_reports": "full",
      "global_settings": "full",
      "supportmgr": "full",
      "client_contact_info": "full",
      "sm_settings": "full",
      "ordermgr": "full",
      "salesmgr_settings": "full",
      "om_settings": "full",
      "cm_settings": "full",
      "api": true,
      "reports_access": "full",
      "devicemgr": "full",
      "queues": {
        "1": 1,
        "2": 1,
        "3": 1,
        "4": 1,
        "5": 1
      },
      "dm_settings": "full",
      "client_comments": "full",
      "ci_settings": "full",
      "client_billing": "full",
      "client_services": "full",
      "client_acctmgmt": "full",
      "all_brands": "full"
    },
    "login": "admin",
    "email": "uber@ubersmith.com"
  },
  "error_message": "",
```

```
    "error_code": null,
    "status": true
}
```

## uber.client_permission_list - List available permissions

This method is used to list permissions that may be set for a client or contact

**Example Output**

```
{
  "status":true,
  "error_code":null,
  "error_message":"",
  "data":{
    "client_profile":{
        "title":"Client Profile",
        "items":{
          "view_profile":{
              "title":"Profile",
              "levels":{
                "":"None",
                "view":"View",
                "edit":"Edit"
              }
          }
        }
    }
  }
}
```

## uber.client_resource_list - List available ACL resources

This method is used to list ACL resources that may be set for a client or contact

**Example Output**

```
{
  "status":true,
  "error_code":null,
  "error_message":"",
  "data":{
    "108": {
        "resource_id": "108",
        "parent_id": "1",
        "label": "Client View",
        "name": "client.portal",
        "lft": "214",
        "rgt": "293",
        "actions": {
            "1": "Create",
            "2": "View",
            "3": "Update",
            "4": "Delete"
        },
        "children": {
            "109": {
                "resource_id": "109",
                "parent_id": "108",
                "label": "Client Profile",
                "name": "client.client_profile",
                "lft": "215",
                "rgt": "226",
                "actions": {
                    "1": "Create",
                    "2": "View",
                    "3": "Update",
                    "4": "Delete"
                }
            }
        }
    }
  }
}
```

## uber.client_role_get - Get role details

Obtains information about a given Client Role

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_role_id | Client Role ID | Y | | |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "client_role_id": "1",
    "name": "New Client Role",
    "descr": "Sample description",
    "assignable": "1",
    "created_ts": "1400251234",
    "modified_ts": "1400252473",
    "user_count": "8",
    "acls": {
      "client.profile": {
        "create": "1",
        "read": "1",
        "update": "1",
        "delete": "-1"
      }
    }
  }
}
```

## uber.client_role_list - List available client roles

This method is used to list client roles that may be set for a client or contact

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | | | (Optional) Filter roles to those assignable by this client |
| contact_id | Contact ID | | | (Optional) Filter roles to those assignable to this contact |
| acls | Client ACLs | | | 1 = Include the ACLs that each role provides |

**Example Output**

```
{
  "status":true,
  "error_code":null,
  "error_message":"",
  "data":{
    1: {
      client_role_id: "1"
      name: "New Client Role"
      descr: "Sample description"
      assignable: "1"
      created_ts: "0"
      modified_ts: "1400252473"
      "acls": {
        "client.profile": {
          "create": "1",
          "read": "1",
          "update": "1",
          "delete": "-1"
        },
        "client.billing_info": {
          "read": "1"
        }
      }
    }
  }
}
```

## uber.client_tag_add - Add a client tag

This method is used to create a client tag

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| tag | Tag Name | Y | | |
| color | Color | | | Color in RGB hex |
| important | Important? | | 0 | If tag should be made prominent in UI |

**Example Output**

```
{
  "data": 8,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.client_tag_delete - Delete a client tag

This method is used to delete a client tag -- warning: this removes the tag permanently from all clients

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| tag_id | Tag ID | Y | | |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.client_tag_list - List client tags

This method is used to list client tags

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | | | |
| has_clients | Has Clients | | | |
| important | Important? | | | If a tag is listed as important |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
"status": true,
"error_code": null,
"error_message": "",
"data":  {
"3": {
"tag_id": "3",
"tag": "Abuse",
"color": "ff000f",
"important": "1",
"clients": "3"},
"2": {
"tag_id": "2",
"tag": "VIP",
"color": "0000ff",
"important": "1",
"clients": "3"}
```

```
}}
```

## uber.client_tag_update - Update a client tag

This method is used to update an existing client tag

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| tag_id | Tag ID | Y | | |
| tag | Tag Name | | | |
| color | Color | | | Color in RGB hex |
| important | Important? | | 0 | If tag should be made prominent in UI |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.client_welcome_stats - Display Client Statistics

This method is used to output the same statistics that are shown at the top of the client interface.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| user_login | Client Login | Y | | The client account |

**Example Output**

```
{
  "data": {
    "ticket": "",
    "inv_count": "0",
    "activity": 0,
    "subject": "",
    "next_inv": "08/31/2011",
    "type": "",
    "ticket_count": 0,
    "updated_by": "",
    "updated": "",
    "pack_count": "0",
    "closed_count": 0
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.comment_add - Add Comment

This method is used to add client, service, device, credit or opportunity comments.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| type | Comment Type | Y | | client, service, device, credit, opportunity or quote |
| ref_id | Related ID | Y | | ID number of client, service, device, credit, opportunity or quote |
| comment | Comment | Y | | |
| user_id | User ID | | api user | |
| attach[x] | Attachment | | | Up to 5 attachments may be specified as attach[0], attach[1], etc. When using cURL the value for each attach[x] parameter should be @ where is the file you wish to attach. |

**Example Output**

```
{
  "data": "41",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.comment_delete - Delete Comment

This method is used to delete client, service, device, credit or opportunity comments.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| type | Comment Type | Y | | client, service, device, credit, opportunity or quote |
| comment_id | Comment ID | Y | | |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.comment_get - Get Comments

This method is used to get client, service, device, credit or opportunity comments.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| type | Comment Type | Y | | client, service, device, credit, opportunity or quote |
| comment_id | Comment ID | Y | | |

**Example Output**

```
{
  "data": {
    "edited": "1312294426",
    "comment": "api comment",
    "editor": "",
    "time": "1312294426",
    "type": "client",
    "clientid": "1080",
    "user": "Administrator",
    "comment_id": "42",
    "client_viewable": "0"
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.comment_list - List Comments

This method is used to list client, service, device, credit or opportunity comments.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| type | Comment Type | Y | | client, service, device, credit, opportunity or quote |
| ref_id | Related ID | Y | | ID number of client, service, device, credit, opportunity or quote |
| filter_text | Filter Text | | | An array of strings to match against comments and attachment filenames. Comments which match all terms are returned. |
| client_viewable | Client Viewable | | | If client_viewable needs to be filtered (1 to only show client_viewable) |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "42": {
      "edited": "1312294426",
      "comment": "api comment",
      "editor": "",
      "time": "1312294426",
      "clientid": "1080",
      "user": "Administrator",
      "comment_id": "42",
      "client_viewable": "0"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.comment_update - Update Comment

This method is used to update client, service, device, credit or opportunity comments.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| type | Comment Type | Y | | client, service, device, credit, opportunity or quote |
| comment_id | Comment ID | Y | | |
| comment | Comment | Y | | |
| user_id | User ID | | api user | |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.config_list - List Configuration Values

This method is used to list configuration values.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| brand_id | Brand ID | Y | | |
| items | Items | | | If specified as either a string or an array, only the configuration items specified will be returned.<br>If a string is specified, % can be used as a wildcard.<br><br>Examples:<br>'look_feel.serverurl' would return the item 'look_feel.serverurl'.<br>'look_feel%' would return all items starting with 'look_feel'.<br>array('look_feel.serverurl', 'look_feel.replymail') would return the items 'look_feel.serverurl' and 'look_feel.replymail'. |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "clientmgr.default_late_fee_scheme_id": "0",
    "clientmgr.inv_date_range_style": "0",
    "clientmgr.inv_title": "Invoice",
    "clientmgr.inv_title_en_US": "Invoice",
    "client_selectable_invoice_delivery": "1"
  }
}
```

## uber.config_update - Update Configuration Value

This method is used to update a configuration value.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| brand_id | Brand ID | Y | | |
| items | Items | Y | | An associative array of the configuration items to be updated with their new values. For locale specific items, append the locale (clientmgr.inv_title_en_US etc.) to the end of the item name. |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": true
}
```

## uber.contract_term_add - Add a Contract Term

This method is used to add a contract term

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| name | Contract Term Name | Y | | |
| term | Term Length | Y | | Length of the contract in months |
| status | Status | | 1 | |

**Example Output**

```
{
"data": "10",
"error_message": "",
"error_code": null,
"status": true
}
```

## uber.contract_term_get - Get Contract Term Details

This method is used to get the details of a specified contract term

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| contract_term_id | Contract Term ID | Y | | |

**Example Output**

```
{
"data": {
"contract_term_id": "1",
"name": "Contract Name",
"status": "1",
"term": "12"
},
"error_message": "",
"error_code": null,
"status": true
}
```

## uber.contract_term_list - List Contract Terms

This method is used to list contract terms

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| status | Status | | | 1 = Active Contract Terms, 0 = Inactive Contract Terms |

**Example Output**

```
{
"data": {
"1": {
"contract_term_id": "1",
"name": "Contract Name",
"status": "1",
"term": "12"
},
"2": {
"contract_term_id": "2",
"name": "Contract Name",
"status": "1",
"term": "12"
}
},
"error_message": "",
"error_code": null,
"status": true
}
```

## uber.contract_term_update - Update a Contract Term

This method is used to update a contract term

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| contract_term_id | Contract Term ID | Y | | |
| name | Contract Term Name | | | |
| term | Term Length | | | Length of the contract in months |
| status | Status | | | 1 = Active, 0 = Inactive |

**Example Output**

```
{
"data": {
"contract_term_id": "1",
"name": "Contract Name",
"status": "1",
"term": "12"
},
"error_message": "",
"error_code": null,
"status": true
}
```

## uber.documentation - Download API Documentation

This method is used to get a document with details of all available API methods.

**Example Output**

```
This method will output PDF data containing the API documentation.
```

## uber.event_list - Access the Event Log

This method is used to retrieve data from the event log.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| user_login | Client Login | | | Client's login name or ID |
| begin | Begin Time | | | A Unix timestamp. If empty, output will begin from the earliest event. |
| end | End Time | | | A Unix timestamp. If empty, output will end at the most recent event. |
| searchfor | Search term | | | Used to return only those events that include a specific keyword. |
| ledger | Ledger events only | | 0 | Return all events or just those that appear in the ledger event log. |
| order_by | Order By | | time | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | 100 | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "6379": {
      "ledger": "0",
      "eventid": "6379",
      "amount": null,
      "time": "1278014859",
      "action": "Invoice {{1083-247}} marked paid.",
      "reference_id": "247",
```

```
            "type": "0",
            "reference_type": "invoice",
            "clientid": "1083",
            "event_type": "95",
            "user": "system",
            "client_viewable": "1",
            "balance": "0.00"
         }
    },
    "error_message": "",
    "error_code": null,
    "status": true
}
```

## uber.event_trigger_action_get - Get Event Trigger Action Details

This method is used to get an event trigger action's details.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| ea_id | Event Trigger Action ID | | | Either this or ea_name must be provided. |
| ea_name | Event Trigger Action Name | | | Either this or ea_id must be provided. |
| brand_id | Brand ID | | | |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "ea_id": 1,
    "ea_title": "Client Added",
    "ea_group_id": 1,
    "ea_var_list": {
      "clientid": "Client Id #",
      "brand_id": "Brand Id #",
      "first": "First Name",
      "last": "Last Name",
      "company": "Company",
      "address": "Address",
      "city": "City",
      "state": "State",
      "zip": "Zip Code",
      "country": "Country/Territory",
      "phone": "Phone",
      "fax": "Fax",
      "email": "Email",
      "referred": "Referred By",
      "datesend": "Invoice Send Date",
      "daysdue": "Grace Period",
      "datedue": "Static Due Date",
      "retry_every": "Charge Retry Interval",
      "uber_login": "Ubersmith Login Name",
      "uber_pass": "Ubersmith Password",
      "priority": "Default Priority Level",
      "discount": "Default Discount",
      "user": "User"
    },
    "scripts": {
      "1": {
        "es_id": "1",
        "es_action_id": "1",
        "es_class_id": "1",
        "es_label": "Example",
        "es_url": "http://www.example.com/",
        "es_priority": "1",
        "es_rs_check": "1",
        "es_no_response": "1",
        "args": {
          "client_id": "clientid",
          "service_id": "packid"
        },
        "matches": [
```

```
        {
          "er_match": "Y",
          "er_value": "success",
          "er_action": "1"
        },
        {
          "er_match": "N",
          "er_value": "failure",
          "er_action": "0"
        }
      ]
    }
  }
 }
}
```

## uber.event_trigger_action_list - List Event Trigger Actions

This method is used to list event trigger actions.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| brand_id | Brand ID | | | |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "client_added": {
      "ea_id": 1,
      "ea_title": "Client Added",
      "ea_group_id": 1,
      "ea_var_list": {
        "clientid": "Client Id #",
        "brand_id": "Brand Id #",
        "first": "First Name",
        "last": "Last Name",
        "company": "Company",
        "address": "Address",
        "city": "City",
        "state": "State",
        "zip": "Zip Code",
        "country": "Country/Territory",
        "phone": "Phone",
        "fax": "Fax",
        "email": "Email",
        "referred": "Referred By",
        "datesend": "Invoice Send Date",
        "daysdue": "Grace Period",
        "datedue": "Static Due Date",
        "retry_every": "Charge Retry Interval",
        "uber_login": "Ubersmith Login Name",
        "uber_pass": "Ubersmith Password",
        "priority": "Default Priority Level",
        "discount": "Default Discount",
        "user": "User"
      },
      "scripts": {
        "1": {
          "es_id": "1",
          "es_action_id": "1",
          "es_class_id": "1",
          "es_label": "Example",
          "es_url": "http://www.example.com/",
          "es_priority": "1",
          "es_rs_check": "1",
          "es_no_response": "1",
          "args": {
            "client_id": "clientid",
            "service_id": "packid"
          },
          "matches": [
            {
              "er_match": "Y",
              "er_value": "success",
```

```
                "er_action": "1"
            },
            {
                "er_match": "N",
                "er_value": "failure",
                "er_action": "0"
            }
        ]
    }
  }
 }
 }
}
```

## uber.event_trigger_script_add - Add a New Event Trigger Script

This method is used to add a new event trigger script.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| ea_id | Event Trigger Action ID | | | Either this or ea_name must be provided. |
| ea_name | Event Trigger Action Name | | | Either this or ea_id must be provided. |
| es_label | Label | Y | | |
| es_url | URL | Y | | |
| args | Variables | | | An associative array of script variables.<br><br>Example: args[client]=clientid&args[service]=packid |
| es_rs_check | Response Action | | 0 | How to handle script responses:<br>0 = Do nothing<br>1 = Expect and interpret output from the called script |
| es_no_response | Script Timeout Action | | | How to handle a script timeout:<br>0 = Continue to next script<br>1 = Stop Event Actions |
| matches | Response Matches | | | A multidimensional array specifying the action(s) to take when a response matches or doesn't match a certain value:<br><br>Example: matches[0][er_match]=Y&matches[0][er_value]=success&matches[0][er_action]=1&matches[1][er_match]=N&matches[1][er_value]=success&matches[1][er_action]=0 would continue to the next script for a response matching 'success' and stop event actions for a response not matching 'success'. |
| brand_id | Brand ID | | | |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": true
}
```

## uber.event_trigger_script_delete - Delete an Event Trigger Script

This method is used to delete an event trigger script.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| es_id | Event Trigger Script ID | Y | | |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": true
}
```

## uber.event_trigger_script_get - Get Event Trigger Script Details

This method is used to get an event trigger script's details.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| es_id | Event Trigger Script ID | Y | | |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "es_id": "1",
    "es_action_id": "1",
    "es_class_id": "1",
    "es_label": "Example",
    "es_url": "http://www.example.com/",
    "es_priority": "1",
    "es_rs_check": "1",
    "es_no_response": "1",
    "args": {
      "client_id": "clientid",
      "service_id": "packid"
    },
    "matches": [
      {
        "er_match": "Y",
        "er_value": "success",
        "er_action": "1"
      },
      {
        "er_match": "N",
        "er_value": "failure",
        "er_action": "0"
      }
    ]
  }
}
```

## uber.event_trigger_script_list - List Event Trigger Scripts

This method is used to list event trigger scripts.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| ea_id | Event Trigger Action ID | | | Either this or ea_name must be provided. |
| ea_name | Event Trigger Action Name | | | Either this or ea_id must be provided. |
| brand_id | Brand ID | | | |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "1": {
      "es_id": "1",
      "es_action_id": "1",
      "es_class_id": "1",
      "es_label": "Example",
      "es_url": "http://www.example.com/",
      "es_priority": "1",
      "es_rs_check": "1",
      "es_no_response": "1",
      "args": {
        "client_id": "clientid",
        "service_id": "packid"
      },
      "matches": [
        {
          "er_match": "Y",
          "er_value": "success",
          "er_action": "1"
        },
        {
          "er_match": "N",
          "er_value": "failure",
          "er_action": "0"
        }
      ]
    }
  }
}
```

## uber.event_trigger_script_update - Update an Event Trigger Script

This method is used to update an event trigger script.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| es_id | Event Trigger Script ID | Y | | |
| es_label | Label | | | |
| es_url | URL | | | |
| args | Variables | | | An associative array of script variables.<br><br>Example: args[client]=clientid&args[service]=packid |
| es_rs_check | Response Action | | 0 | How to handle script responses:<br>0 = Do nothing<br>1 = Expect and interpret output from the called script |
| es_no_response | Script Timeout Action | | | How to handle a script timeout:<br>0 = Continue to next script<br>1 = Stop Event Actions |
| matches | Response Matches | | | A multidimensional array specifying the action(s) to take when a response matches or doesn't match a certain value:<br><br>Example: matches[0][er_match]=Y&matches[0][er_value]=success&matches[0][er_action]=1&matches[1][er_match]=N&matches[1][er_value]=success&matches[1][er_action]=0 would continue to the next script for a response matching 'success' and stop event actions for a response not matching 'success'. |
| brand_id | Brand ID | | | |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": true
}
```

## uber.file_add - Add a file

This method is used to add a file.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | | | |
| name | | | | |
| type | | | | |
| contents | base64 encoded data | | | |
| public | | | | |

**Example Output**

{"status":true,"error_code":null,"error_message":"","data":{"id":"24","hash":"106fee5536e5"}}

## uber.file_delete - Delete a file

This method is used to delete a file.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| id | File ID | | | |
| hash | File Hash | | | |

**Example Output**

{"status":true,"error_code":null,"error_message":"","data":"true"}

## uber.file_get - Get a File

This method is used to return a single file.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| id | File ID | | | |
| hash | File Hash | | | |

**Example Output**

{"status":true,"error_code":null,"error_message":"","data":{
"id":"3","client_id":"1004","hash":"0ff030da0d23","name":"Ubersmith DE 2.3.0 - Client Manager.pdf",
"type":"application\/pdf","contents":"base64 encoded data","public":"0","created":"1358889741","modified":"1358889741"}}

## uber.file_list - Get a List of Files

This method is used to return a list of files.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| id | File ID | | | |
| client_id | Client ID | | | |
| hash | | | | |
| name | | | | |
| type | | | | |
| public | | | | |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{"status":true,"error_code":null,"error_message":"","data":{
"3":{"id":"3","client_id":"1004","hash":"0ff030da0d23","name":"Ubersmith DE 2.3.0 - Client
Manager.pdf","type":"application\/pdf","public":"0","created":"1358889741","modified":"1358889741"}
}}
```

## uber.file_update - Update a file

This method is used to update a file.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| id | File ID | | | |
| client_id | Client ID | | | |
| hash | | | | |
| name | | | | |
| type | | | | |
| contents | base64 encoded data | | | |
| public | | | | |

**Example Output**

```
{"status":true,"error_code":null,"error_message":"","data":"true"}
```

## uber.forgot_pass - Send a Password Reset

Sends a reset password link to the email address on file.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| user_login | Client Login | | | Client's login name or ID |
| email | Email Address | | | The client, contact or admin's email address. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.language_add - Add a New Language

This method is used to add a new language to Ubersmith.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| locale | Locale | Y | | The language locale. Must be a valid locale from uber.language_list |
| name | Name | | | The language name. Defaults to the name in the language file. |
| name_convention | Name Convention | | | The language name convention. Defaults to the name convention in the language file.<br>Options: 0 = Firstname Lastname, 1 = Lastname Firstname, 2 = LastnameFirstname |
| date_format | Date Format | | | The language date format. Defaults to the date format in the language file.<br>Options: M/d/Y, d/M/Y, M/d/y, d/M/y, m/d/Y, d/m/Y, m/d/y, d/m/y, M-d-Y, d-M-Y, M-d-y, d-M-y, m-d-Y, d-m-Y, m-d-y, d-m-y, Y-m-d |
| description | Description | | | An optional language description |
| default | Default | | 0 | Whether to make the new language the default language of Ubersmith.<br>Options: 1 = Yes, 0 = No |
| status | Status | | 1 | Whether the language should be active or inactive.<br>Options: 1 = Active, 0 = Inactive |

**Example Output**

```
{
  "data": "10",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.language_list - Language List

This method is used to list the languages that Ubersmith is aware of.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| lang_id | Language Id | | | Find the language by numeric ID |
| locale | Locale | | | Find the language by locale |
| name | Name | | | Find the language by name |
| default | Default | | | Only fetch the default language |
| status | Status | | | Filter languages by status.<br>Options: 1 = Active, 0 = Inactive |
| installed | Installed | | | Filter languages by installed status.<br>Options: 1 = Installed, 0 = Uninstalled |
| available | Available | | | Get all uninstalled and inactive languages in the list |

**Example Output**

```
{
  "data":{
    "en_US":{
      "lang_id":"1",
      "name":"English(USA)",
      "locale":"en_US",
      "description":"American English",
      "default":"1",
      "date_format":"M\/d\/Y",
      "name_convention":"0",
      "installed":"1",
      "status":"1",
      "name_convention_api": "Firstname Lastname"
    },
    "fr_CA":{
      "lang_id":"2",
      "name":"Fran\u00e7ais canadien",
      "locale":"fr_CA",
      "description":"Fran\u00e7ais canadien",
      "default":"0",
      "date_format":"Y-m-d",
      "name_convention":"0",
      "installed":"0",
      "status":"0",
      "name_convention_api":"Firstname Lastname"
    },
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.language_update - Update Language

This method is used to update an existing language in Ubersmith.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| lang_id | Language ID | | | The numeric language ID. Must specify either lang_id or locale when calling this method |
| locale | Locale | | | The language locale. Must specify either lang_id or locale when calling this method |
| name | Name | | | The language name. |
| name_convention | Name Convention | | | The language name convention.<br>Options: 0 = Firstname Lastname, 1 = Lastname Firstname, 2 = LastnameFirstname |
| date_format | Date Format | | | The language date format.<br>Options: M/d/Y, d/M/Y, M/d/y, d/M/y, m/d/Y, d/m/Y, m/d/y, d/m/y, M-d-Y, d-M-Y, M-d-y, d-M-y, m-d-Y, d-m-Y, m-d-y, d-m-y, Y-m-d |
| description | Description | | | The language description |
| default | Default | | | Whether to make the language the default language of Ubersmith.<br>Options: 1 = Yes, 0 = No |
| status | Status | | | Whether to make the language active or inactive.<br>Options: 1 = Active, 0 = Inactive |

**Example Output**

```
{
  "data":true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.late_fee_schedule_list - List Late Fee Schedules

This method is used to list late fee schedules.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| amount | Amount | | | Amount |
| delay | Days Overdue | | | Days Overdue |
| recur_method | Recur Method | | | Recur Method |
| recur_amount | Recur Amount | | | Recur Amount |
| recur_interval | Recur Interval Days | | | Recur Interval Days |
| class_id | Brand ID | | | Brand ID |
| active | Active | | 1 | 1 = Active Late Fee Schedules, 0 = Deactivated Late Fee Schedules, 2 = All Late Fee Schedules |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "10": {
```

```
        "late_fee_schedule_id": "10",
        "method": "2",
        "amount": "2.5",
        "delay": "30",
        "recur_method": "0",
        "recur_amount": "5",
        "recur_interval": "15",
        "class_id": "1",
        "active": "1"
      }
   },
   "error_message": "",
   "error_code": null,
   "status": true,
}
```

## uber.login_list - List User Logins

This method is used to returns a list of login names defined.

**Example Output**

```
{
   "data": {
     "contacts": {
       "222": {
         "real_name": "Johnny Ubersmith",
         "prefer_lang": "0",
         "email_name": "",
         "priority": "0",
         "contact_id": "222",
         "rwhois_contact": "0",
         "audit_tickets": "0",
         "password_changed": "0",
         "client_id": "1212",
         "phone": "",
         "password_timeout": "0",
         "email_domain": "",
         "description": "Primary Contact",
         "access": "a:1:{s:13:\"billing_email\";s:4:\"edit\";}",
         "password": null,
         "login": "",
         "email": ""
      }
   },
   "error_message": "",
   "error_code": null,
   "status": true
}
```

## uber.mail_get - Get an Email From the Log

This method is used to retrieve data for a specific stored email from the mail log.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| mail_id | Mail ID | Y | | |

**Example Output**

```
{
  "data": {
    "body": "This email is being sent in order to inform you that a new invoice has been generated for your account.  Please
log in to view and/or pay it online.\r\n\r\nThank you.",
    "header": "From: Account Representative <support@ubersmith.com>",
    "time": "1312231187",
    "recipient": "\"Tes<b>tes</b> Client\" <support@ubersmith.com>",
    "subject": "Your account is past due",
    "mail_id": "2881",
    "success": "1",
    "clientid": "1011"
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.mail_list - Access the Mail Log

This method is used to retrieve data from the mail log.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| user_login | Client's login name or id number | | | |
| begin | Begin Time | | | A Unix timestamp. If empty, output will begin from the earliest event. |
| end | End Time | | | A Unix timestamp. If empty, output will end at the most recent event. |
| searchfor | Search Term | | | Used to return only those events that include a specific keyword. |
| searchin | Field that should be used by the 'searchfor' variable. | | | Possible values: subject, body, or recipient. |
| all | Include Message Header and Body | | 0 | Include full email header and body. |
| order_by | Order By | | time | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | 100 | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "99": {
      "header": "From: uber-test <support@ubersmith.com>",
      "time": "1242317909",
      "recipient": "Test Client <support@ubersmith.com>",
      "subject": "overdue invoice from uber-test",
      "mail_id": "99",
      "success": "1",
      "clientid": "1011"
    }
  },
  "error_message": "",
```

```
    "error_code": null,
    "status": true
}
```

## uber.message_list - List Message Board Messages

This method is used to retrieve the messages from Ubersmith's internal message board.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| brand_id | Brand ID | Y | | Return only messages for the specified brand. |
| public | Public-only Flag | | 0 | If set to 1, only return messages which are public (client viewable). |
| order_by | Order By | | | Field to order messages by. Possible fields are: docid, date, title, content, who, author, public, brand_id |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |

**Example Output**

```
{
  "data": {
    "1": {
      "docid": "1",
      "public": "0",
      "author": "ubersmith",
      "title": "Welcome to ubersmith.",
      "class_id": "0",
      "date": "1128440472",
      "content": "Welcome to Ubersmith",
      "who": "ubersmith"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.metadata_bulk_get - Bulk Get Metadata Values

This method is used to bulk retrieve the value of a single metadata item for every client, service, contact, ticket or device.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| variable | Metadata Variable Name | Y | | |
| meta_type | Metadata Object Type | Y | | Possible values:<br>client<br>service<br>contact<br>ticket<br>device |
| brand_id | Brand ID | | | |

**Example Output**

```
{
  "data": {
    "1092": "",
    "1039": ""
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.metadata_field_add - Add Metadata Field

This method is used to add a new metadata field.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| meta_type | Metadata Config Type | Y | | Possible values: client, contact, pack, facility, zone, cage, row, rack, device, ticket, connection. Only required if metagroup_id is not specified or is 0. |
| class_id | Brand ID | | system default | Brand to use for metadata config item, only required if metagroup_id is not specified or is 0. |
| metagroup_id | Metadata Group ID | | | The metadata group to add this config to. Should be the numeric metadata group id. |
| label | Label | Y | | The display label for the metadata config item |
| variable | Variable Name | Y | | The machine name of the metadata config. Has to be unique per meta_type. |
| default_value | Default value | | | The default value for new metadata values of this field |
| client_access | Client Access Level | | 0 | Set the client access. 0 = No access, 1 = View, 2 = Edit. |
| type | Type | Y | | Possible values: select, select_multiple, text, textarea, checkbox, date |
| required | Required | | false | Whether to require this value when saving metadata |
| unique | Unique | | false | Whether to enforce uniqueness when saving values of this metadata field |
| show_in_list | Show in Lists | | false | Whether to show this metadata field in lists of this meta_type. |
| editable | Editable Type | | 0 | What type of editing to allow for this metadata field. -1 = Not Editable, 0 = Editable, 1 = Inline Editing |
| i18n | Internationalization Options | | | The internationalized config for this metadata field. Substitute lang_id with the numeric language id or language locale, e.g. en_US. Options for all fields are:<br><br>i18n[lang_id][label]<br>i18n[lang_id][default_value]<br><br>For type = select:<br><br>i18n[lang_id][options]<br><br>For type = text:<br><br>i18n[lang_id][size]<br><br>For type = textarea:<br><br>i18n[lang_id][rows]<br>i18n[lang_id][cols] |
| regular_expression_match | Regular Expression Match | | | If type = text, the regular expression to match when saving values |
| regular_expression_help | Regular Expression Help Text | | | If type = text and regular_expression_match specified, the help text to display to the user |

**Example Output**

```
{
  "data":{
    "metaconf_id":"123"
  },
  "error_message": "",
```

```
    "error_code": null,
    "status": true
}
```

## uber.metadata_field_delete - Delete Metadata Field

This method is used to delete an existing metadata field.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| metaconf_id | Metadata Config ID | Y | | The numeric metadata id to delete |

**Example Output**

```
{
    "data":true,
    "error_message": "",
    "error_code": null,
    "status": true
}
```

## uber.metadata_field_list - Get Metadata Field Definitions

This method is used to retrieve a list of matching metadata fields.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| metaconf_id | Metadata Config ID | | | If specified, a single record matching the ID is returned and all other params are ignored. |
| class_id | Brand ID | | system default | The class_id to filter metadata fields on |
| meta_type | Metadata Type | | | Required if metaconf_id is not specified. Possible values: client, contact, pack, facility, zone, cage, row, rack, device, ticket, connection |
| variable | Metadata Variable Name | | | Variable name to search for. |

**Example Output**

```
{
  "data": {
    "44": {
      "metaconf_id": "44",
      "variable": "hide_address_in_whois"
      "meta_type": "client",
      "class_id": "1",
      "metagroup_id": "14",
      "label": "RWhois Visibility",
      "suffix": "",
      "type": "select",
      "default_val": "0",
      "options": "\"\",\"0:Display Full Details\",\"1:Hide All Details\",\"2:ARIN Customer Privacy Guidelines\"",
      "size": "0",
      "cols": "0",
      "rows": "0",
      "priority": "3",
      "client_access": "2",
      "regular_expression_match": "",
      "regular_expression_help": "",
      "unique": "0",
      "required": "0",
      "show_in_list": "0",
      "editable": "0",
      "show_if_empty": "0",
      "metagroup_name": "Name"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.metadata_field_update - Update Metadata Field

This method is used to update an existing metadata field.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| metaconf_id | Metadata Config ID | Y | | The numeric metadata id |
| metagroup_id | Metadata Group ID | | | The metadata group to add this config to. Should be the numeric metadata group id. |
| label | Label | | | The display label for the metadata config item |
| variable | Variable Name | | | The machine name of the metadata config. Has to be unique per meta_type. |
| default_value | Default value | | | The default value for new metadata values of this field |
| client_access | Client Access Level | | 0 | Set the client access. 0 = No access, 1 = View, 2 = Edit. |
| type | Type | | | Possible values: select, select_multiple, text, textarea, checkbox, date |
| required | Required | | false | Whether to require this value when saving metadata |
| unique | Unique | | false | Whether to enforce uniqueness when saving values of this metadata field |
| show_in_list | Show in Lists | | false | Whether to show this metadata field in lists of this meta_type. |
| editable | Editable Type | | 0 | What type of editing to allow for this metadata field. -1 = Not Editable, 0 = Editable, 1 = Inline Editing |
| i18n | Internationalization Options | | | The internationalized config for this metadata field. Substitute lang_id with the numeric language id or language locale, e.g. en_US. Options for all fields are:<br><br>i18n[lang_id][label]<br>i18n[lang_id][default_value]<br><br>For type = select:<br><br>i18n[lang_id][options]<br><br>For type = text:<br><br>i18n[lang_id][size]<br><br>For type = textarea:<br><br>i18n[lang_id][rows]<br>i18n[lang_id][cols] |
| regular_expression_match | Regular Expression Match | | | If type = text, the regular expression to match when saving values |
| regular_expression_help | Regular Expression Help Text | | | If type = text and regular_expression_match specified, the help text to display to the user |

**Example Output**

```
{
  "data":true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.metadata_get - Get Metadata Values

This method is used to retrieve metadata values.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| id | Object ID | Y | | ID of Object (Client, Service or Device) |
| metagroup_id | Metadata Group ID | | | If specified, only items from this group are returned. Use 0 to return ungrouped items. |
| meta_type | Metadata Object Type | | | Possible values; client, pack or device. Only required if metagroup_id is not specified or is 0. |
| brand_id | Brand ID | | system default | Brand to search for items, only required if metagroup_id is not specified or is 0. |
| client_access | Client Access Level | | | Only return items with a client access level greater or equal to the specified value: 0 = No access, 1 = View, 2 = Edit |

**Example Output**

```
{
  "data": {
    "44": {
      "required": "0",
      "size": "0",
      "prefix": "RWhois Visibility",
      "metagroup_id": "14",
      "default": 0,
      "editable": "0",
      "client_access": "2",
      "priority": "3",
      "class_id": "1",
      "id": "1018",
      "default_val": "0",
      "value": "2",
      "show_in_list": "0",
      "type": "select",
      "rows": "0",
      "cols": "0",
      "suffix": "",
      "metaconf_id": "44",
      "meta_type": "client",
      "options": "\"\",\"0:Display Full Details\",\"1:Hide All Details\",\"2:ARIN Customer Privacy Guidelines\"",
      "variable": "hide_address_in_whois"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.metadata_group_add - Add Metadata Group

This method is used to add a metadata group.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| meta_type | Metadata Group Type | Y | | Possible values: client, contact, pack, facility, zone, cage, row, rack, device, ticket, connection |
| class_id | Brand ID | | system default | Brand to use for metadata group |
| name | Display Name | Y | | The display name to use for the metadata group |
| variable | Variable Name | Y | | The machine name of the metadata group. Has to be unique per meta_type. |
| priority | Priority | | lowest priority | Controls how high or low the metadata group will appear on the config page in Setup & Admin. If none is provided, will appear at the bottom of the list. |
| i18n | Internationalization Options | | | The internationalized config for this metadata group. Substitute lang_id with the numeric language id or language locale, e.g. en_US. Options are:<br><br>i18n[lang_id][name] |

**Example Output**

```
{
  "data":{
    "metagroup_id":"123"
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.metadata_group_delete - Delete Metadata Group

This method is used to delete an existing metadata group.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| metagroup_id | Metadata Group ID | Y | | The numeric metadata group id to delete |

**Example Output**

```
{
  "data":true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.metadata_group_list - List Metadata Groups

This method is used to list the existing metadata groups.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| meta_type | Metadata Config Type | | | The type of metadata group to filter on. Possible values: client, contact, pack, facility, zone, cage, row, rack, device, ticket, connection |
| metagroup_id | Metadata Group ID | | | The metadata group to find. Should be the numeric metadata group id. |
| class_id | Brand ID | | system default | Brand to filter metadata groups. Only required if metagroup_id is not specified or is 0. |

**Example Output**

```
{
  "data":{
    "1":{
      "metagroup_id":"1",
      "meta_type":"client",
      "class_id":"1",
      "name":"Client Data",
      "priority":"1",
      "variable":"client_data"
    },
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.metadata_group_update - Update Metadata Group

This method is used to update an existing metadata group.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| metagroup_id | Metadata Group ID | Y | | Possible values: client, contact, pack, facility, zone, cage, row, rack, device, ticket, connection |
| name | Display Name | | | The display name to use for the metadata group |
| variable | Variable Name | | | The machine name of the metadata group. Has to be unique per meta_type and class_id. |
| priority | Priority | | | Controls how high or low the metadata group will appear on the config page in Setup & Admin. If none is provided, will appear at the bottom of the list. |
| i18n | Internationalization Options | | | The internationalized config for this metadata group. Substitute lang_id with the numeric language id or language locale, e.g. en_US. Options are:<br><br>i18n[lang_id][name] |

**Example Output**

```
{
  "data":true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.method_get - Get API Method Details

This method is used to get the details of an API method.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| method_name | Method Name | Y | | |

**Example Output**

```
{
  "data": {
    "title": "Get Metadata Values",
    "method": "uber.metadata_get",
    "output": null,
    "params": [
      {
        "required": true,
        "name": "Object ID",
        "default": null,
        "param": "id",
        "description": "ID of Object (Client, Service or Device)"
      },
      {
        "required": false,
        "name": "Metadata Group ID",
        "default": null,
        "param": "metagroup_id",
        "description": "If specified, only items from this group are returned. Use 0 to return ungrouped items."
      },
      {
        "required": false,
        "name": "Metadata Object Type",
        "default": null,
        "param": "meta_type",
        "description": "Possible values; client, pack or device. Only required if metagroup_id is not specified or is 0."
      },
      {
        "required": false,
        "name": "Brand ID",
        "default": "system default",
        "param": "brand_id",
        "description": "Brand to search for items, only required if metagroup_id is not specified or is 0."
      },
      {
        "required": false,
        "name": "Client Access Level",
        "default": null,
        "param": "client_access",
        "description": "Only return items with a client access level greater or equal to the specified value: 0 = No access, 1
= View, 2 = Edit"
      }
    ],
    "description": "This method is used to retrieve metadata values."
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.method_list - List Available API Methods

This method is used to get a list of all available API methods.

**Example Output**

```
{
  "data": {
    "client.credit_add": "Add an Account Credit"
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.msa_get - Get MSA Details

This method is used to get the details of a specified msa.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| brand_id | Brand ID | | | |
| msa_id | MSA ID | | | |
| format | Format | | json | Return format:<br>json = JSON (default)<br>pdf = PDF |

**Example Output**

```
{
  "data": {
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.notification_add - Add Notification

This method adds a notification.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| name | Notification Name | Y | | System name for notification |
| association | Association | | | An array that contains the notification's associations.<br>A notification can contain multiple associations and each association will contain a list of items.<br>For each item, one or multiple clients (client_id), devices (device_id), services (service_id) and locations (facility_id) can be specified.<br>Examples:<br>An association with a single item specifying a single client: association[0][items][0][clientid]=1001<br>An association with multiple items each specifying a different client:<br>association[0][items][0][clientid]=1001&association[0][items][1][clientid]=1002<br>2 associations each with a single item specifying a different client:<br>association[0][items][0][clientid]=1001&association[1][items][0][clientid]=1002<br>An association with a single item specifying a client with a linked service:<br>association[0][items][0][clientid]=1001&association[0][items][0][service_id]=1 |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| notification_type_id | Notification Type ID | Y | | Notification type identifier |
| status_id | Status | | 1 | Options:<br>  1 = Draft<br>  2 = Active<br>  4 = Resolved<br>  8 = Deactivated |
| make_replies_links | Replies will open Linked Tickets | | | Options: 1 = Yes, put replies into linked tickets; 0 = No, put replies into the main ticket. |
| notify_account_team | Notify Account Team | | | Notify account team. 1 = Yes, 0 = No |
| email_from_default | Default From Address | | | Email address used as the default for new messages |
| q_id | Ticket Department ID | | | Department ID used for created tickets. Overrides notification type. |
| assignment | Ticket Assignment User ID | | | Admin ID of assignee for created tickets. Overrides notification type. |
| classification_id | Ticket Classification ID | | | Classification ID for created tickets. Overrides notification type. |
| priority | Ticket Priority | | | Priority for created tickets. Overrides notification type. |
| impact | Ticket Impact | | | Impact for created tickets. Overrides notification type. |

**Example Output**

```
{
  "data": "509",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.notification_list - List Notifications

This method lists existing notifications filtered by the parameters specified.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| notification_id | Notification ID | | | Notification identifier |
| name | Notification Name | | | System name for notification |
| association | Association | | | |
| notification_type_id | Notification Type ID | | | Notification type identifier |
| status_id | Status | | | Options:<br>  1 = Draft<br>  2 = Active<br>  4 = Resolved<br>  8 = Deactivated<br>  255 = All |
| make_replies_links | Replies will open Linked Tickets | | | Options: 1 = Yes, put replies into linked tickets; 0 = No, put replies into the main ticket. |
| notify_account_team | Notify Account Team | | | Notify account team. 1 = Yes, 0 = No |
| email_from_default | Default From Address | | | Email address used as the default for new messages |
| q_id | Ticket Department ID | | | Department ID used for created tickets. Overrides notification type. |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| assignment | Ticket Assignment User ID | | | Staff user ID of assignee for created tickets. Overrides notification type. |
| classification_id | Ticket Classification ID | | | Classification ID for created tickets |
| priority | Ticket Priority | | | Priority for created tickets |
| impact | Ticket Impact | | | Impact for created tickets |
| created_ts | Created Timestamp | | | Notification initial creation time |
| created_by | Created By | | | Party that created the notification |
| updated_ts | Updated Timestamp | | | Notification last update time |
| updated_by | Updated By | | | Party that last updated the notification |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "1": {
      "notification_id": "1",
      "notification_type_id": "1",
      "name": "2014-08-01.mktg.newofferings_r3",
      "status_id": "1",
      "make_replies_links": "0",
      "notify_account_team": "0",
      "email_from_default": "",
      "assignment": "0",
      "ticket_id": null,
      "q_id": "1",
      "classification_id": "0",
      "priority": "1",
      "impact": "1",
      "association": [{"search_query":"%5B%7B%22join%22%3A%22AND%22%2C%22display%22%3Atrue%2C%22what%22%3A%22client%22%2C%22fie
ld%22%3A%22class_id%22%2C%22type%22%3A%22is%22%2C%22string%22%3A%221%22%7D%5D","items":[{"clientid":"1001"}]}],
      "created_ts": "1406932417",
      "created_by": "admin",
      "updated_ts": "1407362456",
      "updated_by": "admin",
      "department": "General Support",
      "classification": null,
      "assignment_name": null,
      "type_name": "Sales Specials",
      "class_id": "1",
      "default_email_from_default": "",
    }
  }
}
```

## uber.notification_message_add - Add Notification Message

This method adds a notification message and can allow a notification to be scheduled for delivery.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| notification_id | Notification ID | Y | | Notification identifier |
| notification_message_type_id | Notification Message Type ID | Y | | Options:<br>1 = Initial<br>2 = Update<br>3 = Resolution |
| email_subject | Email Subject | | | Notification email subject value |
| email_from | Email From | | | Notification email from header value |
| email_body | Email Body | | | Notification email body |
| send_date | Send Date | | | Notification message scheduled send date. POSIX epoch based integer. Defaults to now. |

**Example Output**

```
{
  "data": "7002",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.notification_message_list - List Notification Messages

This method lists notification messages filtered by the parameters specified.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| notification_message_id | Notification Message ID | | | Notification message identifier |
| notification_id | Notification ID | | | Notification identifier |
| notification_message_type_id | Notification Message Type ID | | | Options:<br>1 = Initial<br>2 = Update<br>3 = Resolution |
| email_subject | Email Subject | | | Notification email subject value |
| email_from | Email From | | | Notification email from header value |
| email_body | Email Body | | | Notification email body |
| created_ts | Created Timestamp | | | Message initial creation time |
| created_by | Created By | | | Party that created the message |
| updated_ts | Updated Timestamp | | | Message last update time |
| updated_by | Updated By | | | Party that last updated the message |
| send_date | Send Date | | | Scheduled send date. POSIX epoch based integer. |
| send_date_max | Send Date Before | | | Scheduled send date is before specified time. |
| send_date_min | Send Date After | | | Scheduled send date is after specified time. |
| status_id | Status ID | | | Options (bitwise):<br>255 = All<br>1 = Pending<br>2 = Queued<br>4 = Sent |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "1": {
      "notification_message_id": "1",
      "notification_id": "1",
      "notification_message_type_id": "1",
      "email_subject": "August Deals!",
      "email_from": "noreply@sales",
      "email_body": "Short message \n Thanks",
      "sms_body": "",
      "send_date": "1407362955",
      "status_id": "1",
      "created_ts": "1407362955",
      "created_by": "admin",
      "updated_ts": "0",
      "updated_by": ""
    }
  }
}
```

## uber.notification_message_update - Edit Notification Message

This method edits an existing notification message.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| notification_message_id | Notification Message ID | Y | | Notification message identifier |
| notification_message_type_id | Notification Message Type ID | | | Options:<br>1 = Initial<br>2 = Update<br>3 = Resolution |
| email_subject | Email Subject | | | Notification email subject value |
| email_from | Email From | | | Notification email from header value |
| email_body | Email Body | | | Notification email body |
| send_date | Send Date | | | Notification message scheduled send date. POSIX epoch based integer. If set but empty, defaults to now. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.notification_recipient_add - Manually Add Notification Recipient

This method manually adds a unique recipient to a notification.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| notification_id | Notification ID | Y | | Notification identifier |
| contact_id | Contact ID | | | Recipient contact ID |
| person_id | Admin ID | | | Recipient admin ID |
| status | Status | | 1 | Only recipients with an active status will recieve the notification.<br>Options: 1 = Active, 0 = Inactive |
| email | Email | | | Recipient email address |

**Example Output**

```
{
  "data": "9179",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.notification_recipient_list - List Notification Recipients

This method lists notification recipients filtered by the specified parameters.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| notification_recipient_id | Notification Recipient ID | | | |
| notification_id | Notification ID | | | Notification identifier |
| contact_id | Contact ID | | | Recipient contact ID |
| person_id | Admin ID | | | Recipient admin ID |
| status | Status | | | Only recipients with an active status will recieve the notification.<br>Options: 1 = Active, 0 = Inactive |
| email | Email | | | Recipient email address |
| manually_added | Manually Added | | | Internal flag indicating the recipient was not added through automatic subscription |
| created_by | Created By | | | Party that created the record |
| created_ts | Created Timestamp | | | Record creation time |
| created_ts_max | Created Timestamp Before | | | Record created before specified timestamp |
| created_ts_min | Created Timestamp After | | | Record created after specified timestamp |
| updated_by | Updated By | | | Party that last updated the record |
| updated_ts | Updated Timestamp | | | Record last update time |
| updated_ts_max | Updated Timestamp Before | | | Record updated before specified timestamp |
| updated_ts_min | Updated Timestamp After | | | Record updated after specified timestamp |

**Example Output**

```
{
  "status": true,
  "error_code": null,
```

```
      "error_message": "",
      "data": {
        "3": {
          "notification_recipient_id": "3",
          "notification_id": "1",
          "status": "1",
          "contact_id": "10",
          "person_id": null,
          "email": "jmeld@sprockets.mwg",
          "mobile_phone": "",
          "phone": "",
          "manually_added": "1",
          "created_ts": "1407422565",
          "created_by": "admin",
          "updated_ts": "0",
          "updated_by": "",
          "company": "Spacely Cogs",
          "name": "John Eldridge",
          "client_id": "1009",
          "notification_name": "2014-08-01.mktg.newofferings_r3"
        },
        "4": {
          "notification_recipient_id": "4",
          "notification_id": "1",
          "status": "1",
          "contact_id": null,
          "person_id": "2",
          "email": "steph@ubersmith.com",
          "mobile_phone": "",
          "phone": "",
          "manually_added": "0",
          "created_ts": "1407422572",
          "created_by": "admin",
          "updated_ts": "0",
          "updated_by": "",
          "company": "",
          "name": "Steph Staff",
          "client_id": null,
          "notification_name": "2014-08-01.mktg.newofferings_r3"
        }
      }
    }
```

## uber.notification_recipient_log_list - List Notification Recipient Logs

This method lists notification recipient logs filtered by the specified parameters.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| notification_recipient_log_id | Notification Recipient Log ID | | | |
| notification_message_id | Notification Message ID | | | |
| notification_recipient_id | Notification Recipient ID | | | |
| sent_to | Sent To | | | Notification transmission specific destination |
| status_id | Status ID | | | Notification transmission status. Options:<br>255 = All<br>1 = Pending<br>2 = Processing<br>4 = Success<br>8 = Failure |
| result | Result | | | Notification result detail |
| result_ts | Result Timestamp | | | Notification transmission timestamp |
| result_ts_max | Result Timestamp Before | | | Notification transmission timestamp earlier than specified timestamp |
| result_ts_min | Result Timestamp After | | | Notification transmission timestamp later than specified timestamp |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "1": {
      "notification_recipient_log_id": "1",
      "notification_message_id": "1",
      "notification_recipient_id": "4",
      "method_id": "1",
      "sent_to": "steph@ubersmith.com",
      "status_id": "4",
      "result": "",
      "result_ts": "1407878129",
      "company": "",
      "name": "Steph Staff",
      "contact_id": null,
      "person_id": "2",
      "client_id": null
    },
    "2": {
      "notification_recipient_log_id": "2",
      "notification_message_id": "2",
      "notification_recipient_id": "4",
      "method_id": "1",
      "sent_to": "steph@ubersmith.com",
      "status_id": "4",
      "result": "",
      "result_ts": "1407878129",
      "company": "",
      "name": "Steph Staff",
      "contact_id": null,
      "person_id": "2",
      "client_id": null
    }
  }
}
```

## uber.notification_recipient_populate - Notification Recipient Populate

This method creates all recipient records for a notification based on existing subscriptions.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| notification_id | Notification ID | Y | | |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.notification_recipient_update - Edit a Notification Recipient

This method edits a notification recipient.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| notification_recipient_id | Notification Recipient ID | Y | | |
| status | Status | | | Only recipients with an active status will recieve the notification.<br>Options: 1 = Active, 0 = Inactive |
| email | Email | | | Recipient email address |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.notification_type_add - Add a Notification Type

This method adds a notification type.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| name | Notification Type Name | Y | | System name for notification type |
| status | Status | | 1 | Options: 1 = Active, 0 = Inactive |
| make_replies_links | Replies will open Linked Tickets | | | Options: 1 = Yes, put replies into linked tickets; 0 = No, put replies into the main ticket. |
| notify_account_team | Notify Account Team | | | Notify account team. Options: 1 = Yes, 0 = No |
| email_from_default | Default From Address | | | Email address used as the default for new messages |
| class_id | Brand ID | Y | 0 | Brand ID |
| q_id | Ticket Department ID | | | Department ID used for created tickets |
| assignment | Ticket Assignment User ID | | | Staff user ID of assignee for created tickets |
| classification_id | Ticket Classification ID | | | Classification ID for created tickets |
| priority | Ticket Priority | | | Priority for created tickets |
| impact | Ticket Impact | | | Impact for created ticketss |

**Example Output**

```
{
  "data": "597",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.notification_type_get - Get Notification Type Details

This method retrieves the details of a notification type.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| notification_type_id | Notification Type ID | Y | | Notification type identifier |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "notification_type_id": "5",
    "name": "Power Restored",
    "status": "0",
    "email_from_default": "",
    "class_id": "1",
    "q_id": "1",
    "priority": "1",
    "impact": "1",
    "assignment": "0",
    "classification_id": "0"
  }
}
```

## uber.notification_type_list - List Notification Types

This method lists the notification types registered.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| name | Notification Type Name | | | System name for notification type |
| status | Status | | | Options: 1 = Active, 0 = Inactive |
| make_replies_links | Replies will open Linked Tickets | | | Options: 1 = Yes, put replies into linked tickets; 0 = No, put replies into the main ticket. |
| notify_account_team | Notify Account Team | | | Notify account team. Options: 1 = Yes, 0 = No |
| email_from_default | Default From Address | | | Email address used as the default for new messages |
| class_id | Brand ID | | 0 | Brand ID |
| q_id | Ticket Department ID | | | Department ID used for created tickets |
| assignment | Ticket Assignment User ID | | | Staff user ID of assignee for created tickets |
| classification_id | Ticket Classification ID | | | Classification ID for created tickets |
| priority | Ticket Priority | | | Priority for created tickets |
| impact | Ticket Impact | | | Impact for created ticketss |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "1": {
      "notification_type_id": "1",
      "name": "Power Outage",
      "status": "1",
      "make_replies_links": "0",
      "notify_account_team": "1",
```

```
      "email_from_default": "",
      "class_id": "1",
      "q_id": "1",
      "priority": "1",
      "impact": "1",
      "assignment": "0",
      "classification_id": "0"
    },
    "5": {
      "notification_type_id": "5",
      "name": "Power Restored",
      "status": "0",
      "make_replies_links": "1",
      "notify_account_team": "0",
      "email_from_default": "",
      "class_id": "1",
      "q_id": "1",
      "priority": "1",
      "impact": "1",
      "assignment": "0",
      "classification_id": "0"
    }
  }
}
```

## uber.notification_type_subscriber_add - Add Subscriber to a Notification Type

This method adds a subscriber to notifications of a certain type.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| notification_type_id | Notification Type ID | Y | | Notification Type |
| contact_id | Contact ID | | | Subscriber contact ID |
| person_id | Admin ID | | | Subscriber admin ID |
| override_email | Override email | | | Email address to use. Overrides contact default email address. |

**Example Output**

```
{
  "data": "131",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.notification_type_subscriber_delete - Unsubscribe From a Notification Type

This method unsubscribes a contact or admin from notifications of a certain type.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| notification_type_subscriber_id | Notification Type Subscriber ID | | | Subscription ID |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.notification_type_subscriber_list - List Notification Type Subscriptions

This method lists notification subscriptions filtered by the parameters supplied.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| notification_type_id | Notification Type ID | | | Notification Type ID |
| contact_id | Contact ID | | | Subscriber Contact ID |
| person_id | Admin ID | | | Subscriber Admin ID |
| override_email | Override email | | | Override email address. |
| created_by | Created By | | | Party that created the record |
| created_ts | Created Timestamp | | | Record creation time |
| created_ts_max | Created Timestamp Before | | | Record created before specified timestamp |
| created_ts_min | Created Timestamp After | | | Record created after specified timestamp |
| updated_by | Updated By | | | Party that last updated the record |
| updated_ts | Updated Timestamp | | | Record last update time |
| updated_ts_max | Updated Timestamp Before | | | Record updated before specified timestamp |
| updated_ts_min | Updated Timestamp After | | | Record updated after specified timestamp |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "1": {
      "notification_type_subscriber_id": "1",
      "notification_type_id": "1",
      "contact_id": "12",
      "person_id": null,
      "do_email": "1",
      "override_email": "",
      "do_sms": "0",
      "override_mobile_phone": "",
      "do_phone": "0",
      "override_phone": "",
      "created_ts": "1407422572",
      "created_by": "admin",
      "updated_ts": "0",
      "updated_by": "",
```

```
        "client_id": "1011",
        "type_name": "Sales Specials",
        "company": "Styr-Oh-Foam",
        "name": "Chelsea Styre",
        "doing": "0",
        "email": "noc@styr-o",
        "main_email": "noc@styr-o",
        "mobile_phone": "",
        "main_mobile_phone": "",
        "phone": "",
        "main_phone": ""
      }
    }
}
```

## uber.notification_type_subscriber_update - Edit a Notification Type Subscription

This method edits a subscription for a notification type.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| notification_type_subscriber_id | Notification Type Subscriber ID | Y | | Notification Type Subscriber ID (subscription identifier) |
| override_email | Override email | | | Email address to use. Overrides contact default email address. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.notification_type_update - Edit Notification Type

This method edits an existing notification type.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| notification_type_id | Notification Type ID | Y | | Notification type identifier |
| name | Notification Type Name | | | System name for notification type |
| status | Status | | | Options: 1 = Active, 0 = Inactive |
| make_replies_links | Replies will open Linked Tickets | | | Options: 1 = Yes, put replies into linked tickets; 0 = No, put replies into the main ticket. |
| notify_account_team | Notify Account Team | | | Notify account team. Options: 1 = Yes, 0 = No |
| email_from_default | Default From Address | | | Email address used as the default for new messages |
| class_id | Brand ID | | 0 | Brand ID |
| q_id | Ticket Department ID | | | Department ID used for created tickets |
| assignment | Ticket Assignment User ID | | | Staff user ID of assignee for created tickets |
| classification_id | Ticket Classification ID | | | Classification ID for created tickets |
| priority | Ticket Priority | | | Priority for created tickets |
| impact | Ticket Impact | | | Impact for created ticketss |

**Example Output**

```
{
```

```
    "data": true,
    "error_message": "",
    "error_code": null,
    "status": true
}
```

## uber.notification_update - Edit Notification

This method edits a notification.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| notification_id | Notification ID | Y | | Notification identifier |
| name | Notification Name | | | System name for notification type |
| association | Association | | | An array that contains the notification's associations.<br>A notification can contain multiple associations and each association will contain a list of items.<br>For each item, one or multiple clients (client_id), devices (device_id), services (service_id) and locations (facility_id) can be specified.<br>Examples:<br>An association with a single item specifying a single client:<br>association[0][items][0][clientid]=1001<br>An association with multiple items each specifying a different client:<br>association[0][items][0][clientid]=1001&association[0][items][1][clientid]=1002<br>2 associations each with a single item specifying a different client:<br>association[0][items][0][clientid]=1001&association[1][items][0][clientid]=1002<br>An association with a single item specifying a client with a linked service:<br>association[0][items][0][clientid]=1001&association[0][items][0][service_id]=1 |
| notification_type_id | Notification Type ID | | | Notification type identifier |
| status_id | Status | | | Options:<br>  1 = Draft<br>  2 = Active<br>  4 = Resolved<br>  8 = Deactivated |
| make_replies_links | Replies will open Linked Tickets | | | Options: 1 = Yes, put replies into linked tickets; 0 = No, put replies into the main ticket. |
| notify_account_team | Notify Account Team | | | Notify account team. 1 = Yes, 0 = No |
| email_from_default | Default From Address | | | Email address used as the default for new messages |
| q_id | Ticket Department ID | | | Department ID used for created tickets. Defaults to notification type value. |
| assignment | Ticket Assignment User ID | | | Staff user ID of assignee for created tickets. Defaults to notification type value. |
| classification_id | Ticket Classification ID | | | Classification ID for created tickets. Defaults to notification type value. |
| priority | Ticket Priority | | | Priority for created tickets. Defaults to notification type value. |
| impact | Ticket Impact | | | Impact for created tickets. Defaults to notification type value. |

**Example Output**

{

```
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.password_reset - Reset Password

Reset a user password given a valid reset code and new password.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| reset_code | Reset Code | Y | | Password reset code previously generated by the system. |
| new_pass | New Password | Y | | New password. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.plan_option_add - Add Plan Option

This method is used to add an plan option to an existing upgrade.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| pu_id | Plan Upgrade ID | Y | | The numeric plan upgrade ID to add the option to. |
| pog_id | Plan Option Group ID | | | The numeric plan option group ID to add the option to. Leave blank or use 0 for Ungrouped. |
| po_description | Name | | | The name of the option. |
| po_desc | Description | | | The description of the option. |
| po_priority | Priority | | | The numeric priority of where the plan option should appear in the list. Lower numbers will end up at the top of the list. Higher numbers will appear lower in the list. |
| po_data | Extra Data | | | Any extra data for this plan option. Must be an array of key/value pairs. Format: po_data[field_name]=field_value |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| pricing | Pricing | | | The pricing matrix.<br>Format:<br>  pricing[currency_id][period_id][type_method]=modifier_id<br>  pricing[currency_id][period_id][type]=value<br><br>Options for "currency_id":<br>  0 = Dollar<br><br>Period IDs can be brand specific or global. The global options for "period_id" are:<br>  0 = One Time Fee<br>  1 = Monthly<br>  3 = Quarterly<br>  6 = Semi-Annually<br>  12 = Annually<br><br>Options for "type": price_method, price, setup_method, setup, cost_method, cost, setup_cost_method, setup_cost<br><br>Options for "modifier_id":<br>  0 = Inherit<br>  1 = Override<br>  2 = Modify By Amount<br>  3 = Modify By Percent<br><br>Example:<br>  pricing[0][0][price_method]=0<br>  pricing[0][0][price]=10.00<br>  pricing[0][3][price_method]=1<br>  pricing[0][3][price]=22.00 |
| i18n | Internationalization | | | The internationalization config for this option.<br>Format: i18n[lang_id][po_description]=Serveur<br>Available Field: po_description, po_desc |

**Example Output**

```
{
  "data": "1",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.plan_option_assign - Assign Plan Option

This method is used to assign and unassign a plan option to a service plan.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| po_id | Plan Option ID | Y | | |
| plan_id | Service Plan ID | Y | | |
| po_val | Value | | | The value of the option. |
| po_data | Extra Data | | | Any extra data for this plan option in the service plan. Must be an array of key/value pairs.<br>Format: po_data[field_name]=field_value |
| assign | Assign | Y | | Flag to either assign or unassign the plan option to the service plan.<br>Options: 1 = Assign, 0 = Unassign |
| pricing | Pricing | | | The pricing matrix.<br>Format:<br>  pricing[period_id][type_method]=modifier_id<br>  pricing[period_id][type]=value<br><br>Period IDs can be brand specific or global. The global options for "period_id" are:<br>  0 = One Time Fee<br>  1 = Monthly<br>  3 = Quarterly<br>  6 = Semi-Annually<br>  12 = Annually<br><br>Options for "type": price_method, price, setup_method, setup, cost_method, cost, setup_cost_method, setup_cost<br><br>Options for "modifier_id":<br>  0 = Inherit<br>  1 = Override<br>  2 = Modify By Amount<br>  3 = Modify By Percent<br><br>Example:<br>  pricing[0][price_method]=0<br>  pricing[0][price]=10.00<br>  pricing[3][price_method]=1<br>  pricing[3][price]=22.00 |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.plan_option_group_add - Add Plan Option Group

This method is used to add an plan option group to an existing plan upgrade.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| pu_id | Plan Upgrade ID | Y | | The numeric plan upgrade ID to add the plan option group to. |
| pog_name | Name | Y | | The name of the plan option group |
| pog_desc | Description | | | The description of the plan option group |
| pog_status | Status | | | The status of the plan option group. Options: 1 = Active, 0 = Inactive |
| pog_priority | Priority | | | The numeric priority of where the plan option group should appear in the list. Lower numbers will end up at the top of the list. Higher numbers will appear lower in the list. |
| pog_data | Extra Data | | | Any extra data for this plan option group. Must be an array of key/value pairs. Format: pog_data[custom]=text |
| i18n | Internationalization | | | The internationalization config for this plan option group. Format: i18n[lang_id][pog_name]=Serveur Available Field: pog_name, pog_desc |

**Example Output**

```
{
  "data": "1",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.plan_option_group_assign - Assign Plan Option Group

This method is used to assign and unassign a plan option group to a service plan.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| pog_id | Plan Option Group ID | Y | | |
| pu_id | Plan Upgrade ID | | | Required if assigning the Ungrouped group. |
| plan_id | Service Plan ID | Y | | |
| assign | Assign | Y | | Flag to either assign or unassign the plan option group to the plan. Options: 1 = Assign, 0 = Unassign |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.plan_option_group_list - List Plan Option Groups

This method is used to list plan option groups.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| pu_id | Plan Upgrade ID | Y | | The numeric plan upgrade ID to fetch the option groups for |
| pog_id | Plan Option Group ID | | | The numeric plan option group ID |
| plan_id | Service Plan ID | | | The numeric service plan ID to fetch the option groups for |
| lang_id | Language ID | | | The numeric language ID to fetch the option groups in |
| include_inactive | Include Inactive Items | | 0 | Whether to include inactive option groups (and options, if all_options is specified).<br>Options: 1 = Yes, 0 = No |
| exclude_empty | Exclude Empty Option Groups | | 0 | Whether to exclude empty option groups that do not have any options associated with them.<br>Options: 1 = Yes, 0 = No |
| all_options | Include Options | | 0 | Whether to include options in the list.<br>Options: 1 = Yes, 0 = No |

**Example Output**

```
{
  "data": {
    "1": {
      "pog_id": "1",
      "pu_id": "2",
      "pog_status": "1",
      "pog_priority": "1",
      "pog_data": {
        "custom": "text"
      },
      "pog_name": "SATA",
      "pog_desc": "SATA drives are great for large storage arrays",
      "pu_name": "Hard Disk #1",
      "pug_name": "Hardware",
      "pu_status": "1",
      "pug_id": "2",
      "pug_status": "1"
      "options": {
        "84": {
          "po_id": "84",
          "pog_id": "10",
          "po_price": "10.00",
          "po_setup": "0.00",
          "po_cost": "0.00",
          "po_setup_cost": "0.00",
          "po_priority": "1",
          "po_status": "1",
          "po_val": "320GB",
          "po_data": {
            "metadata_size": "320GB"
          },
          "po_description": "320GB SATA 7.5k",
          "po_desc": "",
          "pog_name": "SATA",
          "pog_status": "1",
          "pricing": {
            "0": {
              "period": "0",
              "api_label": "one_time",
              "price": "0.00",
              "setup": "0.00",
              "cost": "0.00",
              "setup_cost": "0.00"
            },
            "1":{
              "period": "1",
              "api_label": "monthly",
              "price": "10.00",
              "setup": "0.00",
              "cost": "0.00",
```

```
        "setup_cost": "0.00"
      },
      "3":{
        "period": "3",
        "api_label": "quarterly",
        "price":"0.00",
        "setup":"0.00",
        "cost": "0.00",
        "setup_cost": "0.00"
      },
      "6":{
        "period": "6",
        "api_label": "semi",
        "price": "0.00",
        "setup": "0.00",
        "cost": "0.00",
        "setup_cost": "0.00"
      },
      "12":{
        "period": "12",
        "api_label": "annual",
        "price": "0.00",
        "setup": "0.00",
        "cost": "0.00",
        "setup_cost": "0.00"
      }
      }
      }
      }
    },
    "0": {
      "pog_id": "0",
      "pu_id": "2",
      "pog_status": "1",
      "pog_name": "Ungrouped",
      "pog_data": [],
      "pu_name": "Hard Disk #1",
      "pu_status": "1",
      "pug_id": "2",
      "pug_name": "Hardware",
      "pug_status": "1"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.plan_option_group_update - Update Service Plan Option Group

This method is used to update an option group for an existing service plan.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| pog_id | Plan Option Group ID | Y | | The numeric plan option group ID to update. |
| pog_name | Name | | | The name of the plan option group |
| pog_desc | Description | | | The description of the plan option group |
| pog_status | Status | | | The status of the plan option group.<br>Options: 1 = Active, 0 = Inactive |
| pog_priority | Priority | | | The numeric priority of where the plan option group should appear in the list. Lower numbers will end up at the top of the list. Higher numbers will appear lower in the list. |
| pog_data | Extra Data | | | Any extra data for this plan option group. Must be an array of key/value pairs.<br>Format: pog_data[custom]=text |
| i18n | Internationalization | | | The internationalization config for this plan option group.<br>Format: i18n[lang_id][pog_name]=Serveur<br>Available Field: pog_name, pog_desc |

**Example Output**

```
{
  "data":true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.plan_option_list - List Plan Options

This method is used to list plan options.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| po_id | Plan Option ID | | | The numeric plan option ID |
| pu_id | Plan Upgrade ID | | | The numeric plan upgrade ID to fetch the options for |
| pog_id | Plan Option Group ID | | | The numeric plan upgrade ID to fetch the options for |
| plan_id | Service Plan ID | | | The numeric service plan ID to fetch the options for |
| lang_id | Language ID | | | The numeric language ID to fetch the options in |
| brand_id | Brand ID | | | If specified, return the pricing of the option for the particular brand |
| include_inactive | Include Inactive Items | | 0 | Whether to include inactive options in the list. Options: 1 = Yes, 0 = No |

**Example Output**

```
{
  "data": {
    "1": {
      "po_id": "1",
      "pog_id": "3",
      "po_price": "10.00",
      "po_setup": "0.00",
      "po_cost": "0.00",
      "po_setup_cost": "0.00",
      "po_priority": "1",
      "po_status": "1",
      "po_val": "320GB",
      "po_data": {
        "metadata_size": "320GB"
      },
      "po_description": "320GB SATA 7.5k",
      "po_desc": "",
      "pog_name": "SATA2",
      "pog_status": "1",
      "pricing": {
        "0": {
          "period": "0",
          "api_label": "one_time",
          "price": "0.00",
          "setup": "0.00",
          "cost": "0.00",
          "setup_cost": "0.00"
        },
        "1": {
          "period": "1",
          "api_label": "monthly",
          "price": "10.00",
          "setup_fee": "0.00",
          "cost": "0.00",
          "setup_cost": "0.00"
        },
        "3": {
          "period": "3",
          "api_label": "quarterly",
          "price": "0.00",
          "setup": "0.00",
          "cost": "0.00",
          "setup_cost": "0.00"
        },
        "6": {
          "period": "6",
```

```
        "api_label": "semi",
        "price": "0.00",
        "setup": "0.00",
        "cost": "0.00",
        "setup_cost": "0.00"
      },
      "12": {
        "period": "12",
        "api_label": "annual",
        "price": "0.00",
        "setup": "0.00",
        "cost": "0.00",
        "setup_cost": "0.00"
      }
    }
  }
},
"error_message": "",
"error_code": null,
"status": true
}
```

## uber.plan_option_update - Update Plan Option

This method is used to update a plan option.

| Parameter | Name | Required | Default | Description |
| --- | --- | --- | --- | --- |
| po_id | Plan Option ID | Y | | The numeric plan option ID to update. |
| pog_id | Plan Option Group ID | | | The numeric plan option group ID of the plan option. Use 0 for Ungrouped. |
| po_description | Name | | | The name of the option. |
| po_desc | Description | | | The description of the option. |
| po_priority | Priority | | | The numeric priority of where the plan option should appear in the list. Lower numbers will end up at the top of the list. Higher numbers will appear lower in the list. |
| po_data | Extra Data | | | Any extra data for this plan option. Must be an array of key/value pairs. Format: po_data[field_name]=field_value |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| pricing | Pricing | | | The pricing matrix.<br>Format:<br>  pricing[currency_id][period_id][type_method]=modifier_id<br>  pricing[currency_id][period_id][type]=value<br><br>Options for "currency_id":<br>  0 = Dollar<br><br>Period IDs can be brand specific or global. The global options for "period_id" are:<br>  0 = One Time Fee<br>  1 = Monthly<br>  3 = Quarterly<br>  6 = Semi-Annually<br>  12 = Annually<br><br>Options for "type": price_method, price, setup_method, setup, cost_method, cost, setup_cost_method, setup_cost<br><br>Options for "modifier_id":<br>  0 = Inherit<br>  1 = Override<br>  2 = Modify By Amount<br>  3 = Modify By Percent<br><br>Example:<br>  pricing[0][0][price_method]=0<br>  pricing[0][0][price]=10.00<br>  pricing[0][3][price_method]=1<br>  pricing[0][3][price]=22.00 |
| i18n | Internationalization | | | The internationalization config for this option.<br>Format: i18n[lang_id][po_description]=Serveur<br>Available Field: po_description, po_desc |

**Example Output**

```
{
  "data":true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.plan_upgrade_add - Add Plan Upgrade

This method is used to add a plan upgrade.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| pu_name | Name | Y | | The name of the plan upgrade |
| variable | Variable | Y | | The API variable (machine name) of the plan upgrade |
| pug_id | Plan Upgrade Group ID | | | The numeric plan upgrade group id to add the upgrade to. Leave blank or use 0 for Ungrouped |
| pu_desc | Description | | | The description of the plan upgrade |
| pu_status | Status | | | The status of the plan upgrade. Options: 1 = Active, 0 = Inactive |
| pu_priority | Priority | | | The numeric priority of where the plan upgrade should appear in the list. Lower numbers will end up at the top of the list. Higher numbers will appear lower in the list. |
| pu_data | Extra Data | | | Any extra data for this upgrade. Must be an array of key/value pairs. Format: pu_data[custom]=text |
| invoice_hide | Client Visibility | | 0 | Whether to hide this upgrade on client invoices. Options: 0 = Visible to client, 2 = Hidden from client, 1 = Hidden when $ 0.00 |
| i18n | Internationalization | | | The internationalization config for this upgrade. Format: i18n[lang_id][pu_name]=Serveur Available Field: pu_name, pu_desc |

**Example Output**

```
{
  "data": "1",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.plan_upgrade_assign - Assign Plan Upgrade

This method is used to assign and unassign a plan upgrade to a service plan.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| pu_id | Plan Upgrade ID | Y | | |
| plan_id | Service Plan ID | Y | | |
| assign | Assign | Y | | Flag to either assign or unassign the plan upgrade to the plan. Options: 1 = Assign, 0 = Unassign |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.plan_upgrade_group_add - Add Plan Upgrade Group

This method is used to add a plan upgrade group.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| pug_name | Name | Y | | The name of the new upgrade group |
| pug_desc | Description | | | The description of the new upgrade group |
| pug_status | Status | | | The status of the upgrade group, 1 = Active, 0 = Inactive |
| pug_priority | Priority | | | The numeric priority of where the plan upgrade group should appear in the list. Lower numbers will end up at the top of the list. Higher numbers will appear lower in the list. |
| pug_data | Extra Data | | | Any extra data for this upgrade group. Must be an array of key/value pairs.<br>Format: pug_data[custom]=text |
| i18n | Internationalization | | | The internationalization config for this upgrade group.<br>Format: i18n[lang_id][pug_name]=Serveur<br>Available Field: pug_name, pug_desc |

**Example Output**

```
{
  "data": "1",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.plan_upgrade_group_assign - Assign Plan Upgrade Group

This method is used to assign and unassign a plan upgrade group to a service plan.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| pug_id | Plan Upgrade Group ID | Y | | |
| plan_id | Service Plan ID | Y | | |
| assign | Assign | Y | | Flag to either assign or unassign the plan upgrade group to the plan.<br>Options: 1 = Assign, 0 = Unassign |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.plan_upgrade_group_list - List Plan Upgrade Groups

This method is used to list plan upgrade groups.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| pug_id | Plan Upgrade Group ID | | | The numeric plan upgrade group ID, use 0 for Ungrouped |
| plan_id | Service Plan ID | | | The numeric service plan ID to fetch the upgrade groups for |
| lang_id | Language ID | | | The numeric language ID to fetch the upgrade groups in |
| include_inactive | Include Inactive Items | | 0 | Whether to include inactive upgrade groups (and inactive upgrades, option groups, and options, if all_options is specified). Options: 1 = Yes, 0 = No |
| exclude_empty | Exclude Empty Upgrade Groups | | 0 | Whether to exclude empty upgrade groups that do not have any options associated with them. Options: 1 = Yes, 0 = No |
| all_options | Include Options | | 0 | Whether to include upgrades, option groups, and options in the list. Options: 1 = Yes, 0 = No |

**Example Output**

```
{
  "data":{
    "1":{
      "pug_id": "1",
      "pug_status": "1",
      "pug_priority": "1",
      "pug_data": {
        "custom": "text",
        "internal_key": "hardware"
      },
      "pug_name": "Hardware",
      "pug_desc": "Choose the best hardware for your website"
      "upgrades": {
        "3": {
          "pu_id": "3",
          "pu_status": "1",
          "pu_priority": "0",
          "inv_hide": "0",
          "variable": "processor",
          "pu_data": [],
          "pu_name": "Processor",
          "pu_desc": "",
          "pug_id": "1",
          "pug_name": "Hardware",
          "pug_status": "1",
          "options": {
            "41": {
              "po_id": "41",
              "po_price": "0.00",
              "po_setup": "0.00",
              "po_cost": "0.00",
              "po_setup_cost" "0.00",
              "po_priority":"0",
              "po_status":"1",
              "po_val":"x3220",
              "po_data":{
                "processor_speed": "2400",
                "processing_cores": "4",
                "custom": "text"
              },
              "po_description": "Intel Xeon X3220",
              "po_desc": "",
              "pog_id": "0",
              "pog_name": "Ungrouped",
              "pog_status": "1",
              "pricing": {
                "0": {
                  "period": "0",
                  "api_label": "one_time",
```

```
                "price": "0.00",
                "setup": "0.00",
                "cost": "0.00",
                "setup_cost": "0.00"
              },
              "1": {
                "period": "1",
                "api_label": "monthly",
                "price": "0.00",
                "setup": "0.00",
                "cost": "0.00",
                "setup_cost": "0.00"
              },
              "3": {
                "period": "3",
                "api_label": "quarterly",
                "price": "0.00",
                "setup": "0.00",
                "cost": "0.00",
                "setup_cost": "0.00"
              }
            }
          }
        },
        "option_groups": {
          "1": {
            "pog_id": "1",
            "pog_status": "1",
            "pog_name": "Ungrouped",
            "pog_data": [],
            "pu_id": "23",
            "pu_name": "Processor",
            "pu_status": "1",
            "pug_id": "4",
            "pug_name": "Hardware",
            "pug_status": "1"
          }
        }
      }
    }
  },
  "0":{
    "pug_id": "0",
    "pug_status": "1",
    "pug_name": "Ungrouped",
    "pug_data": []
  }
},
"error_message": "",
"error_code": null,
"status": true
}
```

## uber.plan_upgrade_group_update - Update Plan Upgrade Group

This method is used to update a plan upgrade group.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| pug_id | Plan Upgrade Group ID | Y | | |
| pug_name | Name | | | The name of the upgrade group |
| pug_desc | Description | | | The description of the upgrade group |
| pug_status | Status | | | Status of the upgrade group. Options: 1 = Active, 0 = Inactive |
| pug_priority | Priority | | | The numeric priority of where the plan upgrade group should appear in the list. Lower numbers will end up at the top of the list. Higher numbers will appear lower in the list. |
| pug_data | Extra Data | | | Any extra data for this upgrade group. Must be an array of key/value pairs. Format: pug_data[custom]=text |
| i18n | Internationalization | | | The internationalization config for this upgrade group. Format: i18n[lang_id][pug_name]=Serveur Available Field: pug_name, pug_desc |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.plan_upgrade_list - List Plan Upgrades

This method is used to list plan upgrades.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| plan_id | Service Plan ID | | | The numeric service plan id to fetch the upgrades for |
| lang_id | Language ID | | | The numeric language id to fetch the upgrades in |
| pug_id | Plan Upgrade Group ID | | | The numeric upgrade group id to fetch the upgrades for |
| pu_id | Plan Upgrade ID | | | The numeric plan upgrade id to fetch |
| include_inactive | Include Inactive Items | | 0 | Whether to include inactive upgrades (and inactive option groups, and options, if all_options is specified). Options: 1 = Yes, 0 = No |
| exclude_empty | Exclude Empty Upgrades | | 0 | Whether to exclude empty upgrades that don't have any options associated with them. Options: 1 = Yes, 0 = No |
| all_options | Include Option Groups and Options | | 0 | Whether to include option groups, and options in the list. Options: 1 = Yes, 0 = No |

**Example Output**

```
{
  "data": {
    "30": {
      "pu_id": "30",
      "pug_id": "20",
      "pu_status": "1",
      "pu_priority": "2",
      "inv_hide": "0",
      "variable": "processor",
      "pu_data": [],
```

```
      "pu_name": "Processor",
      "pu_desc": "",
      "pug_name": "Hardware",
      "pug_status": "1",
      "options": {
       "41": {
         "po_id": "41",
         "po_price": "0.00",
         "po_setup_fee": "0.00",
         "po_cost": "0.00",
         "po_setup_cost" "0.00",
         "po_priority": "0",
         "po_status": "1",
         "po_val": "x3220",
         "po_data":{
           "processor_speed": "2400",
           "processing_cores": "4",
           "custom": "text"
         },
         "po_description": "Intel Xeon X3220",
         "po_desc": "",
         "pog_id": "0",
         "pog_name": "Ungrouped",
         "pog_status": "1",
         "pricing": {
            "0": {
              "period": "0",
              "api_label": "one_time",
              "price": "0.00",
              "setup_fee": "0.00",
              "cost": "0.00",
              "setup_cost": "0.00"
            },
            "1": {
              "period": "1",
              "api_label": "monthly",
              "price": "0.00",
              "setup_fee": "0.00",
              "cost": "0.00",
              "setup_cost": "0.00"
            },
            "3": {
              "period": "3",
              "api_label": "quarterly",
              "price": "0.00",
              "setup_fee": "0.00",
              "cost": "0.00",
              "setup_cost": "0.00"
            }
         }
       }
      },
      "option_groups": {
       "1": {
         "pog_id": "1",
         "pog_status": "1",
         "pog_name": "Ungrouped",
         "pog_data": [],
         "pu_id": "23",
         "pu_name": "Processor",
         "pu_status": "1",
         "pug_id": "4",
         "pug_name": "Hardware",
         "pug_status": "1"
       }
      }
    }
   },
   "error_message": "",
   "error_code": null,
   "status": true
}
```

## uber.plan_upgrade_update - Update Service Plan Upgrade

This method is used to update an existing upgrade on an existing service plan.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| pu_id | Plan Upgrade ID | Y | | The numeric plan upgrade id to update. |
| pug_id | Plan Upgrade Group ID | | | The numeric plan upgrade group id to add the upgrade to. Use 0 for Ungrouped |
| pu_name | Name | | | The name of the plan upgrade |
| variable | Variable | | | The API variable (machine name) of the plan upgrade |
| pu_desc | Description | | | The description of the plan upgrade |
| pu_status | Status | | | The status of the plan upgrade. Options: 1 = Active, 0 = Inactive |
| pu_priority | Priority | | | The numeric priority of where the plan upgrade should appear in the list. Lower numbers will end up at the top of the list. Higher numbers will appear lower in the list. |
| invoice_hide | Client Visibility | | 0 | Whether to hide this upgrade on client invoices. Options: 0 = Visible to client, 2 = Hidden from client, 1 = Hidden when $ 0.00 |
| pu_data | Extra Data | | | Any extra data for this upgrade. Must be an array of key/value pairs. Format: pu_data[custom]=text |
| i18n | Internationalization | | | The internationalization config for this upgrade. Format: i18n[lang_id][pu_name]=Serveur Available Field: pu_name, pu_desc |

**Example Output**

```
{
  "data":true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.quick_stats - Get Quick System Stats

This method is used to get a quick system overview, as seen in the top bar of the admin interface.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| when | Timestamp | | midnight | Start time for stats |

**Example Output**

```
{
  "data": {
    "messages": "0",
    "money": "0.00",
    "refund": "0.00",
    "client_new": "0",
    "gross": "0.00",
    "active_orders": "1",
    "new_orders": "0",
    "support_new": "49"
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.quick_stats_detail - Get Detailed System Stats

This method is used to provide more detailed information on various system stats.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| stat_name | Statistic to Return | Y | | Possible fields are: messages, support_new, client_new, money, refund, new_orders, active_orders |
| when | Timestamp | | midnight | Start time for stats |

**Example Output**

```
{
  "data": {
    "1": [
      "508",
      "1005",
      "ubersmith, inc.",
      "424",
      "220.54",
      "1312554328",
      "charge",
      "0.00"
    ],
    "2": [
      "509",
      "1005",
      "ubersmith, inc.",
      "424",
      "2.00",
      "1312554404",
      "charge",
      "0.00"
    ],
    "3": [
      "510",
      "1005",
      "ubersmith, inc.",
      "438",
      "85.50",
      "1312554667",
      "charge",
      "0.00"
    ],
    "4": [
      "511",
      "1005",
      "ubersmith, inc.",
      "438",
      "993.38",
      "1312554768",
      "check",
      "0.00"
    ]
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.rate_plan_add - Add a Rate Plan

This method is used to add a rate plan

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| name | Rate Plan Name | Y | | |
| description | Description | | | |
| status | Status | | 1 | |
| brand_id | Brand ID | | | |
| contract_terms | Contract Terms | | | An array of contract terms to be associated with the rate plan, the index of each element should be the contract term ID and the value should be 1 (e.g. contract_terms[8]=1 contract_terms[2]=1) |
| facilities | Facilities | | | An array of facilities to be associated with the rate plan, the index of each element should be the facility ID and the value should be 1 (e.g. facilities[2]=1 facilities[4]=1) |
| data | Data | | | An array of data values to be associated with the rate plan (e.g. data[fieldname]=fieldvalue) |

**Example Output**

```
{
  "data": "10",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.rate_plan_get - Get Rate Plan Details

This method is used to get the details of a specified rate plan

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| rate_plan_id | Rate Plan ID | Y | | |
| include_inactive | Include Inactive Items | | 0 | Flag to include inactive plans, contract terms, facilities and upgrades<br>Options:<br> 1 = Yes,<br> 0 = No |

**Example Output**

```
{
  "data": {
    "rate_plan_id": "1",
    "brand_id": "1",
    "name": "Rate Plan",
    "description": "Description",
    "status": "1",
    "contract_terms": {
      "1": {
        "contract_term_id": "1",
        "name": "Contract Name",
        "status": "1",
        "term": "12"
      },
      "2": {
        "contract_term_id": "2",
        "name": "Contract Name",
        "status": "1",
        "term": "12"
      }
    },
    "facilities": {
      "1": {
        "facility_id": "1",
        "name": "Facility Name 1",
```

```
            "code": "nyjfk",
            "status": "2"
        },
        "2": {
            "facility_id": "2",
            "name": "Facility Name",
            "code": nylga,
            "status": "1"
        }
    },
    "plans": {
        "1": {
            "plan_id": "1",
            "code": "mang0",
            "title": "Plan Title",
            "active": "1"
        },
        "2": {
            "plan_id": "2",
            "code": "ser0",
            "title": "Plan Title",
            "active": "0"
        }
    },
    "data": {
        "key1": "value1",
        "key2": "value2",
        "key2": "value3"
    },
    "upgrades": {
        "1": {
            "pu_id": "1",
            "pug_id": "1",
            "pu_status": "1",
            "inv_hide": "0",
            "variable": "processor",
            "pu_data": {
                "key1": "value1",
                "key2": "value2"
            },
            "pu_name": "Processor",
            "pu_desc": "Processor",
            "pug_name": "Hardware",
            "pug_status": "1",
            "pu_priority": "1",
            "options": {
                "1060": {
                    "po_id": "1060",
                    "pu_id": "1",
                    "pog_id": "1",
                    "po_val": "value",
                    "po_data": {
                        "key1": "value1",
                        "key2": "value2"
                    },
                    "po_priority": "1",
                    "po_description": "2.8 GHz",
                    "po_desc": "Upgrade Description",
                    "pog_name": "Intel",
                    "pu_name": "Processor",
                    "pug_name": "Hardware",
                    "pog_status": "1",
                    "pu_status": "1",
                    "inv_hide": "0",
                    "pug_id": "1",
                    "pug_status": "1",
                    "po_status": "1",
                    "rate_plan_option_id": "2"
                }
            },
            "option_groups": {
                "15": {
                    "pog_id": "15",
                    "pu_id": "12",
                    "pog_status": "1",
                    "pog_priority": "1",
                    "pog_data": {
                        "key1": "value1",
                        "key2": "value2"
```

```
      },
      "pog_name": "Intel",
      "pog_desc": "Description",
      "pu_name": "Processor",
      "pug_name": "Hardware",
      "pu_status": "1",
      "pug_id": "1",
      "pug_status": "1"
      }
    }
  }
},
  "upgrade_groups": {
    "1": {
      "pug_id": "1",
      "pug_status": "1",
      "pug_data": {
        "key1": "value1",
        "key2": "value2"
      }
      "pug_name": "Hardware",
      "pug_desc": ""
    }
  }
},
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.rate_plan_list - List Rate Plans

This method is used to list rate plans

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| brand_id | Brand ID | | | |
| status | Status | | | 1 = Active Rate Plans, 0 = Inactive Rate Plans |

**Example Output**

```
{
  "data": {
    "1": {
      "rate_plan_id": "1",
      "brand_id": "1",
      "name": "Rate Plan",
      "description": "Description",
      "status": "1"
    },
    "2": {
      "rate_plan_id": "1",
      "brand_id": "1",
      "name": "Rate Plan",
      "description": "Description",
      "status": "1"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.rate_plan_option_add - Add a Plan Option to a Rate Plan

This method is used to add an plan option to a rate plan

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| rate_plan_id | Rate Plan ID | Y | | |
| po_id | Plan Option ID | Y | | |
| status | Status | | 1 | Options:<br>  1 = Active<br>  0 = Inactive |
| data | Data | | | An array of data values to be associated with the rate plan option (e.g. data[fieldname]=fieldvalue) |
| pricing | Pricing | | | The pricing matrix.<br>Format:<br>  pricing[period_id][type_method]=modifier_id<br>  pricing[period_id][type]=value<br><br>Period IDs can be brand specific or global. The global options for "period_id" are:<br>  0 = One Time Fee<br>  1 = Monthly<br>  3 = Quarterly<br>  6 = Semi-Annually<br>  12 = Annually<br><br>Options for "type": price_method, price, setup_method, setup, cost_method, cost, setup_cost_method, setup_cost<br><br>Options for "modifier_id":<br>  0 = Inherit<br>  1 = Override<br>  2 = Modify By Amount<br>  3 = Modify By Percent<br><br>Example:<br>  pricing[0][price_method]=0<br>  pricing[0][price]=10.00<br>  pricing[3][price_method]=1<br>  pricing[3][price]=22.00 |

**Example Output**

```
{
  "data": "10",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.rate_plan_option_get - Get Rate Plan Option Details

This method is used to get the details of a specified rate plan option

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| rate_plan_option_id | Rate Plan Option ID | Y | | |
| include_option_info | Include Option Info Flag | | | Boolean flag to include the plan option information in the return |

**Example Output**

```
{
  "data": {
    "rate_plan_option_id": "1",
    "rate_plan_id": "1",
    "po_id": "1103",
    "data": {
      "key1": "value1",
      "key2": "value2",
      "key2": "value3"
    },
    "pricing": {
      "0": {
        "period": "0",
        "price_method": "3",
        "price": "-10.00",
        "setup_method": "3",
        "setup": "-10.00",
        "cost_method": "3",
        "cost": "-10.00",
        "setup_cost_method": "3",
        "setup_cost": "-10.00",
        "api_label": "one_time"
      },
      "3": {
        "period": "3",
        "price_method": "3",
        "price": "-10.00",
        "setup_method": "3",
        "setup": "-10.00",
        "cost_method": "3",
        "cost": "-10.00",
        "setup_cost_method": "3",
        "setup_cost": "-10.00",
        "api_label": "quarterly"
      }
    },
    "option_info": {
      "po_id": "1103",
      "pu_id": "1",
      "pog_id": "12",
      "po_val": "",
      "po_data": {
        "key1": "value1",
        "key2": "value2",
      },
      "po_priority": "1",
      "po_description": "Valencia",
      "po_desc": "",
      "pog_name": "AMD",
      "pu_name": "Processor",
      "pug_name": "Hardware",
      "pog_status": "1",
      "pu_status": "1",
      "inv_hide": "0",
      "pug_id": "1",
      "pug_status": "1",
      "po_status": "1",
      "i18n": {
        "1": {
          "id": "298",
          "po_id": "1103",
          "lang_id": "1",
          "po_description": "",
          "po_desc": "",
```

```
      }
     }
    }
   }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.rate_plan_option_update - Update a Plan Option in a Rate Plan

This method is used to update a plan option in a rate plan

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| rate_plan_option_id | Rate Plan Option ID | Y | | |
| status | Status | | 1 | Options:<br>  1 = Active<br>  0 = Inactive |
| data | Data | | | An array of data values to be associated with the rate plan option (e.g. data[fieldname]=fieldvalue) |
| delete_data | Delete Data | | | An array of data values to be deleted from the rate plan option, the index of each element should be the data name and the value should be 1 (e.g. delete_data[fieldname]=1) |
| pricing | Pricing | | | The pricing matrix.<br>Format:<br>  pricing[period_id][type_method]=modifier_id<br>  pricing[period_id][type]=value<br><br>Period IDs can be brand specific or global. The global options for "period_id" are:<br>  0 = One Time Fee<br>  1 = Monthly<br>  3 = Quarterly<br>  6 = Semi-Annually<br>  12 = Annually<br><br>Options for "type": price_method, price, setup_method, setup, cost_method, cost, setup_cost_method, setup_cost<br><br>Options for "modifier_id":<br>  0 = Inherit<br>  1 = Override<br>  2 = Modify By Amount<br>  3 = Modify By Percent<br><br>Example:<br>  pricing[0][price_method]=0<br>  pricing[0][price]=10.00<br>  pricing[3][price_method]=1<br>  pricing[3][price]=22.00 |

**Example Output**

```
{
  "data": {
    "rate_plan_option_id": "1",
    "rate_plan_id": "3",
    "po_id": "1001",
    "data": {
      "key1": "value1",
      "key2": "value2"
    },
    "pricing": {
      "0": {
```

```
      "period": "0",
      "price_method": "3",
      "price": "-10.00",
      "setup_method": "3",
      "setup": "-10.00",
      "cost_method": "3",
      "cost": "-10.00",
      "setup_cost_method": "3",
      "setup_cost": "-10.00",
      "api_label": "one_time"
    },
    "3": {
      "period": "3",
      "price_method": "3",
      "price": "-10.00",
      "setup_method": "3",
      "setup": "-10.00",
      "cost_method": "3",
      "cost": "-10.00",
      "setup_cost_method": "3",
      "setup_cost": "-10.00",
      "api_label": "quarterly"
    }
  }
},
"error_message": "",
"error_code": null,
"status": true
}
```

## uber.rate_plan_service_plan_add - Add a Service Plan to a Rate Plan

This method is used to add a service plan to a rate plan

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| rate_plan_id | Rate Plan ID | Y | | |
| plan_id | Plan ID | Y | | |
| status | Status | | 1 | |
| pricing | Pricing | | | The pricing matrix.<br>Format:<br>  pricing[period_id][type_method]=modifier_id<br>  pricing[period_id][type]=value<br><br>Period IDs can be brand specific or global. The global options for "period_id" are:<br>  0 = One Time Fee<br>  1 = Monthly<br>  3 = Quarterly<br>  6 = Semi-Annually<br>  12 = Annually<br><br>Options for "type": price_method, price, setup_method, setup, cost_method, cost, setup_cost_method, setup_cost<br><br>Options for "modifier_id":<br>  0 = Inherit<br>  1 = Override<br>  2 = Modify By Amount<br>  3 = Modify By Percent<br><br>Example:<br>  pricing[0][price_method]=0<br>  pricing[0][price]=10.00<br>  pricing[3][price_method]=1<br>  pricing[3][price]=22.00 |

**Example Output**

```
{
  "data": {
    "plan_id": "1",
    "code": "mang0",
    "title": "Plan Title",
    "status": "1",
    "pricing": {
      "0": {
        "period": "0",
        "price_method": "3",
        "price": "-10.00",
        "setup_method": "3",
        "setup": "-10.00",
        "cost_method": "3",
        "cost": "-10.00",
        "setup_cost_method": "3",
        "setup_cost": "-10.00",
        "api_label": "one_time"
      },
      "3": {
        "period": "3",
        "price_method": "3",
        "price": "-10.00",
        "setup_method": "3",
        "setup": "-10.00",
        "cost_method": "3",
        "cost": "-10.00",
        "setup_cost_method": "3",
        "setup_cost": "-10.00",
        "api_label": "quarterly"
      }
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.rate_plan_service_plan_get - Get a Service Plan Detail in a Rate Plan

This method is used to get a service plan detail in a rate plan

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| rate_plan_id | Rate Plan ID | Y | | |
| plan_id | Plan ID | Y | | |

**Example Output**

```
{
  "data": {
    "plan_id": "1",
    "code": "mang0",
    "title": "Plan Title",
    "status": "1",
    "pricing": {
      "0": {
        "period": "0",
        "price_method": "3",
        "price": "-10.00",
        "setup_method": "3",
        "setup": "-10.00",
        "cost_method": "3",
        "cost": "-10.00",
        "setup_cost_method": "3",
        "setup_cost": "-10.00",
        "api_label": "one_time"
      },
      "3": {
        "period": "3",
        "price_method": "3",
        "price": "-10.00",
        "setup_method": "3",
        "setup": "-10.00",
        "cost_method": "3",
        "cost": "-10.00",
        "setup_cost_method": "3",
```

```
        "setup_cost": "-10.00",
        "api_label": "quarterly"
      }
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.rate_plan_service_plan_update - Update a Service Plan in a Rate Plan

This method is used to update a service plan in a rate plan

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| rate_plan_id | Rate Plan ID | Y | | |
| plan_id | Plan ID | Y | | |
| status | Status | | 1 | |
| pricing | Pricing | | | The pricing matrix.<br>Format:<br>  pricing[period_id][type_method]=modifier_id<br>  pricing[period_id][type]=value<br><br>Period IDs can be brand specific or global. The global options for "period_id" are:<br>  0 = One Time Fee<br>  1 = Monthly<br>  3 = Quarterly<br>  6 = Semi-Annually<br>  12 = Annually<br><br>Options for "type": price_method, price, setup_method, setup, cost_method, cost, setup_cost_method, setup_cost<br><br>Options for "modifier_id":<br>  0 = Inherit<br>  1 = Override<br>  2 = Modify By Amount<br>  3 = Modify By Percent<br><br>Example:<br>  pricing[0][price_method]=0<br>  pricing[0][price]=10.00<br>  pricing[3][price_method]=1<br>  pricing[3][price]=22.00 |

**Example Output**

```
{
  "data": {
    "plan_id": "1",
    "code": "mang0",
    "title": "Plan Title",
    "status": "1",
    "pricing": {
      "0": {
        "rate_plan_id": "1",
        "plan_id": "3",
        "period": "0",
        "price_method": "3",
        "price": "-10.00",
        "setup_method": "3",
        "setup": "-10.00",
        "cost_method": "3",
        "cost": "-10.00".
        "setup_cost_method": "3",
        "setup_cost": "-10.00"
```

```
      },
      "3": {
        "rate_plan_id": "1",
        "plan_id": "3",
        "period": "3",
        "price_method": "3",
        "price": "-10.00",
        "setup_method": "3",
        "setup": "-10.00",
        "cost_method": "3",
        "cost": "-10.00".
        "setup_cost_method": "3",
        "setup_cost": "-10.00"
      }
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.rate_plan_update - Update a Rate Plan

This method is used to update a rate plan

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| rate_plan_id | Rate Plan ID | Y | | |
| name | Rate Plan Name | | | |
| description | Description | | | |
| status | Status | | | |
| brand_id | Brand ID | | | |
| contract_terms | Contract Terms | | | An array of contract terms to be associated with the rate plan, the index of each element should be the contract term ID and the value should be 1 (e.g. contract_terms[8]=1 contract_terms[2]=1) |
| facilities | Facilities | | | An array of facilities to be associated with the rate plan, the index of each element should be the facility ID and the value should be 1 (e.g. facilities[2]=1 facilities[4]=1) |
| data | Data | | | An array of data values to be associated with the rate plan (e.g. data[fieldname]=fieldvalue) |
| delete_contract_terms | Delete Contract Terms | | | An array of contract terms to be deleted from the rate plan, the index of each element should be the contract term ID and the value should be 1 (e.g. delete_contract_terms[8]=1 delete_contract_terms[2]=1) |
| delete_facilities | Delete Facilities | | | An array of facilities to be deleted from the rate plan, the index of each element should be the facility ID and the value should be 1 (e.g. delete_facilities[2]=1 delete_facilities[4]=1) |
| delete_data | Delete Data | | | An array of data values to be deleted from the rate plan, the index of each element should be the data name and the value should be 1 (e.g. delete_data[fieldname]=1) |

**Example Output**

```
{
  "data": {
    "rate_plan_id": "1",
    "brand_id": "1",
    "name": "Rate Plan",
    "description": "Description",
    "status": "1",
    "contract_terms": {
      "1": {
        "contract_term_id": "1",
        "name": "Contract Name",
        "status": "1",
        "term": "12"
```

```
    },
    "2": {
      "contract_term_id": "2",
      "name": "Contract Name",
      "status": "1",
      "term": "12"
    }
  },
  "facilities": {
    "1": {
      "facility_id": "1",
      "name": "Facility Name 1",
      "code": "nyjfk",
      "status": "2"
    },
    "2": {
      "facility_id": "2",
      "name": "Facility Name",
      "code": "nylga",
      "status": "1"
    }
  },
  "plans": {
    "1": {
      "plan_id": "1",
      "code": "mang0",
      "title": "Plan Title",
      "status": "1"
    },
    "2": {
      "plan_id": "2",
      "code": "ser0",
      "title": "Plan Title",
      "status": "0"
    }
  },
},
"error_message": "",
"error_code": null,
"status": true
}
```

## uber.service_module_list - List Service Modules

This method is used to list all available service modules.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| module | Service Module | | | The name of a service module to filter the list on |

**Example Output**

```
{
  "data":{
    "bw_billing":{
      "title":"Bandwidth Billing",
      "api_label":"bw_billing",
      "config":{
        "precision":{
          "api_label":"precision",
          "label":"Overage Precision:",
          "default":"0",
          "type":"select"
        },
        "showchart":{
          "api_label":"showchart",
          "label":"Show Usage Chart in PDF invoice",
          "default":"0",
          "type":"select"
        },
        "showgraph":{
          "api_label":"showgraph",
          "label":"Show Usage Graph in PDF invoice",
          "default":"0",
          "type":"select"
        },
        "no_devices_error":{
          "api_label":"no_devices_error",
```

```
            "label":"Error when no devices are associated",
            "default":"1",
            "type":"select"
          }
        }
      },
    }
    "error_message": "",
    "error_code": null,
    "status": true
}
```

## uber.service_plan_activate - Activate Service Plan

This method is used to activate a service plan.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| plan_id | Service Plan ID | Y | | The numeric service plan ID to activate |

**Example Output**

```
{
    "data":true,
    "error_message": "",
    "error_code": null,
    "status": true
}
```

## uber.service_plan_add - Add Service Plan

This method is used to add a new service plan.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| class_id | Brand ID | | | The brand to associate the service plan with |
| title | Service Plan Title | Y | | The title of the service plan |
| code | Service Plan Code | Y | | The code of the service plan. Must be unique per brand |
| category | Category | | | The service plan category |
| api_include | Include in API | | 1 | Whether to include in the API output. Options: 1 = Yes, 0 = No |
| allow_quotes | Allow Quoting | | 0 | Whether to allow admins to create quotes of this service plan for clients. Options: 1 = Yes, 0 = No |
| qbaccount | Quickbooks Account | | | The QuickBooks account to link this service plan with |
| period | Default Billing Period | | | An integer representing the default billing period. Options: one_time = 0 monthly = 1 quarterly = 3 semi = 6 annual = 12 |
| setup_qs | Setup Quantity Sensitive | | 1 | Whether the setup fee is a multiplied by the quantity. Options: 1 = Yes, 0 = No |
| bill_type | Default Billing Type | | 1 | Whether to bill by month or by period. Options: 1 = By Month, 0 = By Period |
| auto_charge | Automatically Charge | | 0 | Whether to automatically charge clients for this service. Options: 1 = Yes, 0 = No |
| auto_prorate | Automatically Prorate | | 0 | Whether to automatically prorate to the service. Options: 1 = Yes, 0 = No |
| post_renew | Post Renew | | 0 | Whether to bill in arrears for this service. Options: 1 = Yes, 0 = No |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| bill_prior | Bill in Advance | | 0 | Whether to bill the service up to 31 days earlier based on the client's billing configuration<br>Options: 1 = Yes, 0 = No |
| default_quantity | Default Quantity | | 1 | The default quantity when creating service packs |
| auto_suspend | Auto Suspend | | | The number of days to wait before suspending the service for non-payment. Leave blank or 0 for disabled |
| auto_cancel | Auto Cancel | | | The number of days to wait before canceling the service for non-payment. Leave blank or 0 for disabled |
| hide_upgrades | Hide Upgrades | | 0 | Whether to hide upgrades from clients on their invoice and within Ubersmith.<br>Options: 1 = Yes, 0 = No |
| zero_invoice | Invoice when Zero | | 0 | Whether the service should be included on invoices when the cost is zero.<br>Options: 1 = Yes, 0 = No |
| welcome_send | Send Welcome Letter | | 0 | Whether to send a welcome letter to client on service pack creation.<br>Options: 1 = Yes, 0 = No |
| welcome_email_from | Welcome Letter From | | | The email address that the welcome letter should come from |
| welcome_subject | Welcome Letter Subject | | | The subject of the welcome letter email |
| welcome_body | Welcome Letter Body | | | The body of the welcome letter email. Should be plain text, and allows for variable substitutions such as ##first_name##. Check Setup & Admin for the full list of valid variable substitutions. |
| tax_situs_code | Tax Situs Rule Code | | | The Tax Situs rule code to use. Check the Tax Engine Options under Setup & Admin > Service Plans > {plan} to see a full list of rule codes. |
| tax_trans_type_code | Tax Transaction Type Code | | | The Tax Transaction type code to use. Check the Tax Engine Options under Setup & Admin > Service Plans > {plan} to see a full list of type codes. |
| plan_data | Data | | | An array of data values to be associated with the service plan.<br>Format: plan_data[fieldname]=fieldvalue |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| pricing | Pricing | | | The pricing matrix.<br>Format:<br>  pricing[period_id][type_method]=modifier_id<br>  pricing[period_id][type]=value<br><br>Period IDs can be brand specific or global. The global options for "period_id" are:<br>  0 = One Time Fee<br>  1 = Monthly<br>  3 = Quarterly<br>  6 = Semi-Annually<br>  12 = Annually<br><br>Options for "type": price_method, price, setup_method, setup, cost_method, cost, setup_cost_method, setup_cost<br><br>Options for "modifier_id":<br>  0 = Inherit<br>  1 = Override<br>  2 = Modify By Amount<br>  3 = Modify By Percent<br><br>Example:<br>  pricing[0][price_method]=0<br>  pricing[0][price]=10.00<br>  pricing[3][price_method]=1<br>  pricing[3][price]=22.00 |
| taxes | Taxes | | | The tax rates to use for this service plan. Usable tax_id can be found in the Manage Tax Rates table. The value sets whether to apply the given tax rate to the fee type.<br>Format: taxes[service][{tax_id}]=1, taxes[setup][{tax_id}]=0<br>Available Type: service, setup |
| i18n | Internationalization | | | The internationalized config for this service plan. Substitute lang_id with the numeric language id.<br>Format: i18n[lang_id][title]=Colo<br>Available Field: title, category, welcome_subject, welcome_body |

**Example Output**

```
{
  "data": "10",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.service_plan_copy - Copy Service Plan

This method is used to copy a service plan to a new service plan.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| plan_id | Service Plan ID | Y | | The numeric service plan id to copy |
| title | Service Plan Title | Y | | The title of the new, copied service plan |
| code | Service Plan Code | Y | | The code of the new, copied service plan. Must be unique per brand. |
| class_id | Brand ID | | | The brand to associate the new, copied service plan with |
| i18n | Internationalization | | | The internationalized config for the new, copied service plan. Substitute lang_id with the numeric language id. Format: i18n[lang_id][title]=Colo Available Field: title, category, welcome_subject, welcome_body |

**Example Output**

```
{
  "data": "123",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.service_plan_deactivate - Deactivate Service Plan

This method is used to deactivate a service plan.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| plan_id | Service Plan ID | Y | | The numeric service plan ID to deactivate |

**Example Output**

```
{
  "data":true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.service_plan_get - Get Service Plan Details

This method is used to get the details of a specified service plan.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| plan_id | Service Plan ID | Y | | |

**Example Output**

```
{
  "data": {
    "welcomeadd": "",
    "bill_type": "1",
    "price": "200.00",
    "upgrades": [

    ],
    "qblistid": "8000000D-1304980286",
    "cancel_after": "0",
    "category": "",
    "suspend_bool": "0",
    "quantity": "1.00",
    "code": "TST3",
    "title": "Test 3",
    "notes": [

    ],
    "auto_bill": "0",
```

 ⸬⸬ ubersmith                                                    **API Documentation**

```
"api_include": "1",
"period": "1",
"metadata": {
  "11": {
    "required": "0",
    "size": "0",
    "prefix": "Included Transfer Unit",
    "metagroup_id": "4",
    "editable": "0",
    "applicable": "1",
    "client_access": "0",
    "priority": "3",
    "class_id": "1",
    "default_val": "",
    "show_in_list": "0",
    "type": "select",
    "value": "mb",
    "rows": "0",
    "cols": "0",
    "suffix": "",
    "metaconf_id": "11",
    "meta_type": "pack",
    "options": "\"\",\"mb:MB\",\"gb:GB\",\"tb:TB\",\"mbps:Mbps\",\"gbps:Gbps\"",
    "variable": "bw_included_transfer_unit",
    "plan_id": "11"
  }
},
"plan_data" : {
  "data_key1": "data_value1",
  "data_key2": "data_value2",
  "data_key3": "data_value3",
},
"tax_engine_id": "3",
"tax_trans_type_code": "010101",
"tax_situs_code": "07",
"tax_trans_type_code_setup": "020202",
"tax_situs_code_setup": "08",
"zero_inv": "0",
"class_id": "1",
"taxes": [
],
"upgrade_groups": [
  {
    "spug_plan_id": "11",
    "spug_id": 0,
    "spug_status": 1,
    "spug_name": "Ungrouped"
  }
],
"lang_id": "1",
"id": "15",
"post_renew": "0",
"bill_prior": "0",
"suspend_after": "0",
"setup_qs": "0",
"cancel_bool": "0",
"setup": "0.00",
"pricing": {
  "0": {
    "price": "100.00",
    "period": "0",
    "setup_fee": "0.00",
    "api_label": "one_time"
  },
  "1": {
    "price": "200.00",
    "period": "1",
    "setup_fee": "0.00",
    "api_label": "monthly"
  },
  "3": {
    "price": "0.00",
    "period": "3",
    "setup_fee": "0.00",
    "api_label": "quarterly"
  }
},
"i18n": {
  "1": {
```

```
          "category": "",
          "title": "Test 3",
          "lang_id": "1",
          "id": "249",
          "welcomebody": "",
          "welcomesub": "",
          "plan_id": "11"
        }
      },
      "offerwelcome": "0",
      "welcomebody": "",
      "service_host": "",
      "qbaccount": "Sales - Software",
      "welcomesub": "",
      "active": "1",
      "plan_id": "11",
      "rate_plans": {
        "1": {
          "rate_plan_id": "1",
          "brand_id": "1",
          "name": "Rate Plan",
          "description": "Description",
          "status": "1"
        },
        "2": {
          "rate_plan_id": "1",
          "brand_id": "1",
          "name": "Rate Plan",
          "description": "Description",
          "status": "1"
        }
      }
    },
    "error_message": "",
    "error_code": null,
    "status": true
}
```

## uber.service_plan_list - List Service Plans

This method is used to get a list of service plans.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| code | Service Plan Code | | | |
| brand_id | Brand ID | | | |
| active | Active | | 1 | 1 = Active Plans, 0 = Deactivated Plans, 2 = All Plans |
| qblistid | Quick Book List ID | | | |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "33": {
      "welcomeadd": "support@ubersmith.com",
      "bill_type": "0",
      "price": "0.00",
      "qblistid": "80000015-1304980295",
      "cancel_after": "0",
      "category": "virt",
      "suspend_bool": "0",
      "quantity": "1.00",
      "code": "cpanel",
      "title": "cPanel Account",
      "auto_bill": "0",
      "api_include": "1",
      "period": "1",
```

```
          "tax_engine_id": "3",
          "tax_trans_type_code": "010101",
          "tax_situs_code": "07",
          "tax_trans_type_code_setup": "020202",
          "tax_situs_code_setup": "08",
          "zero_inv": "0",
          "class_id": "1",
          "post_renew": "0",
          "bill_prior": "0",
          "suspend_after": "0",
          "setup_qs": "0",
          "cancel_bool": "0",
          "setup_fee": "0.00",
          "setup": "0.00",
          "offerwelcome": "1",
          "welcomebody": "This email is being sent to welcome you as a new client.",
          "service_host": "",
          "qbaccount": "Sales - Software",
          "welcomesub": "Welcome",
          "active": "1",
          "plan_id": "33"
      }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.service_plan_metadata_list - List Service Plan Default Metadata

This method is used to list an existing service plan's default metadata.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| plan_id | Service Plan ID | Y | | The numeric service plan id to update the default metadata values for |

**Example Output**

```
{
  "data":{
    "123":{
      "metaconf_id":"123",
      "plan_id":"234",
      "value":"The default metadata value for this plan",
      "variable":"variable_name",
      "meta_type":"pack",
      "class_id":"1",
      "metagroup_id":"1",
      "prefix":"The Variable",
      "suffix":"",
      "type":"textarea",
      "default_val":"The global service default metadata value",
      "options":"",
      "size":"0",
      "cols":"50",
      "rows":"5",
      "priority":"5",
      "client_access":"0",
      "regular_expression_match":"",
      "regular_expression_help":"",
      "unique":"0",
      "required":"0",
      "show_in_list":"0",
      "editable":"0"
    },
  }
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.service_plan_metadata_update - Update Service Plan Metadata

This method is used to update an existing service plan's default metadata.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| plan_id | Service Plan ID | Y | | The numeric service plan id to update the default metadata values for |
| metadata | Metadata | Y | | An array of metadata defaults to update.<br>Format: metadata[api_variable]=default |

**Example Output**

```
{
  "data":true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.service_plan_module_add - Add Service Plan Service Module

This method is used to add a service module to an existing service plan.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| plan_id | Service Plan ID | Y | | The numeric service plan id to add the service module to |
| module | Module | Y | | The machine name of the service module. Should be one of the service modules available from the method uber.service_module_list |
| enabled | Enabled | Y | 1 | Whether the service module should be enabled or not.<br>Options: 1 = Yes, 0 = No |
| client_access | Client Access | | 0 | The client access level for this service module.<br>Options: 0 = No Access, 1 = View, 2 = Edit |
| config | Config | | | An array of config items for the service module. Check uber.service_module_list for a full list of config options.<br>Format: config[name]=value |

**Example Output**

```
{
  "data":"123",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.service_plan_module_delete - Delete Service Plan Service Module

This method is used to delete a service module from an existing service plan.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| plan_id | Service Plan ID | Y | | The numeric service plan id to add the service module to |
| service_module_id | Service Module ID | Y | | The numeric service plan service_module_id to delete, available from the uber.service_plan_module_list method |

**Example Output**

```
{
  "data":true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.service_plan_module_list - List Service Plan Service Modules

This method is used to list an existing service plan's service modules.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| plan_id | Service Plan ID | Y | | The numeric service plan id to update the default metadata values for |

**Example Output**

```
{
  "data":{
    "8":{
      "plan_id":"280",
      "enabled":"1",
      "module":"bw_notification",
      "client_access":"0",
      "config":{
        "approaching_threshold":"5",
        "overage_from":"noc@yourdomain.com",
        "overage_subject":"Bandwidth Overage Notice",
        "overage_body":"Included Bandwidth Usage Exceeded\n\nThis is an automated message to let you know that you have
exceeded the included bandwidth usage for the current period.  Bandwidth usage is now being charged at the rate
below.\n\nCurrent Usage:    ##current_usage##\nIncluded Transfer: ##included_transfer##\nOverage Rate:
$##overage_rate##\/##overage_unit##\nDevice:        ##device_info##\n",
        "approaching_subject":"Approaching Bandwidth Overage",
        "approaching_body":"Approaching Included Bandwidth Usage\n\nThis is an automated message to let you know that you are
approaching the included bandwidth usage for the current period.  Excess bandwidth usage will be charged at the rate
below.\n\nCurrent Usage:    ##current_usage##\nIncluded Transfer: ##included_transfer##\nOverage Rate:
$##overage_rate##\/##overage_unit##\nDevice:        ##device_info##\n"
      },
      "service_module_id":"8"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.service_plan_module_update - Update Service Plan Service Module

This method is used to update a service module on an existing service plan.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| plan_id | Service Plan ID | Y | | The numeric service plan ID |
| service_module_id | Service Module ID | Y | | The numeric service plan module ID to update, available from the uber.service_plan_module_list method |
| module | Module | | | The machine name of the service module. Should be one of the service modules available from the method uber.service_module_list |
| enabled | Enabled | | 1 | Whether the service module should be enabled or not. Options: 1 = Yes, 0 = No |
| client_access | Client Access | | 0 | The client access level for this service module. Options: 0 = No Access, 1 = View, 2 = Edit |
| config | Config | | | An array of config items for the service module. Check uber.service_module_list for a full list of config options. Format: config[name]=value |

**Example Output**

```
{
  "data":true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.service_plan_note_add - Add a New Service Plan Note

This method is used to add a service plan note.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| plan_id | Service Plan ID | Y | | |
| note | Note Text | Y | | The note text. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.service_plan_note_delete - Delete a Service Plan Note

This method is used to delete a service plan note.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| note_id | Note ID | Y | | The note to delete. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.service_plan_note_list - List Service Plan Notes

This method is used to list a service plan's notes.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| plan_id | Service Plan ID | Y | | The numeric service plan ID from which to fetch the service plan notes |

**Example Output**

```
{
  "data":{
    "1":{
      "spn_id": "1",
      "note": "A note about this service plan",
      "priority": "1"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.service_plan_note_update - Update a Service Plan Note

This method is used to update a service plan note.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| note_id | Note ID | Y | | The note to update. |
| note | Note Text | Y | | The note text. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.service_plan_update - Edit Service Plan

This method is used to update an existing service plan.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| plan_id | Service Plan ID | Y | | The service plan to update |
| title | Service Plan Title | | | The title of the service plan. This will override i18n[lang_id][title], where lang_id is the default language ID. |
| code | Service Plan Code | | | The code of the service plan. Must be unique per brand. |
| category | Category | | | The service plan category. This will override i18n[lang_id][category], where lang_id is the default language ID. |
| api_include | Include in API | | 0 | Whether to include in the API output. Options: 1 = Yes, 0 = No |
| allow_quotes | Allow Quoting | | 0 | Whether to allow admins to create quotes of this service plan for clients. Options: 1 = Yes, 0 = No |
| qbaccount | Quickbooks Account | | | The QuickBooks account to link this service plan with |
| period | Default Billing Period | | | An integer representing the default billing period. Options: one_time = 0 monthly = 1 quarterly = 3 semi = 6 annual = 12 |
| setup_qs | Setup Quantity Sensitive | | 1 | Whether the setup fee is a multiplied by the quantity. Options: 1 = Yes, 0 = No |
| bill_type | Default Billing Type | | 1 | Whether to bill by month or by period. Options: 1 = By Month, 0 = By Period |
| auto_charge | Automatically Charge | | 0 | Whether to automatically charge clients for this service. Options: 1 = Yes, 0 = No |
| auto_prorate | Automatically Prorate | | 0 | Whether to automatically prorate to the service. Options: 1 = Yes, 0 = No |
| post_renew | Post Renew | | 0 | Whether to bill in arrears for this service. Options: 1 = Yes, 0 = No |
| bill_prior | Bill in Advance | | 0 | Whether to bill the service up to 31 days earlier based on the client's billing configuration. Options: 1 = Yes, 0 = No |
| default_quantity | Default Quantity | | 1 | The default quantity when creating service packs |
| auto_suspend | Auto Suspend | | | The number of days to wait before suspending the service for non-payment. Leave blank or 0 for disabled |
| auto_cancel | Auto Cancel | | | The number of days to wait before canceling the service for non-payment. Leave blank or 0 for disabled |
| hide_upgrades | Hide Upgrades | | 0 | Whether to hide upgrades from clients on their invoice and within Ubersmith. Options: 1 = Yes, 0 = No |
| zero_invoice | Invoice when Zero | | 0 | Whether the service should be included on invoices when the cost is zero. Options: 1 = Yes, 0 = No |
| welcome_send | Send Welcome Letter | | 0 | Whether to send a welcome letter to client on service pack creation. Options: 1 = Yes, 0 = No |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| welcome_email_from | Welcome Letter From | | | The email address that the welcome letter should come from |
| welcome_subject | Welcome Letter Subject | | | The subject of the welcome letter email |
| welcome_body | Welcome Letter Body | | | The body of the welcome letter email. Should be plain text, and allows for variable substitutions such as ##first_name##. Check Setup & Admin for the full list of valid variable substitutions. |
| tax_situs_code | Tax Situs Rule Code | | | The Tax Situs rule code to use. Check the Tax Engine Options under Setup & Admin > Service Plans > {plan} to see a full list of rule codes. |
| tax_trans_type_code | Tax Transaction Type Code | | | The Tax Transaction type code to use. Check the Tax Engine Options under Setup & Admin > Service Plans > {plan} to see a full list of type codes. |
| pricing | Pricing | | | The pricing matrix.<br>Format:<br>  pricing[period_id][type_method]=modifier_id<br>  pricing[period_id][type]=value<br><br>Period IDs can be brand specific or global. The global options for "period_id" are:<br>  0 = One Time Fee<br>  1 = Monthly<br>  3 = Quarterly<br>  6 = Semi-Annually<br>  12 = Annually<br><br>Options for "type": price_method, price, setup_method, setup, cost_method, cost, setup_cost_method, setup_cost<br><br>Options for "modifier_id":<br>  0 = Inherit<br>  1 = Override<br>  2 = Modify By Amount<br>  3 = Modify By Percent<br><br>Example:<br>  pricing[0][price_method]=0<br>  pricing[0][price]=10.00<br>  pricing[3][price_method]=1<br>  pricing[3][price]=22.00 |
| taxes | Taxes | | | The tax rates to use for this service plan. Usable tax_id can be found in the Manage Tax Rates table. The value sets whether to apply the given tax rate to the fee type.<br>Format: taxes[service][{tax_id}]=1, taxes[setup][{tax_id}]=0<br>Available Type: service, setup |
| i18n | Internationalization | | | The internationalized config for this service plan. Substitute lang_id with the numeric language id.<br>Format: i18n[lang_id][title]=Colo<br>Available Field: title, category, welcome_subject, welcome_body |

**Example Output**

```
{
  "data":true,
  "error_message": "",
  "error_code": null,
  "status": true
```

```
}
```

## uber.tax_engine_list - List Tax Engines

This method is used to list the configured tax engines.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| type | Type | | | Tax engine type |
| description | Description | | | Tax engine description |
| status | Status | | 1 | 0 = Deactivated<br>1 = Active, 2 = All |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "1": {
      "tax_engine_id": "1",
      "status": "1",
      "type": "suretax",
      "description": "SureTax Testing",
      "username": "testing",
      "server": "https://testapi.taxrating.net/Services/V01/SureTax.asmx/PostRequest"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.tax_exemption_type_get - Get a Tax Exemption Type

This method is used to get a tax exemption type.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| tax_exemption_type_id | Tax Exemption Type ID | Y | | Tax Exemption Type ID |

**Example Output**

```
{
  "data": {
    "tax_exemption_type_id": "16",
    "name": "Sales Tax Exemption",
    "code": "SALES",
    "code_engine": "23",
    "status": "1",
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.tax_exemption_type_list - List Tax Exemption Types

This method is used to list tax exemption types.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| name | Name | | | Name |
| code | Ubersmith Code | | | Internal Ubersmith Tax Exemption Code |
| code_engine | Tax Engine Code | | | Tax Engine Exemption Code |
| status | Status | | | 0 = Deactivated<br>1 = Active |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "16": {
      "tax_exemption_type_id": "16",
      "name": "Sales Tax Exemption",
      "code": "SALES",
      "code_engine": "23",
      "status": "1",
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## uber.user_exists - Check whether a Client Exists

This method is used to check if a given login name or client id corresponds to an existing client.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| user_login | Client Login | | | The client account. |

**Example Output**

```
{
  "data": {
    "client_id": "1010"
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

### uber.username_exists - Check Whether a Username Exists

This method is used to check whether the specified username exists in the system.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| username  | Username | Y |  |  |

**Example Output**

```
{
  "data": {
    "fullname": "Johnny Ubersmith",
    "username": "1010",
    "class_id": "1",
    "contact_id": null,
    "id": "1010",
    "type": "client",
    "client_id": "1010",
    "email": "email@ubersmith.com"
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

# client - Client

Functions related to the management of clients, contacts, services, and invoices

## client.ach_add - Add a New Bank Account

This method is used to add a new bank account to a client's account.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| user_login | Client Login | Y | | Client's login name or ID |
| ach_acct | Account Number | Y | | |
| ach_aba | Routing Number | | | |
| ach_type | Account Type | | | C = Checking<br>S = Savings |
| ach_bank | Bank Name | | | |
| fname | First Name | | | |
| lname | Last Name | | | |
| company | Company Name | | | |
| address | Street Address | | | |
| city | City | | | |
| state | State | | | |
| zip | ZIP or Postal Code | | | |
| country | Country | | | ISO 2-character country code |
| phone | Telephone Number | | | |
| email | Email Address | | | |
| set_default | Set as default | | | Value of 1 sets this as default billing method.) |

**Example Output**

```
{
    "data": "225",
    "error_message": "",
    "error_code": null,
    "status": true
}
```

:::: ubersmith

## client.ach_delete - Delete a Bank Account

This method is used to delete a bank account already on file.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| billing_info_id | Bank Account ID | Y | | Account to delete. |
| source | Source | | | Source of deletion request. This shows who initiated the deletion of this bank account. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.ach_update - Update a Bank Account

This method is used to update a bank account already on file.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| billing_info_id | Bank Account ID | Y | | Account to update. |
| ach_acct | Account Number | | | |
| ach_aba | Routing Number | | | |
| ach_type | Account Type | | | |
| ach_bank | Bank Name | | | |
| fname | First Name | | | |
| lname | Last Name | | | |
| company | Company Name | | | |
| address | Street Address | | | |
| city | City | | | |
| state | State | | | |
| zip | ZIP or Postal Code | | | |
| country | Country | | | |
| phone | Telephone Number | | | |
| email | Email Address | | | |
| set_default | Set as default | | | Value of 1 sets this as default billing method.) |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.add - Add a New Client

This method is used to add a new client to the system.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| first | First Name | | | |
| last | Last Name | | | |
| company | Company | | | |
| email | Email Address | | | |
| address | Street Address | | | |
| city | City | | | |
| state | State/Province | | | |
| zip | Zip/Postcode | | | |
| country | Country | | | ISO 2-character country code |
| phone | Telephone Number | | | |
| fax | Fax Number | | | |
| uber_login | Username | | | This is the username that the client will use to log in. If not specified the client id may be used to log in. |
| uber_pass | Password | | | This is the password that the client will use to log in. If not specified, the password will be automatically generated and emailed to the client or left blank, depending on the system settings. |
| grace_due | Due Date Method | | 0 | Method used to calculate invoice due date: 0 = Grace Period (invoice is due 'datepay' days after generation) 1 = Static Due Date (invoice is due on 'datedue' day of the month, only applicable to monthly invoicing) |
| prebill_method | Pre Bill Method | | | Method used in conjunction with Pre Bill Days to determine how far in advance to bill for services. Acceptable values are: 0 - Do not pre bill, 1 - Pre bill up to after Invoice Due Date, 2 - Pre bill up to after Invoice Send Date. |
| prebill_days | Pre Bill Days | | | Number of days used in conjunction with Pre Bill Method to determine how far in advance to bill for services. A value of '0' will invoice up to, and including, either the Invoice Due Date (including any Grace Period) or Invoice Send Date as appropriate. |
| charge_days | Charge Delay | | system_default | The number of days the system should wait after generating an invoice before charging any credit cards. |
| datesend | Invoice Send Date | | system_default | Day of the month client will be invoiced (monthly invoicing only). |
| datedue | Invoice Due Date | | system_default | Day of the month on which the invoice is due. If datedue is less than or equal to datesend, the invoice will be due on that day of the following month. |
| datepay | Grace Period | | system_default | Number of days after which the invoice will be due. |
| referred_by | Referred By | | | |
| discount | Discount Level | | 0 | The default discount level for the client, specified either as a dollar amount or a percentage (the default). |
| discount_type | Discount Type | | 0 | The method for calculating the discount: 0 = percentage 1 = dollar value |
| referred | Referred By | | | This field stores whoever may have referred the client. |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| active | Client Status | | 1 | 1 = Client, 2 = Lead |
| late_fee_scheme_id | Late Fee Schedule | | system_default | Late fee schedule |
| default_renew | Default Renew Date | | | Default renew date for new services |
| prorate_min_days | Prorate Min. Days | | | Minimum number of proratable days (inclusive) |
| brand_id | Brand ID | | system_default | This variable takes the integer brand ID number that corresponds to the client's desired brand. The specific ID number can be obtained on the manage brands page inside the Ubersmith system. If not specified, the default class will be used. |
| retry_every | Retry Interval | | system_default | When a client's credit card is declined, the charge is stored in a queue and is retried at a later date. The number of days between retries is defined by this setting. |
| priority | Ticket Priority | | 1 | This variable takes the integer that corresponds to the desired level of priority given to a client's support tickets.<br>0 = Low<br>1 = Normal<br>2 = High<br>3 = 911 |
| access | Access Settings | | | The access settings for the client. This is an array containing the permissions for the client. |
| strict | Strict Mode | | 0 | This field is used to tell the API script whether or not the new client's Ubersmith login name must be unique.<br>If strict is set to 0 and the specified login name is already in use the client will be added to the system and the API script will return a note saying that their information must be updated inside the system.<br>If strict is set to 1, any duplicate login name will cause the script to return a failure as well as a note explaining why. |
| meta_* | Custom Data | | | Metadata items may be set by passing the values as additional parameters matching the metadata items API field names prefixed with 'meta_'. |
| auto_apply_credit | Auto Apply Credit | | | Automatically apply account credits |

**Example Output**

```
{
  "data": "1264",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.avatar_get - Retrieve a Client Avatar

This method is used to retrieve a client or contact avatar.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | | | |
| contact_id | Contact ID | | | |
| email | Email | | | Email address for Gravatar. This is only relevant when 'gravatar_fallback' is set to true. |
| size | Size | | | Specify 'small' to load a 25 pixel avatar; otherwise a 50 pixel avatar will be returned. |
| gravatar_fallback | Gravatar Fallback | | | Set to true in order to allow the return of a client's Gravatar when it is available. |

**Example Output**

```
{
"data": {
"image_base64": [base64-encoded image data],
"image_type":   "image/png"
},
"error_message": "",
"error_code":    null,
"status":        true
}
```

## client.avatar_set - Set a Client Avatar

This method is used to set a client or contact avatar.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | | | |
| contact_id | Contact ID | | | |
| avatar | Attachment | | | An image file for the avatar should be specified. When using cURL the value for the avatar parameter should be @ where is the avatar you wish to attach. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.billing_dispute_add - Add a new Billing Dispute

This method is used to add a billing dispute for an invoice.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| invoice_id | Invoice ID | Y | | Invoice ID |
| description | Description | Y | | Billing Dispute Description |
| user_opened | User ID | | | Billing Dispute Opened by |

**Example Output**

```
{
  "data": 200,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.billing_dispute_get - Get a Billing Dispute

This method is used to get a billing dispute.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| billing_dispute_id | Billing Dispute ID | Y | | |

**Example Output**

```
{
  "data": {
    "billing_dispute_id": "1",
    "invid: "78",
    "date_opened": "1372173080",
    "date_closed": "0",
    "user_opened": "1",
    "user_closed": "0",
    "description": "A new billing dispute"
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.billing_dispute_list - List billing disputes associated with each invoice

This method is used to list an invoice's billing disputes.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| invoice_id | Invoice ID | Y | | Invoice ID |
| open | Open Dispute Flag | | | Return only open or close disputes, possible values:<br>0 = Closed Disputes,<br>1 = Open Disputes |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "1": {
      "billing_dispute_id": "1",
      "invid: "78",
      "date_opened": "1372173080",
      "date_closed": "0",
      "user_opened": "1",
```

```
        "user_closed": "0",
        "description": "A new billing dispute"
      }
    },
    "error_message": "",
    "error_code": null,
    "status": true
}
```

## client.billing_dispute_update - Update a Billing Dispute

This method is used to update a billing dispute.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| billing_dispute_id | Billing Dispute ID | Y | | Invoice ID |
| description | Description | | | Billing Dispute Description |
| close_dispute | Close Dispute Flag | | | If set to 1, the system will close this billing dispute |
| user_closed | User ID | | | The user ID of the user closing the dispute, only applicable when the 'Close Dispute Flag' is set |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.cc_add - Add a New Credit Card

This method is used to add a new credit card to a client's account.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client Login | Y | | Client's login name or ID. |
| cc_num | Credit Card Number | Y | | Credit card number. Only last 4 digits need be supplied when adding a token |
| cc_expire | Expiration Date | | | MMYY format |
| cc_cvv2 | CVV/CVC Code | | | |
| fname | First Name | | | |
| lname | Last Name | | | |
| company | Company Name | | | |
| address | Street Address | | | |
| city | City | | | |
| state | State | | | |
| zip | ZIP or Postal Code | | | |
| country | Country | | | ISO 2-character country code |
| phone | Telephone Number | | | |
| email | Email Address | | | |
| set_default | Set as default | | | Value of 1 sets this as default billing method.) |
| cc_token | Tokenized Card | | | Tokenized card number or profile id |

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| cc_type | Card Type | | | Only used when passed with cc_token. Options:<br>visa = Visa<br>mastercard = MasterCard<br>amex = American Express<br>discover = Discover<br>dinersclub = Diners Club<br>jcb = JCB<br>maestro = Maestro<br>austbank = Australian BankCard<br>switch = Switch<br>solo = Solo<br>electron = Electron<br>enroute = Enroute<br>delta = Visa Debit |

**Example Output**

```
{
  "data": "226",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.cc_delete - Delete a Credit Card

This method is used to delete a credit card already on file.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| billing_info_id | Bank Account ID | Y | | Account to delete. |
| source | Source | | | Source of deletion request. This shows who initiated the deletion of this bank account. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.cc_info - List a Client's Credit Card Details

This method is used to get a client's credit card information.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | | | |
| billing_info_id | Credit Card ID | | | Credit card to get. |

**Example Output**

```
{
  "data": {
    "64": {
      "city": "Troy",
      "address": "353 Broadway",
      "company": "",
      "zip": "12180",
      "country": "US",
      "last_txn_id": "",
      "lname": "Ubersmith",
      "cc_type": "visa",
      "phone": "212-555-5555",
      "fname": "Johnny",
      "clientid": "1081",
      "billing_info_id": "64",
      "cc_number": "**************1111",
      "cc_cvv2": "***",
      "cc_issuenr": "",
      "cc_expire": "1111",
      "cc_token": "",
      "email": "",
      "state": "NY",
      "payment_type": "cc",
      "is_default": "0"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.cc_update - Update a Credit Card

This method is used to update a credit card already on file.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| billing_info_id | Credit Card ID | Y | | Credit card to update. |
| cc_num | Credit Card Number | | | |
| cc_expire | Expiration Date | | | |
| cc_cvv2 | CVV/CVC Code | | | |
| fname | First Name | | | |
| lname | Last Name | | | |
| company | Company Name | | | |
| address | Street Address | | | |
| city | City | | | |
| state | State | | | |
| zip | ZIP or Postal Code | | | |
| country | Country | | | |
| phone | Telephone Number | | | |
| email | Email Address | | | |

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| set_default | Set as default | | | Value of 1 sets this as default billing method.) |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.comment_list - List a Client's Comments

This method is used to list a client's comments.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| client_id | Client ID | Y | | |
| filter_text | Filter Text | | | An array of strings to match against comments and attachment filenames. Comments which match all terms are returned. |
| client_viewable | Client Viewable | | | If client_viewable needs to be filtered (1 to only show client_viewable) |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "40": {
      "edited": "1311780227",
      "comment": "This is a sample comment.",
      "editor": "",
      "time": "1311780227",
      "clientid": "1263",
      "user": "Administrator",
      "comment_id": "40",
      "client_viewable": "0"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.contact_add - Add a New Contact

This method is used to add a new contact to a client's account.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| client_id | Client ID | Y | | |
| real_name | Contact's Name | | | |
| description | Description | | | |
| phone | Telephone Number | | | |
| email | Email Address | | | |
| login | Login Username | | | This is the username that the contact will use to log in. If not specified it will be left blank and the contact may use "ClientID-ContactID" to log in. |
| password | Password | | | This is the password that the contact will use to log in. If not specified, the password will be left blank and the contact will not be able to log in. |
| prefer_lang | Preferred Interface Language | | system default | Preferred language for the contact. If nothing is provided, it will use the system default. |
| access | Access Settings | | | The access settings for the contact. This is an array containing the permissions for the client. |
| rwhois_contact | RWhois Role | | 0 | RWhois role for the contact. This is a bit field that accepts values between 0 and 7 and designates the contact's role as any combination of POC (1), Tech (2) or Abuse (4). |
| meta_* | Custom Data | | | Metadata items may be set by passing the values as additional parameters matching the metadata items API field names prefixed with 'meta_'. |
| roles | Contact Roles | | | This is an array of role ids to add or remove from the contact. Format: Array[Role ID] => 0|1 |

**Example Output**

```
{
  "data": "281",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.contact_delete - Deactivate a Contact

This method is used to deactivate a contact from a client's account. To reactivate a contact, set active=1 on client.contact_update.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client Login | | | Client's login name or ID. |
| contact_id | Contact ID | Y | | The contact to deactivate. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.contact_facility_add - Add Facility Contact

This method adds a new facility-contact and returns the newly created object. A feature of this method allows for updating facility contacts whose permissions had previously been revoked, their 'active' flag having been set to false. In such cases, the archived ID for the object record will be conserved to ease reconciliation efforts.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| contact_id | Contact ID | Y | | |
| fac_id | Facility ID | Y | | |
| admin | Contact Admin | | | Boolean flag specifying whether the contact is able to manage other contact's facility access for the same client |
| active | Access active | | | Boolean flag specifying whether the contact access is valid for this facility or not. A value of false denotes that this contact has no permissions and overrides all other permission. |
| notification | Notification | | | Contact recieves notifications |
| escalation_priority | Escalation Priority | | | Escalation ordinal for a contact of a given client in a given facility |
| tech_support | Technical Support | | | Contact has ability to recieve assistance from technical support for properties in the facility |
| remote_hands | Remote Hands | | | Contact has the ability to recieve remote manual assistance from technical support for properties in the facility |
| physical_entry | Entry | | | Contact may personally enter the facility |
| thirdparty_entry | Third-Party Entry | | | Contact may enter the facility with a guest |
| notes | Notes | | | Additional notes |

**Example Output**

```
{
"status": true,
"error_code": null,
"error_message":"",
"data": {
"4": {
"id": "4",
"client_id": "1002",
"contact_id": "2",
"fac_id": "1",
"admin": "1",
"active": "1",
"notification": "1",
"escalation_priority": "1",
"tech_support": "0",
"remote_hands": "1",
"physical_entry": "0",
"thirdparty_entry": "0",
"created_ts": "1369324285",
```

```
"created_by": "admin",
"updated_ts": "0",
"updated_by": "",
"notes": "",
"real_name": "Joe User",
"email": "joe@joeuser.com",
"fac_code": "FAC1",
"fac_name": "Facility 1"
},
}
}
```

## client.contact_facility_escalation - Facility Contact Escalation

This method updates a facility-contacts list with new escalation priorities.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | Y | | |
| fac_id | Facility ID | Y | | |
| priorities | Facility priorities | Y | | An array of new escalation priorities, indexed by contact id. The values are the new priority to be set for each facility contact |

**Example Output**

```
{
    "status": true,
    "error_code": null,
    "error_message":"",
    "data": true
}
```

## client.contact_facility_list - List Facility Contacts

This method returns a list of contacts with their escalation and access permissions per facility. If no valid argument is given, a list of all active facility contacts is taken as the default query.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| id | Facility Contact ID | | | Unique ID for contact at a specific facility |
| client_id | Client ID | | | |
| contact_id | Contact ID | | | |
| fac_id | Facility ID | | | |
| admin | Contact Admin | | | Boolean flag specifying whether the contact is able to manage other contact's facility access for the same client |
| active | Access active | | | Boolean flag specifying whether the contact access is valid for this facility or not. A value of false denotes that this contact has no permissions and overrides all other permission. |
| notification | Notification | | | Contact recieves notifications |
| escalation_priority | Escalation Priority | | | Escalation ordinal for a contact of a given client in a given facility |
| tech_support | Technical Support | | | Contact has ability to recieve assistance from technical support for properties in the facility |
| remote_hands | Remote Hands | | | Contact has the ability to recieve remote manual assistance from technical support for properties in the facility |
| physical_entry | Entry | | | Contact may personally enter the facility |
| thirdparty_entry | Third-Party Entry | | | Contact may enter the facility with a guest |
| format | Format | | json | Return format:<br>json = JSON (default)<br>csv = CSV |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
"status": true,
"error_code": null,
"error_message":"",
"data": {
"4": {
"id": "4",
"client_id": "1002",
"contact_id": "2",
"fac_id": "1",
"admin": "1",
"active": "1",
"notification": "1",
"escalation_priority": "1",
"tech_support": "0",
"remote_hands": "1",
"physical_entry": "0",
"thirdparty_entry": "0",
"created_ts": "1369324285",
"created_by": "admin",
"updated_ts": "0",
"updated_by": "",
"notes": "",
"real_name": "Joe User",
"email": "joe@joeuser.com",
"fac_code": "FAC1",
"fac_name": "Facility 1"
},
}
}
```

## client.contact_facility_revoke - Revoke Facility Contact Access

This method revokes all access for a contact at a facility by setting the facility-contact 'active' flag to false. Facility-contact ID's can be obtained through the contact_facility_list API method.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| id | Facility Contact ID | Y | | Unique ID for contact at a specific facility |

**Example Output**

```
{
"status": true,
"error_code": null,
"error_message": "",
"data": {
"4": {
"active": 0
}
}
}
```

::: ubersmith

## client.contact_facility_update - Facility Contact Update

This method updates a facility-contact and returns the updated record.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| id | Facility Contact ID | Y | | Unique ID for contact at a specific facility |
| admin | Contact Admin | | | Boolean flag specifying whether the contact is able to manage other contact's facility access for the same client |
| active | Access active | | | Boolean flag specifying whether the contact's access is valid for this facility. A value of false denotes that this contact has no permissions and is meant to override any other permission specified. |
| notification | Notification | | | Contact recieves notifications |
| escalation_priority | Escalation Priority | | | Escalation ordinal for a contact of a given client in a given facility |
| tech_support | Technical Support | | | Contact has ability to recieve assistance from technical support for properties in the facility |
| remote_hands | Remote Hands | | | Contact has the ability to recieve remote manual assistance from technical support for properties in the facility |
| physical_entry | Entry | | | Contact may personally enter the facility |
| thirdparty_entry | Third-Party Entry | | | Contact may enter the facility with a guest |
| notes | Notes | | | Additional notes |

**Example Output**

```
{
"status": true,
"error_code": null,
"error_message":"",
"data": {
"4": {
"id": "4",
"client_id": "1002",
"contact_id": "2",
"fac_id": "1",
"admin": "1",
"active": "1",
"notification": "1",
"escalation_priority": "1",
"tech_support": "0",
"remote_hands": "1",
"physical_entry": "0",
"thirdparty_entry": "0",
"created_ts": "1369324285",
"created_by": "admin",
"updated_ts": "1369324289",
"updated_by": "admin",
"notes": "",
"real_name": "Joe User",
"email": "joe@joeuser.com",
"fac_code": "FAC1",
"fac_name": "Facility 1"
},
}
}
```

## client.contact_get - Get Contact Details

This method is used to get a contact's details.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| contact_id | Contact ID | | | |
| user_login | Contact Login Name | | | |
| acls | Client ACLs | | | 1 = Include client ACLs |

**Example Output**

```
{
  "data": {
    "real_name": "Johnny Ubersmith",
    "prefer_lang": "1",
    "email_name": "Johnny",
    "last": "Ubersmith",
    "class_id": "1",
    "priority": "1",
    "contact_id": "282",
    "rwhois_contact": "0",
    "audit_tickets": "0",
    "password_changed": "1311780807",
    "client_id": "1263",
    "first": "Johnny",
    "listed_company": "Ubersmith",
    "phone": "",
    "password_timeout": "0",
    "email_domain": "ubersmith.com",
    "description": "Employee",
    "access": {
      "view_profile":"view"
    },
    "password": "{ssha1}5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8",
    "login": "j_uber",
    "email": "tommy@willsmythe.com",
    "active": "1",
    "roles": {
      "1": "The Name",
      "2": "Another Name"
    },
    "acls": {
      "client.profile": {
        "create": "1",
        "read": "1",
        "update": "1",
        "delete": "-1"
      },
      "client.billing_info": {
        "read": "1"
      }
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.contact_list - List a Client's Contacts

This method is used to list a client's contacts.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client Login | | | Client's login name or ID. |
| class_id | Class ID | | | Client's class ID |
| contact_active | Contact Status | | 1 | Set to 0 to retrieve deactivated contacts, 1 for active contacts, or 2 for all contacts |
| active | Client Status | | | Set to 0 to retrieve contacts belonging to deactivated clients, 1 for active clients, or 2 for leads |
| email | Contact Email | | | Retrieve contacts matching given email address |
| metadata | Include Metadata | | | Possible values:<br>0 or blank: Include fields with show_in_list flag set<br> 1: Include all metadata fields<br> array: Specify individual fields to include by entering "metadata[field_name1]=1 metadata[field_name2]=1 etc..." |
| | Metadata Filters | | | Filter list by custom field values, ie. "myfield1=somevalue myfield2=somevalue etc..." |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "282": {
      "real_name": "Johnny Uber",
      "prefer_lang": "1",
      "email_name": "johnny",
      "priority": "1",
      "contact_id": "282",
      "rwhois_contact": "0",
      "audit_tickets": "0",
      "password_changed": "1311780807",
      "client_id": "1263",
      "phone": "",
      "password_timeout": "0",
      "email_domain": "ubersmith.com",
      "description": "title",
      "access": "a:0:{}",
      "password": "{ssha1}5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8",
      "login": "j_uber",
      "email": "johnny@ubersmith.com",
      "active": "1"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.contact_metadata_get - Get a Contact's Metadata

This method is used to get a contact's metadata.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| contact_id | Contact ID | Y | | |

**Example Output**

```
{
  "data": {
    "44": {
      "required": "0",
      "size": "0",
      "prefix": "RWhois Visibility",
      "metagroup_id": "14",
      "default": 0,
      "editable": "0",
      "client_access": "2",
      "priority": "3",
      "class_id": "1",
      "id": "1081",
      "default_val": "0",
      "value": "0",
      "show_in_list": "0",
      "type": "select",
      "rows": "0",
      "cols": "0",
      "suffix": "",
      "metaconf_id": "44",
      "meta_type": "client",
      "options": "\"\",\"0:Display Full Details\",\"1:Hide All Details\",\"2:ARIN Customer Privacy Guidelines\"",
      "variable": "hide_address_in_whois"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.contact_metadata_single - Get a Contact's Metadata Value

This method is used to get a single metadata value from a contact.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| contact_id | Contact ID | Y | | |
| metaconf_id | Metadata Item ID | | | |
| variable | Metadata Variable Name | | | |

**Example Output**

```
{
  "data": "192.168.0.249",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.contact_permission_list - List Client Contact Permissions

This method is used to return a tree/list of client contact permissions.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| contact_id | Contact ID | Y | | |
| resource_name | Resource Name | | | |
| effective | Include Effective Actions | | 0 | Include the effective (inherited) actions along with explicit actions |

**Example Output**

```
{
  "data": {
      "1": {
          "resource_id": "1",
          "name": "*",
          "label": "All Resources",
          "parent_id": "0",
          "resource_action_id": "1",
          "action": {
              "create": "1",
              "read": "1",
              "update": "1",
              "delete": "1"
          },
          "effective": {
              "create": "1",
              "read": "1",
              "update": "1",
              "delete": "1"
          },
          "actions": "",
          "children": {
              "2": {
                  "resource_id": "2",
                  "name": "clientmgr",
                  "label": "Client Manager",
                  "parent_id": "1",
                  "resource_action_id": "1",
                  "action": {
                      "create": "1",
                      "read": "1",
                      "update": "1",
                      "delete": "1"
                  },
                  "actions": "",
              }
          }
      }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

::: ubersmith

## client.contact_permission_set - Set Client Contact Permissions

This method is used to set client contact permissions.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| contact_id | Contact ID | Y | | |
| resource_name | Resource Name | Y | | |
| action | Action | Y | | |
| type | Permission Type | Y | | Available values: allow, deny, inherit |

**Example Output**

```
{
  "data": ""
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.contact_update - Update a Contact

This method is used to update a client's contact.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| contact_id | Contact ID | Y | | The contact to update. |
| real_name | Contact's Name | | | |
| description | Description | | | |
| phone | Telephone Number | | | |
| email | Email Address | | | |
| login | Login Username | | | This is the username that the contact will use to log in. If not specified it will be left blank and the contact may use "ClientID-ContactID" to log in. |
| password | Password | | | This is the password that the contact will use to log in. If not specified, the password will not be changed from its current value. |
| prefer_lang | Preferred Interface Language | | | Preferred language for the contact. If nothing is provided, it will use the system default. |
| access | Access Settings | | | The access settings for the contact. This is an array containing the permissions for the client. |
| rwhois_contact | RWhois Role | | 0 | RWhois role for the contact. This is a bit field that accepts values between 0 and 7 and designates the contact's role as any combination of POC (1), Tech (2) or Abuse (4). |
| active | Contact active | | | Disable or enable the contact. |
| roles | Contact Roles | | | This is an array of role ids to add or remove from the contact. Format: Array[Role ID] => 0|1 |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.count - Count Active Clients

This method is used to count the number of clients in the system.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| brand_id | Brand ID | | | If specified, only clients from the specified brand will be returned. |
| inactive | Active Flag | | 2 | Specify whether to return only active, only inactive, or all clients.<br>1 = Inactive<br>0 = Active<br>2 = All |
| active | Status | | | 1 = Clients<br>2 = Leads<br>0 = Deactivated |

**Example Output**

```
{
  "data": "264",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.credit_add - Add an Account Credit

This method is used to add a new credit to a client's account.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | Y | | Client ID. |
| value | Credit Amount | | 0 | The amount that is to be credited to the client's account. |
| reason | Description | | | This is the "name" of the account credit and is what will appear in any invoices that include it. |
| comment | Additional Information | | | If provided, will be stored as a comment in the credit record. |
| payment_type | Type of Account Credit | Y | other | This field determines what type of credit is being created. Possible values:<br>check = Received Check<br>cash = Received Cash<br>comp = Company Issued<br>cred = Credit card charge<br>other = Other |
| payment_number | Number to be Associated with Credit | | | This field is for any record ID that may need to be kept track of for the credit. Examples include a check number or staff ID. |
| start_date | Start Date | | | This field determines what date the credit becomes applicable. |
| auto_apply | Automatic Application Flag | | 1 | This field determine if this credit can be automatically applied to new invoices. Possible values:<br>2 = Automatically Apply to All Invoices<br>1 = Automatically Apply to New Invoices<br>0 = Do Not Automatically Apply |
| pack_ids | Package IDs | | | An array of packages to this credit may be applied to, the index of each element should be the package ID and the value should be 1 (e.g. pack_ids[89]=1) |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| line_items | Invoice Line Item IDs | | | An array of invoice line items to associate this credit with. The index of each element should be the item_id and the value should be an array containing amount, date_range_start & date_range_end. Example: line_items[1][amount]=4.84&line_items[1][date_range_start]=1/1/2013&line_items[1][date_range_end]=2/1/2013 |
| send_note | Send Credit Note | | 0 | Set to 1 to email the client a note with details of their credit. |

**Example Output**

```
{
  "data": "54",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.credit_apply - Apply a Credit to an Invoice

This method is used to apply a credit to a client's invoice.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| credit_id | Credit ID | Y | | |
| inv_id | Invoice ID | Y | | |
| total | Total | | | The total amount of the credit to be applied to the invoice. The total will be applied to any available services in the invoice. If the 'total' parameter is set, the parameters 'packages' and 'taxes' can not be set. |
| packages | Packages | | | An array of packages with amount for the credit to apply to (e.g. package[1]=100.00). The index of the array is the inv_xml_id of inv_packs. |
| taxes | Taxes | | | An array of taxes with amount for the credit to apply to (e.g. tax[1]=100.00). The index of the array is the tax_id of inv_taxes. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.credit_comment_list - List a Credit's Comments

This method is used to list a credit's comments.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| credit_id | Credit ID | Y | | |
| filter_text | Filter Text | | | An array of strings to match against comments and attachment filenames. Comments which match all terms are returned. |
| client_viewable | Client Viewable | | | If client_viewable needs to be filtered (1 to only show client_viewable) |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "7": {
      "edited": "1311783080",
      "comment": "Comment 1",
      "credit_id": "54",
      "editor": "",
      "time": "1311783080",
      "clientid": "1263",
      "user": "Administrator",
      "comment_id": "7",
      "client_viewable": "0"
    },
    "8": {
      "edited": "1311783086",
      "comment": "Comment 2",
      "credit_id": "54",
      "editor": "",
      "time": "1311783086",
      "clientid": "1263",
      "user": "Administrator",
      "comment_id": "8",
      "client_viewable": "0"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.credit_deactivate - Deactivate an Account Credit

This method is used to deactivate a credit in a client's account.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| user_login | Client Login | Y | | Client's login name or ID. |
| credit_id | Credit ID | Y | | ID of the credit to deactivate. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.credit_get - Get details of an Account Credit

This method is used to get full details of an account credit.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| credit_id | Credit ID | Y | | |

**Example Output**

```
{
  "data": {
    "credit_id": "3",
    "clientid": "1002",
    "reason": "this is a test credit",
    "amount": "-200.00",
    "payment_type": "comp",
    "comments": "",
    "payment_number": "12345",
    "date": "1370232000",
    "balance": "-1.00",
    "active": "1",
    "auto_apply": "1",
    "order_id": "0",
    "pack_ids": {
      "2": {
        "credit_id": "3",
        "pack_id": "2",
        "desserv": "Managed Server",
        "servtype": "mang"
      },
    },
    "line_items": {
      "25": {
        "credit_id": "3",
        "item_id": "25",
        "amount": "13.33",
        "date_range_start": "1367035200",
        "date_range_end": "1367380800",
        "invid": "18",
        "desserv": "Managed Server (mang)",
        "servtype": "mang",
        "value": "100.00",
        "packid": "2",
        "clientid": "1002",
        "inv_date_range_start": "1367035200",
        "inv_date_range_end": "1369627200"
      }
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.credit_list - List a Client's Credits

This method is used to list a client's credits.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| client_id | Client ID | Y | | |
| active | Credit Active Status | Y | 1 | Specify active status to return.<br>0 = Inactive<br>1 = Current<br>2 = All |
| auto_apply | Automatic Application Flag | | | If specified, return only credits with the automatic application flag set to the value:<br>2 = Credits that can be automatically and manually applied to all invoices<br>1 = Credits that can only be automatically and manually applied to new invoices<br>0 = Credits that can only be manually applied to invoices |

**Example Output**

```
{
  "data": {
    "55": {
      "payment_number": "",
      "comments": "",
      "order_id": "0",
      "credit_id": "55",
      "amount": "-10.00",
      "date": "1311739200",
      "reason": "Courtesy",
      "clientid": "1263",
      "active": "1",
      "auto_apply": "1",
      "balance": "-10.00",
      "payment_type": "comp"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.credit_update - Update an Account Credit

This method is used to update a credit on a client's account.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| credit_id | Credit ID | Y | | Credit ID. |
| amount | Credit Amount | | 0 | The original amount credited to the client's account. |
| balance | Credit Balance | | 0 | The current remaining balance of the credit. |
| reason | Description | | | This is the "name" of the account credit and is what will appear in any invoices that include it. |
| payment_number | Number to be Associated with Credit | | | This field is for any record ID that may need to be kept track of for the credit. Examples include a check number or staff ID. |
| auto_apply | Automatic Application Flag | | 1 | This field determine if this credit can be automatically applied to new invoices. Possible values:<br>2 = Automatically Apply to All Invoices<br>1 = Automatically Apply to New Invoices<br>0 = Do Not Automatically Apply |
| pack_ids | Package IDs | | | An array of packages to this credit may be applied to, the index of each element should be the package ID and the value should be 1 (e.g. pack_ids[89]=1) |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| line_items | Invoice Line Item IDs |  |  | An array of invoice line items to associate this credit with. The index of each element should be the item_id and the value should be an array containing amount, date_range_start & date_range_end. Example: line_items[1][amount]=4.84&line_items[1][date_range_start]= 1/1/2013&line_items[1][date_range_end]=2/1/2013 |
| send_note | Send Credit Note |  | 0 | Set to 1 to email the client a note with details of their credit. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.deactivate - Deactivate a Client

This method is used to deactivate a client.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | Y |  |  |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.domain_add - Add a Domain

This method is used to add a new domain to a client's account.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | Y |  |  |
| service_id | Service ID |  |  | Optional field to associate domain with a specific service. |
| name | Domain Name | Y |  | The name of the domain. The TLD may be specified separately using the tld parameter (e.g. name=example tld=com), or included here (e.g. name=example.com). |
| tld | Top-Level Domain |  |  | The TLD for the domain (net,com,org,etc). Required if not included in the name parameter. |
| registered | Registration Date | Y |  | The registration date of the domain. May be specified as a Unix timestamp or a php- compatible date string. |
| expires | Expiration Date | Y |  | Expiration date of the domain, same format as registration date. |
| orderid | Order ID |  |  | Associate domain with a specific order. |
| username | User Name |  |  | User Name for domain. |
| password | Password |  |  | Password for domain. |
| registrar_id | Domain Registrar |  | manual | Supported registrar. Options: namecheap = Namecheap opensrs = OpenSRS enom = Enom nominet = Nominet manual = Manual |

**Example Output**

```
{
  "data": 4,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.domain_list - List a Client's Domains

This method is used to list a client's domains.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| user_login | Client Login | Y | | Client's login name or ID. |
| domain_type | Domain Type | | 0 | 0 = Current (Non-expired) Domains<br>1 = All Domains |
| order_by | Order By | | name | Field to order results by, available fields:<br>name<br>registered<br>expires<br>registrar |
| direction | Direction | | asc | asc = Ascending<br>desc = Descending |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | 0 | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "109": {
      "name": "example",
      "packid": "0",
      "tld": ".com",
      "orderid": "",
      "expires": "1427947200",
      "registered": "1311785232",
      "registrar": "opensrs",
      "username": "",
      "domain_id": "109",
      "clientid": "1263",
      "password": "",
      "active": "1"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.domain_lookup - Look Up a Domain

This method is used to look up a client's domain using WHOIS.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| domain | Domain Name | Y | | The domain to look up. |

**Example Output**

```
{
  "data": "Whois Server Version 2.0\n\nDomain names in the .com and .net domains can now be registered\nwith many different
competing registrars. Go to http://www.internic.net\nfor detailed information.\n\n  Domain Name: UBERSMITH.COM\n  Registrar:
TUCOWS.COM CO.\n  Whois Server: whois.tucows.com\n  Referral URL: http://domainhelp.opensrs.net\n  Name Server:
NS1.UBERSMITH.COM\n  Name Server: NS2.UBERSMITH.COM\n  Name Server: NS3.UBERSMITH.COM\n  Status: ok\n  Updated Date:
20-feb-2011\n  Creation Date: 21-mar-2001\n  Expiration Date: 21-mar-2012\n\n>>> Last update of whois database: Wed, 27 Jul
2011 17:12:00 UTC",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.domain_register - Register a Domain

This method is used to register a client's domain.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| client_id | Client ID | Y | | |
| tld | Top-Level Domain | | | |
| name | Domain Name | | | |
| username | Username | | | |
| password | Password | | | |
| years | Registration Period | | | |
| service_id | Service ID | | | |
| autorenew | Auto-Renew | | | |
| lockdomain | Lock Domain | | | |
| owner_fname | Owner's First Name | | from client | |
| owner_lname | Owner's Last Name | | from client | |
| owner_company | Owner's Company | | from client | |
| owner_email | Owner's Email | | from client | |
| owner_address1 | Owner's Address 1 | | from client | |
| owner_address2 | Owner's Address 2 | | from client | |
| owner_city | Owner's City | | from client | |
| owner_state | Owner's State | | from client | |
| owner_zip | Owner's ZIP or Postal Code | | from client | |
| owner_country | Owner's Country | | from client | |
| owner_phone | Owner's Telephone Number | | from client | |
| owner_fax | Owner's Fax Number | | from client | |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| admin_info | Administrator's Information | | custom | Possible Values:<br>owner<br>custom |
| admin_fname | Administrator's First Name | | | |
| admin_lname | Administrator's Last Name | | | |
| admin_company | Administrator's Company | | | |
| admin_email | Administrator's Email | | | |
| admin_address1 | Administrator's Address 1 | | | |
| admin_address2 | Administrator's Address 2 | | | |
| admin_city | Administrator's City | | | |
| admin_state | Administrator's State | | | |
| admin_zip | Administrator's ZIP or Postal Code | | | |
| admin_country | Administrator's Country | | | |
| admin_phone | Administrator's Telephone Number | | | |
| admin_fax | Administrator's Fax Number | | | |
| billing_info | Billing Information | | custom | Possible Values:<br>owner<br>custom |
| billing_fname | Billing First Name | | | |
| billing_lname | Billing Last Name | | | |
| billing_company | Billing Company | | | |
| billing_email | Billing Email | | | |
| billing_address1 | Billing Address 1 | | | |
| billing_address2 | Billing Address 2 | | | |
| billing_city | Billing City | | | |
| billing_state | Billing State | | | |
| billing_zip | Billing ZIP or Postal Code | | | |
| billing_country | Billing Country | | | |
| billing_phone | Billing Telephone Number | | | |
| billing_fax | Billing Fax Number | | | |
| technical_info | Technical Information | | custom | Possible Values:<br>owner<br>custom |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| technical_fname | Technical First Name | | | |
| technical_lname | Technical Last Name | | | |
| technical_company | Technical Company | | | |
| technical_email | Technical Email | | | |
| technical_address 1 | Technical Address 1 | | | |
| technical_address 2 | Technical Address 2 | | | |
| technical_city | Technical City | | | |
| technical_state | Technical State | | | |
| technical_zip | Technical ZIP or Postal Code | | | |
| technical_country | Technical Country | | | |
| technical_phone | Technical Telephone Number | | | |
| technical_fax | Technical Fax Number | | | |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.domain_transfer - Transfer a Domain

This method is used to transfer a client's domain.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | Y | | |
| tld | Top-Level Domain | | | |
| name | Domain Name | | | |
| username | Username | | | |
| password | Password | | | |
| years | Registration Period | | | |
| service_id | Service ID | | | |
| autorenew | Auto-Renew | | | |
| lockdomain | Lock Domain | | | |
| owner_fname | Owner's First Name | | from client | |
| owner_lname | Owner's Last Name | | from client | |
| owner_company | Owner's Company | | from client | |
| owner_email | Owner's Email | | from client | |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| owner_address1 | Owner's Address 1 | | from client | |
| owner_address2 | Owner's Address 2 | | from client | |
| owner_city | Owner's City | | from client | |
| owner_state | Owner's State | | from client | |
| owner_zip | Owner's ZIP or Postal Code | | from client | |
| owner_country | Owner's Country | | from client | |
| owner_phone | Owner's Telephone Number | | from client | |
| owner_fax | Owner's Fax Number | | from client | |
| admin_info | Administrator's Information | | custom | Possible Values:<br>owner<br>custom |
| admin_fname | Administrator's First Name | | | |
| admin_lname | Administrator's Last Name | | | |
| admin_company | Administrator's Company | | | |
| admin_email | Administrator's Email | | | |
| admin_address1 | Administrator's Address 1 | | | |
| admin_address2 | Administrator's Address 2 | | | |
| admin_city | Administrator's City | | | |
| admin_state | Administrator's State | | | |
| admin_zip | Administrator's ZIP or Postal Code | | | |
| admin_country | Administrator's Country | | | |
| admin_phone | Administrator's Telephone Number | | | |
| admin_fax | Administrator's Fax Number | | | |
| billing_info | Billing Information | | custom | Possible Values:<br>owner<br>custom |
| billing_fname | Billing First Name | | | |
| billing_lname | Billing Last Name | | | |
| billing_company | Billing Company | | | |
| billing_email | Billing Email | | | |
| billing_address1 | Billing Address 1 | | | |
| billing_address2 | Billing Address 2 | | | |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| billing_city | Billing City | | | |
| billing_state | Billing State | | | |
| billing_zip | Billing ZIP or Postal Code | | | |
| billing_country | Billing Country | | | |
| billing_phone | Billing Telephone Number | | | |
| billing_fax | Billing Fax Number | | | |
| technical_info | Technical Information | | custom | Possible Values:<br>owner<br>custom |
| technical_fname | Technical First Name | | | |
| technical_lname | Technical Last Name | | | |
| technical_company | Technical Company | | | |
| technical_email | Technical Email | | | |
| technical_address 1 | Technical Address 1 | | | |
| technical_address 2 | Technical Address 2 | | | |
| technical_city | Technical City | | | |
| technical_state | Technical State | | | |
| technical_zip | Technical ZIP or Postal Code | | | |
| technical_country | Technical Country | | | |
| technical_phone | Technical Telephone Number | | | |
| technical_fax | Technical Fax Number | | | |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.domain_update - Update a Domain

This method is used to add a new domain to a client's account.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| domain_id | Domain ID | Y | | |
| client_id | Client ID | | | |
| service_id | Service ID | | | Optional field to associate domain with a specific service. |
| name | Domain Name | | | The name of the domain. The TLD may be specified separately using the tld parameter (e.g. name=example tld=com), or included here (e.g. name=example.com). |
| tld | Top-Level Domain | | | The TLD for the domain (net,com,org,etc). Required if not included in the name parameter. |
| registered | Registration Date | | | The registration date of the domain. May be specified as a Unix timestamp or a php- compatible date string. |
| expires | Expiration Date | | | Expiration date of the domain, same format as registration date. |
| orderid | Order ID | | | Associate domain with a specific order. |
| username | User Name | | | User Name for domain. |
| password | Password | | | Password for domain. |
| registrar_id | Domain Registrar | | manual | Supported registrar. Options: <br> namecheap = Namecheap <br> opensrs = OpenSRS <br> enom = Enom <br> nominet = Nominet <br> manual = Manual |
| active | Active | | | Options: 0 = Inactive, 1 = Active |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.find_similar - Get a List of Similar Clients

This method is used to return a list of similar clients.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| client_id | Client ID | | | The ID of the client to match against. Either this or client must be specified. |
| client | Client Details | | | An array of client details to match against. Either this or client_id must be specified. |

**Example Output**

```
{
  "data": {
    "1001": {
      "referred_by": "0",
      "ss": "",
      "city": "Troy",
      "address": "353 Broadway",
      "company": "Ubersmith",
      "qblistid": "80000003-1304980324",
      "discount": "0.00",
      "tier_commission_rate": "0.00",
      "zip": "12180",
      "prefer_lang": "1",
      "commission": "0.00",
      "cpass": null,
      "chost": null,
      "country": "US",
      "comments": "",
      "salesperson": null,
      "sales": "0",
      "discount_type": "0",
      "grace_due": "0",
      "prebill_method": "1",
      "prebill_days": "45",
      "metadata": "",
      "tier_sales": "0",
      "commission_rate": "0.00",
      "class_id": "1",
      "priority": "1",
      "datesend": "18",
      "last": "Smith",
      "late_fee_scheme_id": "0",
      "default_renew": "0",
      "password_changed": "1310566873",
      "latest_inv": "1276016884",
      "charge_days": "0",
      "permnote": null,
      "datepay": "16",
      "name_convention": "0",
      "acctmgr": null,
      "default_billing_info": "0",
      "password_timeout": "0",
      "full_name": "Johnny Ubersmith",
      "listed_company": "Ubersmith",
      "commission_type": "0",
      "credit_bool": "13",
      "crv": "0",
      "fax": "1.5185551235",
      "phone": "1.5185551234",
      "first": "Johnny",
      "qbeditseq": "1304980898",
      "datedue": "1",
      "tier_commission_type": "0",
      "referred": "Jane",
      "retry_every": "1",
      "access": "a:1:{s:4:\"cbms\";s:6:\"client\";}",
      "clogin": null,
      "tempnote": null,
      "clientid": "1001",
      "login": "",
      "password": "{ssha1}5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8",
      "created": "1205812800",
      "tier_commission": "0.00",
      "active": "1",
```

```
        "balance": "14329.75",
        "email": "johnny@ubersmith.com",
        "state": "NY",
        "checkname": ""
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.get - Get Client Details

This method is used to get a client's details.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | | | Either this, user_login or email must be provided. |
| user_login | Client Login Name | | | Either this, client_id or email must be provided. |
| email | Client Email Address | | | Either this, user_login or client_id must be provided. |
| metadata | Include Client Metadata | | 0 | 1 = Include Metadata, 0 = No Metadata |
| allclients | All Clients Flag | | 0 | Setting this parameter to 1 allows the system to return the details for deactivated clients. Otherwise only active clients/leads are returned. |
| tags | Client Tags | | | 1 = Include client tags |
| acls | Client ACLs | | | 1 = Include client ACLs |

**Example Output**

```
{
  "status":true,
  "error_code":null,
  "error_message":"",
  "data":{
      "clientid":"1010",
      "first":"Johnny",
      "last":"Ubersmith",
      "checkname":"",
      "company":"Ubersmith",
      "address":"353 Broadway",
      "city":"Troy",
      "state":"NY",
      "zip":"12180",
      "phone":"+1.2125555555",
      "fax":"+1.2125555555",
      "ss":"",
      "email":"johnny@ubersmith.com",
      "comments":"",
      "country":"US",
      "balance":"187559.67",
      "datesend":"1",
      "datepay":"0",
      "password":"{ssha1}g9C693Onu7s1268QWA18S76v027w78dj",
      "active":"1",
      "permnote":null,
      "tempnote":null,
      "priority":"1",
      "class_id":"1",
      "login":"johnny",
      "crv":"0",
      "chost":null,
      "clogin":null,
      "cpass":null,
      "access":{
        "cbms":"client"
      },
      "retry_every":"1",
      "referred":"Jane Ubersmith",
      "referred_by":"0",
      "credit_bool":"5",
```

```
        "commission_rate":"0.00",
        "commission_type":"0",
        "commission":"0.00",
        "tier_commission_rate":"0.00",
        "tier_commission_type":"0",
        "tier_commission":"0.00",
        "sales":"0",
        "tier_sales":"0",
        "discount":"0.00",
        "latest_inv":"1278014814",
        "metadata":null,
        "datedue":"0",
        "grace_due":"0",
        "listed_company":"Ubersmith",
        "full_name":"Johnny Ubersmith",
        "created":"1236056400",
        "password_timeout":"0",
        "password_changed":"1304603745",
        "charge_days":"0",
        "discount_type":"0",
        "default_billing_info":"0",
        "qblistid":"8000000C-1304980335",
        "qbeditseq":"1304980909",
        "prefer_lang":"1",
        "salesperson":null,
        "late_fee_scheme_id":"0",
        "default_renew":"0",
        "acctmgr":null,
        "name_convention":"",
        "inv_balance":"187,559.67",
        "prebill_method": "1",
        "prebill_days": "45",
        "tags":{
          "1":{
              "id":"1",
              "tag":"Abuse",
              "important":"1",
              "color":"ff0000"
          }
        },
        "roles": {
          "1": "The Name",
          "2": "Another Name"
        },
        "acls": {
          "client.profile": {
              "create": "1",
              "read": "1",
              "update": "1",
              "delete": "-1"
          },
          "client.billing_info": {
              "read": "1"
          }
        }
    }
}
```

## client.invoice_charge - Charge an Invoice

This method is used to charge a client's invoice.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| inv_id | Invoice ID | Y | | |
| payment_type | Payment Type | | cc | Specify payment type.<br>cc = Credit Card<br>ach = Bank Account |
| billing_info_id | Payment Method ID | | | |
| paid_in_full | Paid In Full | | | |
| package | Packages | | | |
| tax | Taxes | | | |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.invoice_count - Count Invoices

This method is used to count the number of invoices associated with a client.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| client_id | Client ID | Y | | |
| inv_type_select | Invoice Status | | all | 1 = Paid, 0 = Unpaid, 2 = Disregarded, all = All |

**Example Output**

```
{
  "data": "5",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.invoice_disregard - Disregard an Invoice

This method is used to disregard a client's invoice.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| client_id | Client ID | Y | | |
| inv_id | Invoice ID | Y | | |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.invoice_generate - Generate an Invoice

This method is used to generate an invoice for a client.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| client_id | Client ID | Y | | |
| command | Charge Command | | | To charge the client's card on file, pass "charge" for the command parameter. To capture a previously-authorized transaction, pass "charge_prior_auth" for the command parameter, and the transaction ID of the authorization for the trans_id parameter. |
| duedate | Due Date | | | |
| pre_bill | Pre-Bill Date | | | Date to look ahead to for including services to bill. If not supplied the client's pre-bill setting used. |
| trans_id | Transaction ID | | | |
| include_packs | Included Services | | | Pass an array keyed by service IDs to generate an invoice for a particular set of services to be billed. The values in the array should be set to 1. To exclude all services that would be billed, set this to 0 instead of an array. |
| include_prebills | Included Prebills | | | Pass an array keyed by pre-billed service IDs to generate an invoice for a particular set of services to be pre-billed. The values in the array should be set to 1. To exclude all services that would be pre-billed, set this to 0 instead of an array. |
| include_credits | Included Credits | | | Pass an array keyed by credit IDs to generate an invoice for a particular set of credits to be included. The values in the array should be set to 1. To exclude all credits that would be included, set this to 0 instead of an array. |

**Example Output**

```
{
  "data": {
    "invid": "422",
    "total": 0,
    "time": 1311802259,
    "duedate": 1311912000
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.invoice_get - Get an Invoice

This method is used to get a client's invoice.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| invoice_id | Invoice ID | Y | | |
| format | Format | | json | Return format:<br>json = JSON (default)<br>pdf = PDF<br>html = HTML<br>xml = XML<br>csv = CSV |

**Example Output**

```
{
  "data": {
    "last_late_fee_time": "0",
    "comment": null,
    "qblistid": "",
    "invid": "421",
    "current_packs": {
      "1": {
        "section": "current_packs",
        "packid": "1333",
        "discount": "",
        "invid": "421",
        "details": {
          "singular": "detail"
        },
        "cost": "150.00",
        "quantity": "1.00",
        "attributes": {
          "id": "1"
        },
        "notes": null,
        "paid": "1.00",
        "period": "0",
        "auto_bill": "0",
        "suspend_on": "0",
        "date_range_start": "1311739200",
        "canceled": "0",
        "parentpack": "1332",
        "unit_price": "150.00",
        "cancel_on": "0",
        "value": "150.00",
        "billed": "1",
        "inv_xml_id": "1",
        "suspended": "0",
        "servtype": "setp",
        "desserv": "Hosting Setup Fee (setp)",
        "clientid": "1263",
        "unpaid": "149.00",
        "due": "1311912000",
        "item_id": "2297",
        "ob_label": "package",
        "options": null,
        "retry_pack_id": "0",
        "date_range_end": "0"
      }
    },
    "qbtxnid": "",
    "body": "",
    "sent": "1",
    "late_fee_amount": "5.00000000",
    "account_balance": "149.00",
    "amount_unpaid": "149.00",
    "paid": "2",
    "invnum": "421",
    "id_number": "421",
    "overdue": "1311912000",
    "datepaid": "1311802197",
    "amount": "150.00",
    "taxes": [
```

```
      ],
      "late_fee_recur_method": "2",
      "date": "1311801174",
      "packs": [
        {
          "packid": 1332,
          "attributes": {
            "pack": 1332
          },
          "desserv_only": true,
          "desserv": "Hosting",
          "ob_label": "package"
        },
        {
          "section": "current_packs",
          "packid": "1333",
          "attributes": {
            "pack": "1333"
          },
          "parentpack": 1332,
          "desserv": "Hosting Setup Fee (setp)",
          "ob_label": "package"
        }
      ],
      "client": {
        "city": "",
        "address": "",
        "company": "",
        "zip": "",
        "country": "",
        "last": "smith",
        "first": "will",
        "state": ""
      },
      "late_fee_recur_amount": "5.00000000",
      "reason": "",
      "retryid": "",
      "credits": {
        "2": {
          "invid": "421",
          "comments": "",
          "attributes": {
            "id": "2"
          },
          "credit_id": "56",
          "amount": "-1.00",
          "date": "1311739200",
          "value": "-1.00",
          "inv_xml_id": "2",
          "reason": "",
          "inv_credit_id": "41",
          "ob_label": "credit"
        }
      },
      "summary": {
        "account_credits": "1",
        "total_current_charges": "149",
        "amount_charged": "0",
        "account_balance": "149",
        "unpaid_balance": "",
        "payments": "0",
        "invoice_note": "This invoice has been sent manually without automatically processing a payment.\nPlease contact us if
you wish to arrange an alternate method of payment.\n",
        "current_charges": "150",
        "invnote": "",
        "total_tax": "0",
        "prior_balance": "0"
      },
      "late_fee_delay": "1",
      "qbeditseq": "",
      "clientid": "1263",
      "pre_billed_packs": [

      ],
      "aged": {
        "aged_ninety_onetwenty": 0,
        "aged_sixty_ninety": 0,
        "aged_current": 0,
        "aged_thirty_sixty": 0,
```

```
      "aged_onetwenty_plus": 0,
      "aged_zero_thirty": 0
    },
    "late_fee_recur_interval": "5",
    "due": "1311912000",
    "late_fee_method": "2",
    "after_tni": "1"
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.invoice_list - List a Client's Invoices

This method is used to list a client's invoices.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client Login | | | Client ID |
| brand_id | Brand ID | | | If specified, only invoices from the specified brand will be returned. |
| paid | Paid Status | | all | 0 = Unpaid<br>1 = Paid<br>2 = Disregarded |
| since | Start Date | | | If specified, only invoices generated on or after this date will be returned. |
| order_by | Order By | | client_id | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "417": {
      "comment": null,
      "invid": "417",
      "amount_unpaid": "0.00",
      "paid": "1",
      "datepaid": "1311800378",
      "amount": "1100.00",
      "date": "1311799605",
      "reason": null,
      "clientid": "1263",
      "due": "1311912000",
      "after_tni": "1",
      "open_dispute": "0"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.invoice_payments - List an Invoice's Payments

This method is used to list an invoice's payments.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| invoice_id | Invoice ID | Y | | |

**Example Output**

```
{
  "data": {
    "469": {
      "refunded": "0.00",
      "invid": "418",
      "payment_details": {
        "604": {
          "packid": "1331",
          "refunded": "0.00",
          "amount": "100.00",
          "inv_xml_id": "1",
          "pay_record_id": "469",
          "desserv": "Hosting Service--Setup Fee (setp)",
          "payment_id": "604"
        }
      },
      "qbtxnid": "",
      "card_type": "visa",
      "details": "1111",
      "tax_details": [

      ],
      "credit_id": "0",
      "transaction_id": "1189360436",
      "time": "1311800641",
      "amount": "100.00",
      "money_received": "1",
      "type": "charge",
      "pay_record_id": "469",
      "qbeditseq": "",
      "billing_info_id": "227",
      "clientid": "1263",
      "user": "admin"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.invoice_post_gw_payment - Record a Payment

This method is used to record a payment received outside Ubersmith through Paypal, WorldPay, 2Checkout or an ACH provider.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | Y | | |
| inv_id | Invoice ID | Y | | |
| gateway | Gateway | Y | | Possible values:<br>paypal<br>worldpay<br>2checkout<br>ach |
| amount | Amount | Y | | |
| transaction_id | Transaction ID | Y | | |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.latest_client - Get the Latest Client

This method is used to get the latest client in the system.

**Example Output**

```
{
  "data": 1265,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.list - List Clients

This method is used to list clients in the system.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| short | Short Output Mode | | 0 | If set to 1, only the Client ID (and billing information if billing_info is set) will be returned. |
| client_id | Client ID | | | If specified, only the specified client will be returned. |
| brand_id | Brand ID | | | If specified, only clients from the specified brand will be returned. |
| inactive | Active Flag | | 2 | Specify whether to return only active, only inactive, or all clients.<br>1 = Inactive<br>0 = Active<br>2 = All |
| active | Status | | | 1 = Clients<br>2 = Leads<br>0 = Deactivated |
| qblistid | QuickBooks ID | | | If specified, only the client which matches the specified QuickBooks ID will be returned. |
| tag_id | Tag ID | | | Filter clients by tag |
| tag_ids | Tag IDs | | | Return only clients with the specified tags. An array of tag IDs |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| metadata | Include Metadata | | 0 | Possible values:<br>0 or blank: Include fields with show_in_list flag set<br>1: Include all metadata fields<br>array: Specify individual fields to include by entering "metadata[field_name1]=1 metadata[field_name2]=1 etc..." |
| | Metadata Filters | | | Filter list by custom field values, ie. "myfield1=somevalue myfield2=somevalue etc..." |
| order_by | Order By | | client_id | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "1001": {
      "referred_by": "0",
      "ss": "",
      "city": "Troy",
      "address": "353 Broadway",
      "company": "Ubersmith",
      "qblistid": "80000003-1304980324",
      "discount": "0.00",
      "tier_commission_rate": "0.00",
      "zip": "12180",
      "prefer_lang": "1",
      "commission": "0.00",
      "cpass": null,
      "chost": null,
      "country": "US",
      "comments": "",
      "salesperson": null,
      "sales": "0",
      "discount_type": "0",
      "grace_due": "0",
      "prebill_method": "1",
      "prebill_days": "45",
      "metadata": "",
      "tier_sales": "0",
      "commission_rate": "0.00",
      "class_id": "1",
      "priority": "1",
      "datesend": "18",
      "last": "Smith",
      "late_fee_scheme_id": "0",
      "default_renew": "0",
      "password_changed": "1310566873",
      "latest_inv": "1276016884",
      "charge_days": "0",
      "permnote": null,
      "datepay": "16",
      "name_convention": "0",
      "acctmgr": null,
      "default_billing_info": "0",
      "password_timeout": "0",
      "full_name": "Johnny Ubersmith",
      "listed_company": "Ubersmith",
      "commission_type": "0",
      "credit_bool": "13",
      "crv": "0",
      "fax": "1.5185551235",
      "phone": "1.5185551234",
      "first": "Johnny",
      "qbeditseq": "1304980898",
      "datedue": "1",
      "tier_commission_type": "0",
      "referred": "Jane",
      "retry_every": "1",
      "access": "a:1:{s:4:\"cbms\";s:6:\"client\";}",
```

```
        "clogin": null,
        "tempnote": null,
        "clientid": "1001",
        "login": "",
        "password": "{ssha1}5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8",
        "created": "1205812800",
        "tier_commission": "0.00",
        "active": "1",
        "balance": "14329.75",
        "email": "johnny@ubersmith.com",
        "state": "NY",
        "checkname": ""
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.lookup - Look Up a Client

This method is used to look up a client.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| brand_id | Brand ID | | | |
| active | Client Status | | 1 | 1 = Active<br>0 = Deactivated |

**Example Output**

```
{
  "data": {
    "1092": {
      "clientid": "1092"
    },
    "1065": {
      "clientid": "1065"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.metadata_get - Get a Client's Metadata

This method is used to get a client's metadata.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | Y | | |

**Example Output**

```
{
  "data": {
    "44": {
      "required": "0",
      "size": "0",
      "prefix": "RWhois Visibility",
      "metagroup_id": "14",
      "default": 0,
      "editable": "0",
      "client_access": "2",
      "priority": "3",
      "class_id": "1",
      "id": "1081",
      "default_val": "0",
      "value": "0",
      "show_in_list": "0",
      "type": "select",
      "rows": "0",
      "cols": "0",
      "suffix": "",
      "metaconf_id": "44",
```

```
        "meta_type": "client",
        "options": "\"\",\"0:Display Full Details\",\"1:Hide All Details\",\"2:ARIN Customer Privacy Guidelines\"",
        "variable": "hide_address_in_whois"
      }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.metadata_single - Get a Client's Metadata Value

This method is used to get a single metadata value from a client.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | Y | | |
| metaconf_id | Metadata Item ID | | | |
| variable | Metadata Variable Name | | | |

**Example Output**

```
{
  "data": "192.168.0.249",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.msa_get - Get Client MSA Details

This method is used to get the details of a specified client msa.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | | | |
| client_msa_id | Client MSA ID | | | |
| format | Format | | json | Return format:<br>json = JSON (default)<br>pdf = PDF |

**Example Output**

```
{
  "data": {
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.payment_list - List a Client's Payments

This method is used to list a client's payments.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| client_id | Client ID | Y | | |
| invoice_id | Invoice ID | | | |
| begin | Begin Time | | | A Unix timestamp. If empty, output will begin from the earliest payment. |
| end | End Time | | | A Unix timestamp. If empty, output will end at the most recent payment. |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "1": {
      "pay_record_id": "1",
      "clientid": "1001",
      "invid": "1",
      "amount": "10.00",
      "time": "1372867250",
      "type": "check",
      "details": "",
      "money_received": "1",
      "transaction_id": null,
      "refunded": "0.00",
      "billing_info_id": null,
      "card_type": null,
      "user": "admin",
      "qbtxnid": "",
      "credit_id": "0",
      "avs_code": "",
      "auth_num": "",
      "response_code": "",
      "payment_details": [],
      "tax_details": {
        "1": {
          "tax_payment_id": "1",
          "pay_record_id": "1",
          "tax_id": "1",
          "packid": "1",
          "invid": "1",
          "tax_xml_id": "2",
          "amount": "10.00",
          "refunded": "0.00",
          "code_engine": "",
          "class_id": "1",
          "name": "Tax rate",
          "rate": "0.10000000",
          "active": "1",
          "include_other": "0",
          "qblistid": "",
          "qbvdrlistid": "",
          "name_en_US": "Tax rate"
        }
      }
    },
    "2": {
      "pay_record_id": "2",
      "clientid": "1001",
      "invid": "1",
      "amount": "20.00",
```

```
          "time": "1372867263",
          "type": "cash",
          "details": "",
          "money_received": "1",
          "transaction_id": null,
          "refunded": "0.00",
          "billing_info_id": null,
          "card_type": null,
          "user": "admin",
          "qbtxnid": "",
          "credit_id": "0",
          "avs_code": "",
          "auth_num": "",
          "response_code": "",
          "payment_details": {
            "1": {
              "payment_id": "1",
              "pay_record_id": "2",
              "packid": "1",
              "desserv": "Service (colo)",
              "amount": "20.00",
              "inv_xml_id": "1",
              "refunded": "0.00"
            }
          },
          "tax_details": []
      }
    }
}
```

## client.payment_method_list - List a Client's Payment Methods

This method is used to list a client's payment methods.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | Y | | |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "64": {
      "city": "Troy",
      "address": "353 Broadway",
      "company": "",
      "zip": "12180",
      "country": "US",
      "last_txn_id": "",
      "lname": "Ubersmith",
      "cc_type": "visa",
      "phone": "212-555-5555",
      "fname": "Johnny",
      "clientid": "1081",
      "billing_info_id": "64",
      "cc_num": "1111",
      "cc_expire": "1111",
      "email": "",
      "state": "NY",
      "payment_type": "cc",
      "is_default": "0"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

### client.payment_refund - Refund a payment.

This method is used to refund a payment, either in full or to specific payment items and/or tax payments.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| pay_record_id | Payment Record ID | Y | | |
| note | Notes | | | |
| refund_items | Refund Services | | | An array of amounts keyed by payment_id to be refunded to specific payment items. |
| tax_items | Refund Taxes | | | An array of taxes keyed by tax_id to be refunded to specific tax payments |
| full_refund | Full Refund | | | Boolean to determine if a full refund will be performed. Note that this will override any refund or tax items that are defined. |
| refund_type | Refund Type | | | If 1 - Ubersmith will only mark the refund in the account and will not credit a clients credit card/bank. |

**Example Output**

```
{
  "data": {
    "pay_record_id": "552",
    "clientid": "1005",
    "invid": "513",
    "amount": 10.04,
    "note": "Partial refund.",
    "refund_details":[{
      "amount": ".03",
      "inv_xml_id": 5,
      "packid": "992",
      "desserv": "Web Hosting"
    },
    {
      "amount": 0.01,
      "tax_id": "1",
      "tax_xml_id": 21,
      "packid": "379",
      "name": "NYS Sales Tax"
    }],
    "ref_transaction_id": 140293607
  }
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.paypal_subscription_add - Create a new Paypal Subscription

Generate an HTML form to make a new Paypal subscription; Add the necessary information to the DB

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | Y | | |
| services | A list of services, eg: &services[]=1234&services[]=2345 | | | Services must belong to client_id. If none are provided, defaults to all eligible services. |
| period | Specify a period of services | | | |
| format | Format | | json | Return format:<br>json = JSON (default)<br>html = HTML |

## client.permission_get - Get Client Permission

This method is used to get a client permission.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | Y | | |
| resource_name | Resource Name | Y | | |
| action | Action | Y | | |

**Example Output**

```
{
  "data": ""
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.permission_list - List Client Permissions

This method is used to return a tree/list of client permissions.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | Y | | |
| resource_name | Resource Name | | | |
| effective | Include Effective Actions | | 0 | Include the effective (inherited) actions along with explicit actions |

**Example Output**

```
{
  "data": {
    "1": {
      "resource_id": "1",
      "name": "*",
      "label": "All Resources",
      "parent_id": "0",
      "resource_action_id": "1",
      "action": {
        "create": "1",
        "read": "1",
        "update": "1",
        "delete": "1"
      },
      "effective": {
        "create": "1",
        "read": "1",
        "update": "1",
        "delete": "1"
      },
```

```
            "actions": "",
            "children": {
                "2": {
                    "resource_id": "2",
                    "name": "clientmgr",
                    "label": "Client Manager",
                    "parent_id": "1",
                    "resource_action_id": "1",
                    "action": {
                        "create": "1",
                        "read": "1",
                        "update": "1",
                        "delete": "1"
                    },
                    "actions": "",
                }
            }
        }
    },
    "error_message": "",
    "error_code": null,
    "status": true
}
```

## client.permission_set - Set Client Permissions

This method is used to set client permissions.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | Y | | |
| resource_name | Resource Name | Y | | |
| action | Action | Y | | |
| type | Permission Type | Y | | Available values: allow, deny, inherit |

**Example Output**

```
{
    "data": ""
    "error_message": "",
    "error_code": null,
    "status": true
}
```

## client.quote_comment_list - List a Quote's Comments

This method is used to list a quote's comments.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| quote_id | Quote ID | Y | | |
| filter_text | Filter Text | | | An array of strings to match against comments and attachment filenames. Comments which match all terms are returned. |
| client_viewable | Client Viewable | | | If client_viewable needs to be filtered (1 to only show client_viewable) |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
    "data": {
        "40": {
            "edited": "1311780227",
            "comment": "This is a sample comment.",
```

```
        "editor": "",
        "edited": "",
        "time": "1311780227",
        "quote_id": "1263",
        "user": "Administrator",
        "comment_id": "40",
        "quote_viewable": "0",
        "client_viewable": "0"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.quote_get - Get an Quote

This method is used to get details of a quote.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| quote_id | Quote ID | Y | | |
| format | Format | | json | Return format:<br>json = JSON (default)<br>pdf = PDF |

**Example Output**

```
{
  "data": {
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.quote_list - List a Client's Quotes

This method is used to list a client's quotes.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| client_id | Client ID | | | Client ID |
| order_by | Order By | | client_id | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.reactivate - Reactivate a Client

This method is used to reactivate a client.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| client_id | Client ID | Y | | |

## client.refund_list - List a Client's Refunds

This method is used to list a client's refunds.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| client_id | Client ID | Y | | |
| invoice_id | Invoice ID | | | |
| begin | Begin Time | | | A Unix timestamp. If empty, output will begin from the earliest refund. |
| end | End Time | | | A Unix timestamp. If empty, output will end at the most recent refund. |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "1": {
      "refund_id": "1",
      "clientid": "1001",
      "pay_record_id": "1",
      "invid": "1",
      "amount": "10.00",
      "time": "1372867466",
      "note": "Invoice refund.",
      "ref_transaction_id": "",
      "user": "admin",
      "qbtxnid": "",
      "ref_auth_num": "",
      "ref_response_code": "",
      "refund_details": [],
      "tax_details": {
        "1": {
          "tax_refund_id": "1",
          "refund_id": "1",
          "tax_id": "1",
          "packid": "1",
          "tax_xml_id": "2",
          "amount": "10.00",
          "name": "Tax rate",
          "code_engine": "",
          "class_id": "1",
          "rate": "0.10000000",
          "active": "1",
          "include_other": "0",
          "qblistid": "",
          "qbvdrlistid": "",
          "name_en_US": "Tax rate"
        }
      }
    },
    "2": {
      "refund_id": "2",
      "clientid": "1001",
      "pay_record_id": "2",
```

```
      "invid": "1",
      "amount": "20.00",
      "time": "1372867475",
      "note": "Invoice refund.",
      "ref_transaction_id": "",
      "user": "admin",
      "qbtxnid": "",
      "ref_auth_num": "",
      "ref_response_code": "",
      "refund_details": {
        "1": {
          "refund_item_id": "1",
          "refund_id": "2",
          "inv_xml_id": "1",
          "packid": "1",
          "amount": "20.00",
          "desserv": "Service (colo)"
        }
      },
      "tax_details": []
    }
  }
}
```

## client.renewal_list - List Services for Renewal

This method is used to generate a list of services which are due for renewal in the specified period.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| user_login | Client Login | Y | | Client's login name or ID. |

**Example Output**

```
{
  "data": {
    "824": {
      "referred_by": "0",
      "srv_domains": null,
      "bill_type": "0",
      "comment": null,
      "price": "20.00",
      "packid": "824",
      "cancel_after": "0",
      "srv": "0",
      "discount": "0.00",
      "prorate_reason": null,
      "suspend_bool": "0",
      "quantity": "1.00",
      "cost": "20.00",
      "order_id": "0",
      "prorated": "0",
      "auto_bill": "0",
      "servcode": "0",
      "period": "1",
      "discount_type": "0",
      "metadata": null,
      "username": "",
      "zone_id": "0",
      "bill_prior": "0",
      "suspend_after": "0",
      "post_renew": "0",
      "servername": "",
      "parentpack": "0",
      "lastrenew_success": "1",
      "renewdate": "1293858000",
      "lastrenew": "1291179600",
      "end": "0",
      "cancel_bool": "0",
      "billed": "0",
      "lastrenew_error": "Service has not yet been invoiced for the current period.",
      "start": "1286251200",
      "servtype": "FBBB",
      "billing_info_id": "0",
      "desserv": "Pre-start (not BiA)",
      "clientid": "1081",
      "password": "",
      "plan_id": "27",
```

```
        "created": "1286309839",
        "dev": "0",
        "active": "1",
        "ip_address": "",
        "balance": "0.00"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.send_welcome - Send a Welcome Letter

This method is used to send the welcome letter to a client.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| user_login | Client Login | Y | | Client's login name or ID. |
| service_id | Service ID | Y | | |
| body | Welcome Letter Body | | service plan or system default | |
| subject | Welcome Letter Subject | | service plan or system default | |
| reply | Reply-To Address | | service plan or system default | |
| service | Service Description | | from service | |
| ipaddress | IP Address | | from service | |
| username | Username | | from service | |
| password | Password | | from service | |
| servername | Service Domain or Server Name | | from service | |
| email | Email Address | | client email | |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

# client.service_add - Add a New Service

This method is used to add a new service to a client's account.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | Y | | |
| description | Service Description | | | Required if Service Plan is not specified. |
| plan_id | Service Plan ID | | | Overrides servtype if specified. |
| servtype | Service Plan code | | | Deprecated. |
| parent_id | Parent Service ID | | | To create the new service as a child of a preexisting service, set this parameter to the id of the parent service. The new service will be created as its child. |
| period | Period | | 1 | |
| price | Price | | 0 | |
| bill_type | How Pricing is Calculated | | 0 | Specifies if the renewal cost of the service should be determined by month or by period.<br>0 = By Period<br>1 = By Month<br>If not set, the default value will be taken from the specified service plan. If no service plan is specified, By Period pricing is used. |
| quantity | Quantity | | 1 | |
| status | Service Status | | 1 | 1 = Active<br>2 = Pending<br>3 = Suspended<br>4 = Cancelled |
| discount | Discount Level | | | The discount level for the service, specified either as a dollar amount or a percentage (the default). |
| discount_type | Discount Type | | | The method for calculating the discount:<br>0 = percentage<br>1 = dollar value |
| auto_bill | Automatically Charge | | 0 | This setting determines if the service should be included when the system tries to charge the client's credit card to pay an invoice. If not set, the default value will be taken from the specified service plan. |
| setup | Setup Fee Amount | | 0 | If non-zero, a one time 'setup' child service will be added for that amount. The bill_type and auto_bill settings will match those of the main service being added. |
| post_renew | Post Renew Flag | | 0 | Setting this flag to 1 will enable Post Renew functionality for the service. This is only required for very specific situations and in general isn't needed. |
| bill_prior | Number of Days Before to Bill for Service | | 0 | If specified, the service will be billed in advance for upcoming renewals. Use this in conjunction with the cancel_after setting to prevent renewals for unpaid services. This cannot be used together with post_renew. |
| suspend_bool | Auto Suspend Enable | | 0 | Set to 1 to enable automatic service suspension if an invoice goes unpaid for suspend_after days. |
| suspend_after | Auto Suspend After X Days | | 0 | Number of days after which to suspend unpaid services. |
| cancel_bool | Auto Cancel Enable | | 0 | Set to 1 to enable automatic service cancellation if an invoice goes unpaid for cancel_after days. |
| cancel_after | Auto Cancel After X Days | | 0 | Number of days after which to cancel unpaid services. |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| planned_activation_date | Planned Activation Date | | 0 | The date on which the service is intended to be provided to the client. For reporting purposes only. |
| activation_date | Activation Date | | 0 | The date on which the service gets provided to the client. |
| client_acceptance_date | Client Acceptance Date | | 0 | The date on which the client approved the provided service. For reporting purposes only. |
| expected_cancellation_date | Expected Cancellation Date | | 0 | The date on which the service will likely be ended. For reporting purposes only. |
| start | Billing Start Date | | current date | When the service should start. |
| end | End Date | | 0 | When the service should end. Setting it to 0 will create a service with no end date. |
| userid | Service User ID | | | A user id for the service. |
| pass | Service Password | | | A password for the service. |
| server | Service Domain or Server Name | | | A domain or server name for the service. |
| ip_address | Service IP Address | | | An IP address for the service. |
| options | Service Options Array | | | An array of options which should be enabled for the service. |
| tax_engine_id | Tax Engine ID | | | Tax engine to use for this service, specify 0 to use the internal tax engine. If not set, the default value will be taken from the specified service plan. |
| tax_trans_type_code | Transaction Type Code | | | Transaction Type Code for third party tax engine use. If not set, or passed as an empty string, the system will always use the code set in the service plan. |
| tax_situs_code | Situs Rule | | | Situs Rule for third party tax engine use. If not set, or passed as an empty string, the system will always use the code set in the service plan. |
| tax_zip_code | Zip Code | | | Zip Code for third party tax engine use. |
| tax_p2p_zip_code | Point to Point Zip Code | | | Point to Point Zip Code for third party tax engine use. |
| tax_trans_type_code_setup | Setup Fee Transaction Type Code | | | Setup Fee Transaction Type Code for third party tax engine use. If not set, or passed as an empty string, the setup fee service will be set to use the plan default. |
| tax_situs_code_setup | Setup Fee Situs Rule | | | Setup Fee Situs Rule for third party tax engine use. If not set, or passed as an empty string, the setup fee service will be set to the use the plan default. |
| tax_zip_code_setup | Setup Fee Zip Code | | | Setup Zip Code for third party tax engine use. |
| tax_p2p_zip_code_setup | Point to Point Setup Fee Zip Code | | | Point to Point Setup Zip Code for third party tax engine use. |
| taxes | Taxes | | | An array of tax ids which should be enabled for the service. |
| setup_taxes | Setup Taxes | | | An array of tax ids which should be enabled for the setup fee (if applicable). |
| comment | Service Comment | | | If specified, this text will be added as a comment on the service. |
| custom_notes | Service Notes | | | An array of custom service notes. |
| meta_* | Custom Data | | | Metadata items may be set by passing the values as additional parameters matching the metadata items API field names prefixed with 'meta_'. |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| zone_id | Zone ID | | | Assigned location for reporting purposes. |
| balance | Balance | | | Balance of next billing period |
| prorated | Prorated | | 0 | Flag that the balance is prorated |
| renewdate | Renewal Date | | | Service renewal date |
| contract_term_id | Contract Term ID | | | Contract to be associated with this service |
| rate_plan_id | Rate Plan ID | | | Rate plan to be applied to this service |

**Example Output**

```
{
  "data": 200,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.service_comment_list - List a Service's Comments

This method is used to list a service's comments.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| service_id | Service ID | Y | | |
| filter_text | Filter Text | | | An array of strings to match against comments and attachment filenames. Comments which match all terms are returned. |
| client_viewable | Client Viewable | | | If client_viewable needs to be filtered (1 to only show client_viewable) |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "29": {
      "edited": "1312252800",
      "comment": "this is a nice comment",
      "packid": "354",
      "editor": "",
      "time": "1312252800",
      "clientid": "1010",
      "user": "Administrator",
      "comment_id": "29",
      "client_viewable": "0"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.service_deactivate - Deactivate a Service

This method is used to deactivate a service associated with a client.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| client_id | Client ID | Y | | |
| service_id | Service ID | Y | | ID of the service to deactivate. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.service_get - Get a Service

This method is used to get a service associated with a client.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| service_id | Service ID | Y | | |
| metadata | Metadata Flag | | 0 | If set to 1, metadata items will be included. |
| modules | Modules Flag | | 0 | If set to 1, service module data will be included. |
| bill_date | Bill Date | | | If specified, return the effective cost for the date, default to current renewal period |
| notes | Notes Flag | | 0 | If set to 1, service notes and service plan notes will be included. |

**Example Output**

```
{
  "data": {
    "referred_by": "0",
    "srv_domains": null,
    "bill_type": "0",
    "comment": null,
    "price": "35.48",
    "packid": "1334",
    "cancel_after": "0",
    "srv": "0",
    "discount": "0.00",
    "title": "Hosting Plan",
    "prorate_reason": null,
    "tax_engine_id": 3,
    "tax_trans_type_code": "010101",
    "tax_situs_code": "07",
    "tax_zip_code": "12180",
    "tax_p2p_zip_code": "12180",
    "suspend_bool": "0",
    "quantity": "1.00",
    "cost": "35.48",
    "order_id": "0",
    "prorated": "0",
    "auto_bill": "0",
    "servcode": "0",
    "period": "0",
    "unpaid_balance": 0,
    "discount_type": "0",
    "username": "",
    "zone_id": "0",
    "bill_prior": "0",
    "suspend_after": "0",
    "post_renew": "0",
    "servername": "",
    "parentpack": "379",
    "lastrenew_success": "0",
    "renewdate": "1314072000",
    "lastrenew": "1312171200",
    "end": "0",
```

```
    "cancel_bool": "0",
    "billed": "0",
    "lastrenew_error": "",
    "start": "1312171200",
    "servtype": "",
    "billing_info_id": "0",
    "desserv": "Price Adjustment",
    "clientid": "1005",
    "password": "",
    "plan_id": "44",
    "created": "1312229764",
    "dev": "0",
    "active": "1",
    "ip_address": "",
    "balance": "0.00",
    "discount_start_date": "1314072000",
    "discount_end_date": "1324072000",
    "discount_value": "0.00",
    "balance": "0.00"
    "options": {
      "1": {
        "option_id": "1",
        "spo_id": "1",
        "packid": "1334",
        "price": "100.00",
        "setup": "500.00",
        "cost": null,
        "access": "0",
        "discount_type": "0",
        "discount": "0.00",
        "inv_hide": "0",
        "spo_val": "",
        "spg_id": "1",
        "variable": "upgrade",
        "spug_id": 0,
        "spog_id": 0,
        "description": "Upgrade option",
        "spo_description": "Upgrade option",
        "spg_name": "Upgrade",
        "spug_name": "Ungrouped",
        "spog_name": "Ungrouped"
      }
    },
    "taxes": {
      "1": {
        "tax_id": "1",
        "code_engine": "",
        "class_id": "1",
        "name": "Tax rate",
        "rate": "0.10000000",
        "active": "0",
        "include_other": "0",
        "qblistid": "",
        "qbvdrlistid": "",
        "packid": "1334",
        "tax_exemption_id_list": []
      }
    },
    "metadata": {
      "cp_account_domain": "cpanel.example.com",
      "cp_account_username": "username",
      "cp_account_password": "password",
    },
    "notes": {
      "1": {
        "id": "1",
        "packid": "1334",
        "spn_id": "1",
        "note": "Service Note"
      }
    },
    "plan_notes": {
      "1": {
        "spn_id": "1",
        "spn_plan_id": "44",
        "spn_note": "Service Plan Note",
        "spn_priority": "1"
      }
    },
```

```
    "devices": {
      "827": {
        "dev": "827",
        "dev_desc": "Server 32",
        "label": "server32.yul",
      }
    },
    "connections": {
      "22328": {
        "connection_id": "22328",
        "name": "Here to There"
      }
    },
    "tickets": {
      "113": {
        "ticket_id": "113",
        "subject": "Help!"
      }
    },
    "locations": [
      {
        "facility": {
          "fac_id": "1",
          "fac_code": "NL",
          "fac_name": "Netherlands"
        },
        "zone": {
          "zone_id": "2",
          "zone_code": "NL2",
          "zone_name": "NL Zone 2"
        },
        "cage": {
          "cage_id": "14",
          "cage_code": "NL2CO",
          "cage_name": "Cage Other"
        },
        "row": {
          "row_id": "14",
          "row_code": "NL2COSR",
          "row_name": "Some Row"
        },
        "rack": {
          "rack_id": "17",
          "rack_code": "NL2CORO",
          "rack_name": "Rack Other"
        }
      }
    ],
    "ip_assignments": {
      "3": {
        "assign_id": "3",
        "address": "8.9.10.11",
        "assign_description": "Given"
      }
    },
    "domains": {
      "2": {
        "domain_id": "2",
        "domain": "example.com"
      }
    },
    "dns_zones": {
      "2": {
        "zone_id": "2",
        "name": "sub.example.com"
      }
    },
    "modules": {
      "backup_billing": true,
      "bw_billing": true
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.service_list - List a Client's Services

This method is used to list a client's services.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | | | |
| brand_id | Brand ID | | | |
| plan_id | Service Plan ID | | | |
| parentpack | Parent Service ID | | | |
| period | Period | | | |
| pack_type_select | Service Selection | | 0 | 0 = all active recurring services and unpaid one-time services<br>1 = active recurring services<br>2 = active one-time services<br>3 = all services<br>4 = all active services |
| order_id | Order ID | | | ID of order that created the service |
| bill_date | Bill Date | | | If specified, return the effective cost for the date, default to today |
| metadata | Include Metadata | | 0 | Possible values:<br>0 or blank: Include fields with show_in_list flag set<br> 1: Include all metadata fields<br> array: Specify individual fields to include by entering "metadata[field_name1]=1 metadata[field_name2]=1 etc..." |
| options | Include Service Options | | | Set to 1 to include service options in output. |
| devices | Include Devices | | | Set to 1 to include associated devices in output. |
| modules | Include Service Modules | | | Set to 1 to include service modules in output. |
| notes | Include Notes | | | Set to 1 to include service notes and service plan notes in output. |
| | Metadata Filters | | | Filter list by custom field values, ie. "myfield1=somevalue myfield2=somevalue etc..." |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "243": {
      "referred_by": "0",
      "bill_type": "0",
      "comment": null,
      "price": "100000.00",
      "packid": "243",
      "cancel_after": "0",
      "discount": "0.00",
      "code": "FBBB",
      "title": "Hosting Plan",
      "prorate_reason": null,
      "tax_engine_id": 3,
      "tax_trans_type_code": "010101",
      "tax_situs_code": "07",
      "tax_zip_code": "12180",
      "tax_p2p_zip_code": "12180",
      "suspend_bool": "0",
      "quantity": "1.00",
```

```
"cost": "100000.00",
"order_id": "0",
"prorated": "0",
"auto_bill": "0",
"period": "1",
"discount_type": "0",
"username": "",
"zone_id": "0",
"bill_prior": "0",
"suspend_after": "0",
"post_renew": "0",
"servername": "",
"parentpack": "0",
"lastrenew_success": "1",
"renewdate": "1294117200",
"lastrenew": "1291438800",
"end": "0",
"cancel_bool": "0",
"billed": "1",
"lastrenew_error": "Service has not yet been invoiced for the current period.",
"start": "1265259600",
"billing_info_id": "0",
"desserv": "DELETE",
"clientid": "1011",
"password": "",
"plan_id": "27",
"created": "1265309281",
"active": "1",
"ip_address": "",
"balance": "0.00",
"metadata": {
  "cp_account_domain": "cpanel.example.com",
  "cp_account_username": "username",
  "cp_account_password": "password",
},
"options": {
  "1": {
    "option_id": "1",
    "spo_id": "1",
    "packid": "243",
    "price": "1000.00",
    "setup": "500.00",
    "cost": null,
    "access": "0",
    "discount_type": "0",
    "discount": "0.00",
    "inv_hide": "0",
    "spo_val": "",
    "spg_id": "1",
    "variable": "upgrade",
    "spug_id": 0,
    "spog_id": 0,
    "description": "Upgrade option",
    "spo_description": "Upgrade option",
    "spg_name": "Upgrade",
    "spug_name": "Ungrouped",
    "spog_name": "Ungrouped"
  }
},
"devices": {
  "22": {
    "row_id": "1",
    "packid": "243",
    "company": "Ubersmith",
    "label": "us.ubersmith.com",
    "devtype_group_id": "2",
    "location": "dlfra.dflr.dflc.dflz.eqxtry:39",
    "rack_code": "dlfra",
    "rack_pos": "39",
    "parent": "0",
    "fac_name": "Troy",
    "power_status": null,
    "down": "3",
    "total": "5",
    "fac_code": "eqxtry",
    "last": "Uber",
    "type_id": "12",
    "up": "1",
    "zone_id": "1",
```

```
            "cage_name": "default Cage",
            "row_name": "Default Row",
            "rack_id": "1",
            "type": "Virtual IP",
            "cage_code": "dflc",
            "row_code": "dflr",
            "phone": "+1.2125555555",
            "first": "Johnny",
            "rack_units" : "1",
            "height": "45.000000",
            "width": "482.60000",
            "depth": "609.60000",
            "front_image_file_id": "19",
            "back_image_file_id": "0"
            "height_display_value": 1.75,
            "height_display_unit": "inches",
            "width_display_value": 19,
            "width_display_unit": "inches",
            "depth_display_value": 2,
            "depth_display_unit": "feet",
            "disabled": "1",
            "cage_id": "1",
            "desserv": "server.ubersmith.com",
            "clientid": "1010",
            "devtype_group": "Network",
            "fac_id": "1",
            "zone_code": "dflz",
            "dev_desc": "us.ubersmith.com",
            "warn": "0",
            "zone_name": "Default Zone",
            "rack_name": "Default Rack",
            "email": "support@ubersmith.com",
            "owner_type": "PACKAGE",
            "owner": "1096",
            "active": "1",
            "dev": "22"
          }
        },
        "modules": {
          "backup_billing": true,
          "bw_billing": true
        },
        "notes": {
          "1": {
            "id": "1",
            "packid": "243",
            "spn_id": "1",
            "note": "Service Note"
          }
        },
        "plan_notes": {
          "1": {
            "spn_id": "1",
            "spn_plan_id": "27",
            "spn_note": "Service Plan Note",
            "spn_priority": "1"
          }
        },
      }
    },
    "error_message": "",
    "error_code": null,
    "status": true
}
```

## client.service_metadata_get - Get a Service's Metadata

This method is used to get a service's metadata.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| service_id | Service ID | Y | | |

**Example Output**

```
{
  "data": {
    "21": {
      "required": "0",
      "size": "0",
      "prefix": "Unit",
      "metagroup_id": "6",
      "default": 0,
      "editable": "0",
      "client_access": "0",
      "priority": "3",
      "class_id": "1",
      "id": "233",
      "default_val": "",
      "value": "",
      "show_in_list": "0",
      "type": "select",
      "rows": "0",
      "cols": "0",
      "suffix": "",
      "metaconf_id": "21",
      "meta_type": "pack",
      "options": "\"\",\"hr:Hours\",\"min:Minutes\"",
      "variable": "ticket_unit"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.service_metadata_single - Get a Service's Metadata Value

This method is used to get a single metadata value from a service.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| service_id | Service ID | Y | | |
| metaconf_id | Metadata Item ID | | | |
| variable | Metadata Variable Name | | | |

**Example Output**

```
{
  "data": "mb",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

### client.service_module_call - Call a Service Module Function

This method is used to execute a service module function.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| service_id | Service ID | Y | | |
| module_id | Module Name | Y | | |
| function | Function Name | Y | | |
| module_params | Parameters | | | Array |

**Example Output**

```
{
  "data": {
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

### client.service_note_add - Add a New Service Note

This method is used to add a service note.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| service_id | Service ID | Y | | |
| spn_id | Service Plan Note ID | | | If specified, the service note will be customized based on an existing service plan note. |
| note | Note Text | Y | | The note text. Optional if service plan note id is specified. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

### client.service_note_delete - Delete a Service Note

This method is used to delete a service note.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| note_id | Note ID | Y | | The note to delete. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.service_note_update - Update a Service Note

This method is used to update a service note.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| note_id | Note ID | Y | | The note to update. |
| note | Note Text | Y | | The note text. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.service_prorate - Prorate a Service

This method is used to prorate a service associated with a client.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| service_id | Service ID | Y | | |
| new_renew | New Renewal Date | | today | |
| new_balance | Desired Balance | | | If not set, the system will determine the prorated balance based on the specified date. |
| reason | Prorate Reason | | | If not set, the system will provide a default pro-rate reason based on the specified date. |

**Example Output**

```
{
  "data": "0.00",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.service_storage_delete - Delete a Service's Storage

This method is used to delete a service's storage.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| service_id | Service ID | Y | | |
| item | Item | Y | | |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": true
}
```

## client.service_storage_get - Get a Service's Storage

This method is used to get a service's storage.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| service_id | Service ID | Y | | |
| item | Item | Y | | |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": "passed"
}
```

## client.service_storage_list - List a Service's Storage

This method is used to list a service's storage.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| service_id | Service ID(s) | | | A single service ID or an array of service IDs |
| items | Item(s) | | | A single item or an array of items |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "8": {
      "desserv":"Some Service Description",
      "baz":"val"
      "foo":"bar",
    },
    "38": {
      "desserv":"Another Service Description",
      "baz":"rum"
      "foo":"beer",
    }
  }
}
```

## client.service_storage_set - Set a Service's Storage

This method is used to set a service's storage.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| service_id | Service ID | Y | | |
| item | Item | Y | | |
| value | Value | Y | | |
| encrypt | Encrypt | | | |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": true
}
```

## client.service_update - Update a Service

This method is used to update a service associated with a client.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| service_id | Service ID | Y | | |
| description | Service Description | | | Required if Service Plan is not specified. |
| plan_id | Service Plan ID | | | Overrides servtype if specified. |
| servtype | Service Plan code | | | Deprecated. |
| parent_id | Parent Service ID | | | To create the new service as a child of a preexisting service, set this parameter to the id of the parent service. The new service will be created as its child. |
| period | Period | | 1 | |
| price | Price | | 0 | |
| bill_type | How Pricing is Calculated | | 0 | Specifies if the renewal cost of the service should be determined by month or by period.<br>0 = By Period<br>1 = By Month<br>If not set, the default value will be taken from the specified service plan. If no service plan is specified, By Period pricing is used. |
| quantity | Quantity | | 1 | |
| status | Service Status | | 1 | 1 = Active<br>2 = Pending<br>3 = Suspended<br>4 = Cancelled |
| discount | Discount Level | | | The discount level for the service, specified either as a dollar amount or a percentage (the default). |
| discount_type | Discount Type | | | The method for calculating the discount:<br>0 = percentage<br>1 = dollar value |
| auto_bill | Automatically Charge | | 0 | This setting determines if the service should be included when the system tries to charge the client's credit card to pay an invoice. If not set, the default value will be taken from the specified service plan. |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| post_renew | Post Renew Flag | | 0 | Setting this flag to 1 will enable Post Renew functionality for the service. This is only required for very specific situations and in general isn't needed. |
| bill_prior | Number of Days Before to Bill for Service | | 0 | If specified, the service will be billed in advance for upcoming renewals. Use this in conjunction with the cancel_after setting to prevent renewals for unpaid services. This cannot be used together with post_renew. |
| suspend_bool | Auto Suspend Enable | | 0 | Set to 1 to enable automatic service suspension if an invoice goes unpaid for suspend_after days. |
| suspend_after | Auto Suspend After X Days | | 0 | Number of days after which to suspend unpaid services. |
| cancel_bool | Auto Cancel Enable | | 0 | Set to 1 to enable automatic service cancellation if an invoice goes unpaid for cancel_after days. |
| cancel_after | Auto Cancel After X Days | | 0 | Number of days after which to cancel unpaid services. |
| planned_activation_date | Planned Activation Date | | 0 | The date on which the service is intended to be provided to the client. For reporting purposes only. |
| activation_date | Activation Date | | 0 | The date on which the service gets provided to the client. |
| client_acceptance_date | Client Acceptance Date | | 0 | The date on which the client approved the provided service. For reporting purposes only. |
| expected_cancellation_date | Expected Cancellation Date | | 0 | The date on which the service will likely be ended. For reporting purposes only. |
| start | Billing Start Date | | | When the service should start. |
| end | End Date | | 0 | When the service should end. Setting it to 0 will create a service with no end date. |
| lastrenew | Last Renewal Date | | | The last renewal date of the service. |
| renewdate | Next Renewal Date | | | The next renewal date of the service. |
| userid | Service User ID | | | A user id for the service. |
| pass | Service Password | | | A password for the service. |
| server | Service Domain or Server Name | | | A domain or server name for the service. |
| ip_address | Service IP Address | | | An IP address for the service. |
| options | Service Options Array | | | An array of options which should be enabled for the service. |
| tax_engine_id | Tax Engine ID | | | Tax engine to use for this service, specify 0 to use the internal tax engine. |
| tax_trans_type_code | Transaction Type Code | | | Transaction Type Code for tax engine use. If set to blank, this value will be inherited from the specified service plan and will change when the plan's value changes. |
| tax_situs_code | Situs Rule | | | Situs Rule for tax engine use. If set to blank, this value will be inherited from the specified service plan and will change when the plan's value changes. |
| tax_zip_code | Zip Code | | | Zip Code for tax engine use. |
| tax_p2p_zip_code | Point to Point Zip Code | | | Point to Point Zip Code for tax engine use. |
| taxes | Taxes | | | An array of tax ids which should be enabled for the service. |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| meta_* | Custom Data | | | Metadata items may be set by passing the values as additional parameters matching the metadata items API field names prefixed with 'meta_'. |
| zone_id | Zone ID | | | Assigned location for reporting purposes. |
| prorate | Pro-rate Price Change | | | 1 = Pro-rate price change to the specified day,0 = Do not pro-rate price change. |
| prorate_date | Pro-rate Date | | | If specified together with prorate=1, this will be the effective date of the price change. |
| billed | Billed Status | | | The updated invoice status 0=unbilled, 1=billed, 2=pre-billed |
| contract_term_id | Contract Term ID | | | Contract to be associated with this service |
| rate_plan_id | Rate Plan ID | | | Rate plan to be applied to this service |
| auto_adjustment | Auto Proration Adjustment | | | Automatically apply a proration adjustment if the service is being canceled<br>0=No, 1=Yes |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.set_login - Set a Client's Login

This method is used to set a client's login.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | Y | | |
| uber_pass | New Password | | | |
| uber_login | New Login Name | | | |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.storage_delete - Delete a Client's Storage

This method is used to delete a client's storage.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| client_id | Client ID | Y | | |
| item | Item | Y | | |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": true
}
```

## client.storage_get - Get a Client's Storage

This method is used to get a client's storage.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| client_id | Client ID | Y | | |
| item | Item | Y | | |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": "passed"
}
```

## client.storage_list - List a Client's Storage

This method is used to list a client's storage.

| Parameter | Name | Required | Default | Description |
|-----------|---------|----------|---------|-------------|
| client_id | Client ID(s) | | | A single client ID or an array of client IDs |
| item | Item(s) | | | A single item or an array of items |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "1001": {
      "listed_company":"Some Company",
      "baz":"val"
      "foo":"bar",
    },
    "1002": {
      "listed_company":"Doe, John",
      "baz":"rum"
      "foo":"beer",
    }
  }
}
```

## client.storage_set - Set a Client's Storage

This method is used to set a client's storage.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | Y | | |
| item | Item | Y | | |
| value | Value | Y | | |
| encrypt | Encrypt | | | |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": true
}
```

## client.tag - Add a tag to a Client

This method is used to add a tag or tags to a client.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID(s) | Y | | A single client ID, or an array of client IDs |
| tag_id | Tag ID(s) | | | A single tag ID or an array of tag IDs |
| tag_name | Tag Name(s) | | | A single tag name or an array of tags names. If no tag with by that name exists, it will be created. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.tax_exemption_add - Add a new Tax Exemption

This method is used to add a tax exemption for a client.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | | | Client ID |
| tax_exemption_type_id | Tax Exemption Type ID | | | Tax Exemption Type ID |
| certificate_id | Certificate ID | | | Certificate ID |
| start_date | Effective Date | | | Effective Date |
| end_date | Expiration Date | | | Expiration Date. Set to 0 for none. |
| status | Status | | | -1 = Pending Approval<br>0 = Deactivated<br>1 = Active |
| pdf | Certificate PDF | | | PDF for certificate. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.tax_exemption_get - Get a Client's Tax Exemption

This method is used to get a client's tax exemption.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| tax_exemption_id | Tax Exemption ID | Y | | Tax Exemption ID |

**Example Output**

```
{
  "data": {
    "tax_exemption_id": "11",
    "client_id": "1001",
    "tax_exemption_type_id": "47",
    "certificate_id": "AKCJ-1",
    "file_id": "87",
    "start_date": "1311739301",
    "end_date": "1311799605",
    "status": "-1",
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.tax_exemption_list - List a Client's Tax Exemptions

This method is used to list a client's tax exemptions.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | | | Client ID |
| tax_exemption_type_id | Tax Exemption Type ID | | | Tax Exemption Type ID |
| certificate_id | Certificate ID | | | Certificate ID |
| start_date | Effective Date | | | If specified, only tax exemptions effective on or after this date will be returned. |
| end_date | Expiration Date | | | If specified, only tax exemptions effective before this date will be returned. |
| status | Status | | | -1 = Pending Approval<br>0 = Deactivated<br>1 = Active |
| active | Active Exemptions | | | If set to 1 or TRUE, only active and unexpired tax exemptions will be returned. |
| inactive | Inactive Exemptions | | | If set to 1 or TRUE, only inactive and/or expired tax exemptions will be returned. |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "11": {
      "tax_exemption_id": "11",
      "client_id": "1001",
      "tax_exemption_type_id": "47",
      "certificate_id": "AKCJ-1",
      "file_id": "87",
      "start_date": "1311739301",
      "end_date": "1311799605",
      "status": "-1",
```

```
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.tax_exemption_update - Update a Client's Tax Exemption

This method is used to update a Client's tax exemption.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| tax_exemption_id | Tax Exemption ID | Y | | Tax Exemption ID |
| client_id | Client ID | | | Client ID |
| tax_exemption_type_id | Tax Exemption Type ID | | | Tax Exemption Type ID |
| certificate_id | Certificate ID | | | Certificate ID |
| start_date | Effective Date | | | Effective Date |
| end_date | Expiration Date | | | Expiration Date. Set to 0 for none. |
| status | Status | | | -1 = Pending Approval<br>0 = Deactivated<br>1 = Active |
| pdf | Certificate PDF | | | PDF for certificate. Set to FALSE to remove current file. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.untag - Untag to a Client

This method is used to remove a tag from a client.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| client_id | Client ID(s) | Y | | A single client ID, or an array of client IDs |
| tag_id | Tag ID(s) | | | A single tag ID or an array of tag IDs |
| tag_name | Tag Name(s) | | | A single tag name or an array of tags names. If no tag with by that name exists, it will be created. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## client.update - Update a Client

This method is used to update a client's account information.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | Y | | |
| first | First Name | | | |
| last | Last Name | | | |
| company | Company | | | |
| email | Email Address | | | |
| address | Street Address | | | |
| city | City | | | |
| state | State/Province | | | |
| zip | Zip/Postcode | | | |
| country | Country | | | |
| phone | Telephone Number | | | |
| fax | Fax Number | | | |
| uber_login | Username | | | This is the username that the client will use to log in. If not specified the client id may be used to log in. |
| credit_bool | Billing Method | | 0 | The default client billing method to use. Acceptable ID values are obtained from the client.list_payment_method call. |
| grace_due | Due Date Method | | 0 | Method used to calculate invoice due date:<br>0 = Grace Period (invoice is due 'datepay' days after generation)<br>1 = Static Due Date (invoice is due on 'datedue' day of the month, only applicable to monthly invoicing) |
| prebill_method | Pre Bill Method | | | Method used in conjunction with Pre Bill Days to determine how far in advance to bill for services. Acceptable values are: 0 - Do not pre bill, 1 - Pre bill up to after Invoice Due Date, 2 - Pre bill up to after Invoice Send Date. |
| prebill_days | Pre Bill Days | | | Number of days used in conjunction with Pre Bill Method to determine how far in advance to bill for services. A value of '0' will invoice up to, and including, either the Invoice Due Date (including any Grace Period) or Invoice Send Date as appropriate. |
| charge_days | Charge Delay | | | The number of days the system should wait after generating an invoice before charging any credit cards. |
| datesend | Invoice Send Date | | system_default | Day of the month client will be invoiced (monthly invoicing only). |
| datedue | Invoice Due Date | | | Day of the month on which the invoice is due.<br>If datedue is less than or equal to datesend, the invoice will be due on that day of the following month. |
| datepay | Grace Period | | | Number of days after which the invoice will be due. |
| referred_by | Referred By | | | |
| discount | Discount Level | | 0 | The default discount level for the client, specified either as a dollar amount or a percentage (the default). |
| discount_type | Discount Type | | 0 | The method for calculating the discount:<br>0 = percentage<br>1 = dollar value |
| referred | Referred By | | | This field stores whoever may have referred the client. |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| active | Client Status | | 1 | 1 = Client, 2 = Lead |
| late_fee_scheme_id | Late Fee Schedule | | | Late fee schedule |
| default_renew | Default Renew Date | | | Default renew date for new services |
| prorate_min_days | Prorate Min. Days | | 0 | Minimum number of proratable days (inclusive) |
| brand_id | Brand ID | | | Active Brand |
| retry_every | Retry Interval | | | When a client's credit card is declined, the charge is stored in a queue and is retried at a later date. The number of days between retries is defined by this setting. |
| priority | Ticket Priority | | | This variable takes the integer that corresponds to the desired level of priority given to a client's support tickets.<br>0 = Low<br>1 = Normal<br>2 = High<br>3 = 911 |
| access | Access Settings | | | The access settings for the client. This is an array containing the permissions for the client. |
| strict | Strict Mode | | 0 | This field is used to tell the API script whether or not the new client's Ubersmith login name must be unique.<br>If strict is set to 0 and the specified login name is already in use the client will be added to the system and the API script will return a note saying that their information must be updated inside the system.<br>If strict is set to 1, any duplicate login name will cause the script to return a failure as well as a note explaining why. |
| meta_* | Custom Data | | | Metadata items may be set by passing the values as additional parameters matching the metadata items API field names prefixed with 'meta_'. |
| auto_apply_credit | Auto Apply Credit | | | Automatically apply account credits |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device - Device

Functions related to the management of devices, monitors, and IP assignments

### device.add - Add a New Device

This method is used to create a new device.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| dev_desc | Device Description | Y | | |
| label | Device Label | | | |
| parent | Parent Device ID | | 0 | |
| type_id | Device Type ID | | 0 | |
| rack_id | Rack ID | | 0 | |
| rack_pos | Position Within Rack | | | |
| height | Device Height | | 1 | |
| client_id | Client ID | | | |
| service_id | Service ID | | | |
| client_owned | Client Owns Physical Device | | 0 | 0 = Client does not own physical device, 1 = Client owns physical device |
| connection_node_type_id | Connection Node Type ID | | | |
| front_image_file_id | Front Image File ID | | | A file id to use for the device's front image |
| back_image_file_id | Back Image File ID | | | A file id to use for the device's back image |
| front_image | Attachment | | | An image file for the device's front image. When using cURL the value for the parameter should be @ where is the image you wish to attach. |
| back_image | Attachment | | | An image file for the device's back image. When using cURL the value for the parameter should be @ where is the image you wish to attach. |
| meta_* | Custom Data | | | Metadata items may be set by passing the values as additional parameters matching the metadata items API field names prefixed with 'meta_'. |

**Example Output**

```
{
  "data": "83",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.comment_list - List a Device's Comments

This method is used to list a device's comments.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| device_id | Device ID | Y | | |
| filter_text | Filter Text | | | An array of strings to match against comments and attachment filenames. Comments which match all terms are returned. |
| client_viewable | Client Viewable | | | If client_viewable needs to be filtered (1 to only show client_viewable) |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "19": {
      "edited": "1312260059",
      "comment": "this is a device comment",
      "editor": "",
      "time": "1312260059",
      "clientid": "0",
      "user": "Administrator",
      "comment_id": "19",
      "client_viewable": "0",
      "dev": "23"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.connection_add - Add a Device Connection

This method is used to add a new Connection between Devices.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| name | Name | Y | | |
| links | Links | Y | | Array of links |
| connection_type_id | Connection Type ID | | | |
| status | Status | | | A single status ID. Options:<br>1 = Active<br>2 = Testing<br>4 = Failed<br>8 = New<br>16 = Inactive |

**Example Output**

```
{
  "data": 4,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.connection_delete - Delete a Device Connection

This method is used to remove Connections between Devices.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| connection_id | Connection ID | | | One or more (comma-separated) Connection ID numbers. |
| device_id | Device ID | | | Delete all connections on the given Device ID. Ignored if a Connection ID is supplied. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.connection_link_add - Add a Connection Link

This method is used to add a new Link within a Connection.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| connection_id | Connection ID | Y | | |
| name | Name | Y | | |
| segments | Segments | Y | | Array of segments |

**Example Output**

```
{
  "data": 4,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.connection_link_list - List Connection Links

This method is used to list Connection Links.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| connection_id | Connection ID | | | |
| device_id | Device ID | | | |
| status | Status | | | 1 = Active, 2 = Testing, 4 = Failed, 0 = Inactive |

**Example Output**

```
{
  "data": {
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.connection_link_update - Update a Connection Link

This method is used to update a Link within a Connection.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| connection_link_id | Connection Link ID | Y | | |
| name | Name | | | |
| segments | Segments | | | Array of segments |

**Example Output**

```
{
  "data": 4,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.connection_list - List Device Connections

This method is used to list a device's Connections.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| device_id | Device ID | | | |
| status | Status | | | A single status ID. Options:<br>1 = Active<br>2 = Testing<br>4 = Failed<br>8 = New<br>16 = Inactive |
| status_bit | Status | | | Multiple status IDs, using bitwise logic. Options:<br>1 = Active<br>2 = Testing<br>4 = Failed<br>8 = New<br>16 = Inactive<br>255 = All<br>15 = Not Inactive |

**Example Output**

```
{
  "data": {
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.connection_node_type_list - List Connection Node Types

This method is used to list available connection node types.

**Example Output**

```
{
  "data": {
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.connection_type_list - List Connection Types

This method is used to list available connection types.

**Example Output**

```
{
  "data": {
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.connection_update - Update a Device Connection

This method is used to update a Connection between Devices.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| connection_id | Connection ID | Y | | |
| name | Name | | | |
| status | Status | | | A single status ID. Options:<br>1 = Active<br>2 = Testing<br>4 = Failed<br>8 = New<br>16 = Inactive |

**Example Output**

```
{
  "data": 4,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.cpanel_add - Add a cPanel Account

This method is used to add a cPanel account.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| service_id | Service ID | Y | | |
| create | Create Flag | | 1 | Set to 0 if account already exists on the cPanel server. This will only create the association within Ubersmith to manage the account. |
| cpanel_plan | cPanel Plan Code | | from service plan | |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.delete - Delete a Device

This method is used to delete a device.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| device_id | Device ID | Y | | |
| force_delete_connections | Allow deleting a device that has connections | | | By default, deleting a device is blocked if it has connections. Setting this to 1 automatically deletes connections this device has before deleting the device. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.dns_record_add - Add a DNS Record

This method is used to add a new Record to a DNS Zone.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| zone_id | Zone ID | Y | | Id of specific dns zone. |
| type | Record Type | Y | | Record Type |
| name | Record Name | Y | | Record Name |
| content | Record Content | Y | | Record Content |
| ttl | TTL | | | The TTL of the record. |
| prio | Priority | | | The priority of the record. |

**Example Output**

```
{
  "data": 4,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.dns_record_delete - Delete a DNS Record

This method is used to delete a DNS Record.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| record_id | Record ID | Y | | DNS Record ID. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.dns_record_list - List DNS Records

This method is used to list DNS Recrods.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| record_id | Record ID | | | ID of specific dns record. |
| zone_id | Zone ID | | | ID of specific dns zone. |
| zone_name | Zone Name | | | Name of specific dns zone. |
| type | Record Type | | | Record Type |
| name | Record Name | | | Record Name |
| content | Record Content | | | Record Content |
| client_id | Client ID | | | Client ID |
| service_id | Service ID | | | Service ID |
| exclude_special | Exclude Special Records | | 0 | Set to 1 to exclude SOA, MX & NS records |
| order_by | Order By | | name | Field to order results by, available fields: record_id, name |
| direction | Direction | | asc | asc = Ascending<br>desc = Descending |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | 0 | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "109": {
      "name": "www.example.com",
      "zone_id": "109",
      "record_id": "1263",
      "content": "127.0.0.1",
      "type": "A",
      "ttl": "3600",
      "prio": "0",
      "change_date": "1359455515",
      "zone_name": "example.com",
      "client_id": "1234",
      "service_id": "0"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.dns_record_update - Update a DNS Record

This method is used to update a DNS Recrord.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| record_id | Record ID | | | Id of specific dns record. |
| type | Record Type | | | Record Type |
| name | Record Name | | | Record Name |
| content | Record Content | | | Record Content |
| ttl | TTL | | | The TTL of the record. |
| prio | Priority | | | The priority of the record. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.dns_zone_add - Add a DNS Zone

This method is used to add a new DNS Zone

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| client_id | Client ID | | | Optional field to associate the zone with a specific client. |
| service_id | Service ID | | | Optional field to associate the zone with a specific service. |
| name | Domain Name | Y | | The name of the DNS Zone. |
| hostmaster | Hostmaster | | | The Hostmaster for the Zone. |
| refresh | Refresh Interval | | | The Refresh time in seconds for the zone. |
| retry | Retry Interval | | | The Retry time in seconds for the zone. |
| expire | Expire Interval | | | The Expire time in seconds for the zone. |
| default_ttl | Default TTL | | | The Default TTL of the Zone. |
| ns1 | Primary Nameserver | | | The Primary Nameserver address. |
| ns2 | Secondary Nameserver | | | The Secondary Nameserver address. |

**Example Output**

```
{
  "data": 4,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.dns_zone_delete - Delete a DNS Zone

This method is used to delete a DNS Zone.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| zone_id | Zone ID | Y | | DNS Zone ID. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.dns_zone_list - List DNS Zones

This method is used to list DNS Zones.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| zone_id | Zone ID | | | DNS Zone ID |
| name | Zone Name | | | DNS Zone Name |
| client_id | Client ID | | | Client ID |
| service_id | Service ID | | | Service ID |
| order_by | Order By | | name | Field to order results by, available fields: zone_id, name |
| direction | Direction | | asc | asc = Ascending<br>desc = Descending |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | 0 | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "109": {
      "zone_id": "109",
      "name": "example.com",
      "client_id": "1263",
      "service_id": "0",
      "soa": {
          "primary" : "example.com",
          "hostmaster": "hostmaster.example.com",
          "serial": "2013041101",
          "refresh": "10800",
          "retry": "3600",
          "expire": "604800",
          "default_ttl": "3600",
          }
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.dns_zone_update - Update a DNS Zone

This method is used to update a DNS Zone.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| zone_id | Zone ID | Y | | DNS Zone ID. |
| client_id | Client ID | | | Optional field to associate the domain with a specific client. |
| service_id | Service ID | | | Optional field to associate domain with a specific service. |
| name | Domain Name | | | The name of the DNS Zone. |
| hostmaster | Hostmaster | | | The Hostmaster for the Zone. |
| refresh | Refresh Interval | | | The Refresh time in seconds for the zone. |
| retry | Retry Interval | | | The Retry time in seconds for the zone. |
| expire | Expire Interval | | | The Expire time in seconds for the zone. |
| default_ttl | Default TTL | | | The Default TTL in seconds for the Zone. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.event_list - List Device Events

This method is used to list device events.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| device_id | Device ID | Y | | |
| begin | Start Date | | | |
| end | End Date | | | |
| eventuser | User | | | |
| searchfor | Search String | | | |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "1494": {
      "device": "83",
      "device_event_id": "1494",
      "after": {
        "row_id": "0",
        "packid": null,
        "devtype_group_id": "0",
        "label": "",
        "remote_reboot_port_2": "",
        "switch_port_eth0": "",
        "rack_pos": "",
        "parent": "0",
        "remote_reboot_port_3": "",
        "power_circuit_phase": "",
        "power_panel_voltage": "",
        "switch_port_eth1": "",
        "power_circuit_status": "1",
        "power_circuit_num": "",
```

```
            "switch_port_eth2": "",
            "power_status": null,
            "root_password": "",
            "power_panel_amp_rating": "",
            "ubersmith_version": "",
            "mysql_password": "",
            "rack_position": "",
            "down": "0",
            "total": "0",
            "last": null,
            "type_id": "0",
            "power_panel_fed_from": "",
            "zone_id": "0",
            "up": "0",
            "rack_id": "0",
            "power_panel_wire": "",
            "remote_reboot_unit_2": "",
            "power_circuit_breaker": "",
            "parent_desc": null,
            "type": null,
            "power_panel_main_breaker": "",
            "remote_reboot_unit_3": "",
            "remote_reboot_unit": "",
            "phone": null,
            "first": null,
            "listed_company": null,
            "height": "1",
            "power_panel_phase": "",
            "power_circuit_max_load": "",
            "remote_reboot_port": "",
            "distribution": "debian6",
            "power_panel_type": "house",
            "cage_id": "0",
            "disabled": "0",
            "desserv": null,
            "clientid": null,
            "fac_id": "0",
            "devtype_group_name": "Ungrouped",
            "dev_desc": "test device",
            "some_parameter": "",
            "rack_name": "",
            "warn": "0",
            "email": null,
            "active": "1",
            "dev": "83"
        },
        "time": "1312259948",
        "action": "Device 83 added",
        "reference_id": "83",
        "reference_type": "device",
        "before": null,
        "event_type": "1",
        "user": "admin",
        "client_viewable": "1",
        "ip_address": "192.168.0.134"
      }
    },
    "error_message": "",
    "error_code": null,
    "status": true
}
```

## device.facility_list - List Device Facilities

This method is used to list device facilities.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| fac_id | Facility ID | | | |
| fac_code | Facility Code | | | |
| zone_id | Zone ID | | | |
| zone_code | Zone Code | | | |
| cage_id | Cage ID | | | |
| cage_code | Cage Code | | | |
| row_id | Row ID | | | |
| row_code | Row Code | | | |
| rack_id | Rack ID | | | |
| rack_code | Rack Code | | | |
| client_id | Client ID | | | |
| metadata | Location Metadata | | | |
| exclude_downstream | Exclude downstream zones, cages, rows, and racks | | | |

**Example Output**

```
{
  "data": {
    "1": {
      "facility": {
        "fac_country": "US",
        "fac_city": "Troy",
        "fac_name": "Equinix Troy",
        "foobar": null,
        "fac_code": "eqxtry",
        "fac_zip": "12018",
        "fac_state": "NY",
        "fac_address": "353 Broadway",
        "fac_description": "",
        "fac_id": "1"
      },
      "id": "1",
      "zones": {
        "1": {
          "cages": {
            "1": {
              "cage_name": "default Cage",
              "zone_id": "1",
              "rows": {
                "1": {
                  "row_id": "1",
                  "row_name": "Default Row",
                  "racks": {
                    "1": {
                      "row_id": "1",
                      "rack_code": "dlfra",
                      "rack_slots": "0",
                      "rack_direction": "bottom",
                      "rack_shelves": "0",
                      "rack_type": "standard",
                      "rack_id": "1",
                      "rack_u": "42",
                      "rack_name": "Default Rack"
                    }
                  },
                  "row_code": "dflr",
                  "cage_id": "1"
                }
              },
```

```
            "cage_code": "dflc",
            "cage_id": "1"
          }
        },
        "zone_id": "1",
        "zone_code": "dflz",
        "fac_id": "1",
        "zone_name": "Default Zone"
      }
    }
  }
},
"error_message": "",
"error_code": null,
"status": true
}
```

## device.get - Get a Device

This method is used to get a device's details.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| device_id | Device ID | Y | | |
| metadata | Device Metadata | | | |
| service | Service ID | | | |
| modules | Device Modules | | 0 | 1 = modules & module_details, 2 = module_details. If a device module encounters an error, an array containing uber_error_code and uber_error_message keys is returned. |
| tags | Device Tags | | 0 | 1 = include device tags sub-array |
| locations | Device Locations | | 1 | 0 = do not include rack/row/cage/zone/facility sub-arrays |

**Example Output**

```
{
  "data": {
    "row_id": 0,
    "packid": null,
    "devtype_group_id": "1",
    "label": "",
    "location": "",
    "rack_pos": "",
    "parent": "0",
    "zone": {
      "code": 1,
      "userinfo": null,
      "mode": 1,
      "error_message_prefix": "",
      "level": 1024,
      "callback": null,
      "message": "No zone specified",
      "backtrace": null
    },
    "power_status": null,
    "row": {
      "code": 1,
      "userinfo": null,
      "mode": 1,
      "error_message_prefix": "",
      "level": 1024,
      "callback": null,
      "message": "No row specified",
      "backtrace": null
    },
    "rack_position": "",
    "down": "0",
    "total": "1",
    "last": null,
    "type_id": "1",
    "facility": {
      "code": 1,
      "userinfo": null,
      "mode": 1,
      "error_message_prefix": "",
```

```
            "level": 1024,
            "callback": null,
            "message": "No facility specified",
            "backtrace": null
        },
        "zone_id": 0,
        "rack": {
            "code": 1,
            "userinfo": null,
            "mode": 1,
            "error_message_prefix": "",
            "level": 1024,
            "callback": null,
            "message": "No rack specified",
            "backtrace": null
        },
        "up": "1",
        "rack_id": "0",
        "cage": {
            "code": 1,
            "userinfo": null,
            "mode": 1,
            "error_message_prefix": "",
            "level": 1024,
            "callback": null,
            "message": "No cage specified",
            "backtrace": null
        },
        "parent_desc": null,
        "type": "Generic Server",
        "phone": null,
        "first": null,
        "listed_company": null,
        "rack_units": "1",
        "height": "45.000000",
        "width": "482.60000",
        "depth": "609.60000",
        "front_image_file_id":"19",
        "back_image_file_id":"0"
        "cage_id": 0,
        "assignments": [

        ],
        "disabled": "0",
        "desserv": null,
        "clientid": null,
        "fac_id": 0,
        "height_display_value": 1.75,
        "height_display_unit": "inches",
        "width_display_value": 19,
        "width_display_unit": "inches",
        "depth_display_value": 2,
        "depth_display_unit": "feet",
        "front_image_url":"\/admin\/file\/19",
        "back_image_url":"",
        "devtype_group_name": "Server",
        "dev_desc": "Server",
        "rack_name": "",
        "warn": "0",
        "email": null,
        "owner_type": null,
        "owner": "0",
        "active": "1",
        "dev": "80"
    },
    "error_message": "",
    "error_code": null,
    "status": true
}
```

## device.hostname_get - Get a Device Hostname

This method is used to get control panel hostname information.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| plan_id | Service Plan ID | Y | | |

**Example Output**

```
{
  "data": {
    "name": "host.example.com",
    "username": "root",
    "accesskey": "",
    "id": "1",
    "type": "whm",
    "host": "host.example.com",
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.interface_add - Add a Device Interface

This method is used to add a new Interface to a Device.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| device_id | Device ID | Y | | |
| name | Interface Name | Y | | |
| interface_type_id | Interface Type ID | | | |
| mac | MAC Address | | | |
| status | Status | | 1 | 1 = Active, 2 = Testing, 4 = Failed, 0 = Inactive, 8 = Deactivated, 16 = Internal |
| description | Description | | | |

**Example Output**

```
{
  "data": 4,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.interface_list - List Device Interfaces

This method is used to list a device's Interfaces.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| device_id | Device ID | | | |
| mac | MAC Address | | | |
| name | Name | | | |
| status | Status | | | 1 = Active, 2 = Testing, 4 = Failed, 0 = Inactive, 8 = Deactivated, 16 = Internal |

**Example Output**

```
{
  "data": {
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.interface_type_list - List Device Interface Types

This method is used to list available device interface types.

**Example Output**

```
{
  "data": {
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.interface_update - Update a Device Interface

This method is used to update an Interface on a Device.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| device_interface_id | Device Interface ID | Y | | |
| name | Interface Name | | | |
| interface_type_id | Interface Type ID | | | |
| mac | MAC Address | | | |
| status | Status | | | 1 = Active, 2 = Testing, 4 = Failed, 0 = Inactive, 8 = Deactivated, 16 = Internal |
| description | Description | | | |

**Example Output**

```
{
  "data": 4,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.ip_assign - Assign an IP to a Device

This method is used to assign an IP address to a device.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| device_id | Device ID | Y | | |
| addr | IP Assignment | | | |
| cidr | CIDR Assignment Size | | 32 | |
| vlan_num | VLAN Number | | | |
| group_id | IP Group ID | | | |
| client_id | Client ID | | | |
| force | Force Flag | | | |

**Example Output**

```
{
  "data": {
    "assign_id": "159",
    "addr_readable": "10.0.0.33",
    "vlan_num": null,
    "created_ts": "1295376042",
    "assign_description": "Description",
    "vlan_range_id": null,
    "updated_ts": "1312261879",
    "service_id": "419",
    "group_id": "0",
    "vlan_type_id": null,
    "group_client_id": null,
    "group_description": null,
    "client_id": "1010",
    "device_id": "25",
    "servtype": "cpanel",
    "listed_company": "Ubersmith",
    "last_ip": "00000000000000000000000000a000021",
    "portable": "0",
    "desserv": "",
    "addr_type_readable": "Global",
    "dev_desc": "ubersmith.com",
    "num_ips": "1",
    "addr_type": "1",
    "addr": "00000000000000000000000000a000021/128"
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.ip_assignment_add - Create a New IP Assignment

This method is used to create a new IP address assignment.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| addr | Address(es) to Assign | | | String or Array, CIDR or range notation. |
| cidr | CIDR Assignment Size | | | Automatically create a new assignment of this size, only used if addr is not specified. |
| device_id | Device ID | | | |
| client_id | Client ID | | | |
| service_id | Service ID | | | |
| group_id | Group ID | | | |
| fac_id | Facility ID | | | |
| pool_id | Pool ID | | | |
| block_id | Block ID | | | |
| assign_description | Description | | | |
| addr_type | Address Type | | 1 | 1 = Global<br>2 = Local<br>4 = Gateway |
| portable | Portable | | 0 | 0 = Not Portable<br>1 = Portable |

**Example Output**

```
{
  "data": {
    "errors": [

    ],
    "assignments": {
      "160": {
        "gateway_readable": "10.0.0.36",
        "assign_id": "160",
        "addr_readable": "10.0.0.36",
        "vlan_num": null,
        "created_ts": "1312262119",
        "network_readable": "10.0.0.36",
        "assign_description": "10.0.0.36",
        "broadcast_readable": "10.0.0.36",
        "vlan_range_id": null,
        "updated_ts": "1312262119",
        "service_id": "354",
        "group_id": "0",
        "vlan_type_id": null,
        "group_client_id": null,
        "group_description": null,
        "client_id": "1010",
        "device_id": "25",
        "servtype": "123456",
        "listed_company": "Ubersmith",
        "last_ip": "00000000000000000000000000a000024",
        "portable": "0",
        "desserv": "Bargain Dedicated Server",
        "addr_type_readable": "Global",
        "dev_desc": "ubersmith.com",
        "num_ips": "1",
        "addr_type": "1",
        "addr": "00000000000000000000000000a000024/128"
      }
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.ip_assignment_delete - Delete a Device IP Assignment

This method is used to delete an IP assignment from a device.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| assign_id | IP Assignment ID | Y | | |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.ip_assignment_list - List Device IP Assignments

This method is used to list a device's IP assignments.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| device_id | Device ID | | | |
| client_id | Client ID | | | |
| group_id | IP Group ID | | | |
| service_id | Service ID | | | |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "160": {
      "assign_id": "160",
      "addr_readable": "10.0.0.36",
      "vlan_num": null,
      "created_ts": "1312262119",
      "assign_description": "10.0.0.36",
      "vlan_range_id": null,
      "updated_ts": "1312262119",
      "service_id": "354",
      "group_id": "0",
      "vlan_type_id": null,
      "group_client_id": null,
      "group_description": null,
      "client_id": "1010",
      "device_id": "25",
      "servtype": "123456",
      "listed_company": "Ubersmith",
      "last_ip": "000000000000000000000000000a000024",
      "portable": "0",
      "desserv": "Bargain Dedicated Server",
      "addr_type_readable": "Global",
      "dev_desc": "ubersmith.com",
      "label": "ip address",
      "num_ips": "1",
      "addr_type": "1",
      "addr": "000000000000000000000000000a000024/128"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.ip_assignment_update - Update a Device IP Assignment

This method is used to update a device's IP assignment.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| assign_id | IP Assignment ID | Y | | |
| addr | IP Address | | | CIDR notation. |
| assign_description | Description | | | |
| device_id | Device ID | | | |
| client_id | Client ID | | | |
| service_id | Service ID | | | |
| group_id | Group ID | | | |
| addr_type | Address Type | | 1 | 1 = Global<br>2 = Local<br>4 = Gateway |
| portable | Portable | | 0 | 0 = Not Portable<br>1 = Portable |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.ip_block_list - List IP Blocks

This method is used to list IP blocks.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| block_id | Block ID | | | |
| pool_id | Pool ID | | | |
| fac_id | Facility ID | | | |
| block_assign | Auto-assignment | | | Return only IP blocks with the auto-assignment set to the specific flag: 1 = Yes, 0 = No |
| ipver | IP Version | | | Possible versions:<br>4<br>6 |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "1": {
      "block_id": "9",
      "pool_id": "2",
      "addr": "00000000000000000000000ffffffff/128",
      "priority": "1",
      "block_assign": "1",
      "created_ts": "1353361624",
      "updated_ts": "1353361624",
      "block_description": "brief description",
      "addr_readable": "255.255.255.255"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
```

}

## device.ip_get_available - List Available IP Addresses

This method is used to list existing IP Assignments which are not currently associated with a device.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| device_id | Device ID | | | If specified, use the client associated with this device |
| client_id | Client ID | | | If specified, use this client |
| cidr | CIDR Assignment Size | | 32 | |
| vlan_num | VLAN Number | | | |
| group_id | IP Group ID | | | |

**Example Output**

```
{
  "data": {
    "33": {
      "gateway_readable": "172.0.0.161",
      "addr_readable": "172.0.0.169",
      "network_readable": "172.0.0.160/28",
      "broadcast_readable": "172.0.0.175",
      "hsrp1_readable": "172.0.0.162",
      "hsrp2_readable": "172.0.0.163"
    },
    "22": {
      "gateway_readable": "11.11.11.1",
      "addr_readable": "11.11.11.60",
      "network_readable": "11.11.11.0/29",
      "broadcast_readable": "11.11.11.7",
      "hsrp1_readable": "11.11.11.2",
      "hsrp2_readable": "11.11.11.3"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.ip_get_unassigned - Get Unassigned IP Addresses

This method is used to get a list of unassigned IP Addresses.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| fac_id | Facility ID | | | |
| pool_id | IP Address Pool ID | | | |
| block_id | IP Address Block ID | | | |
| cidr | CIDR Assignment Size | | 32 | |
| portable | Portable | | 0 | 0 = Not Portable<br>1 = Portable |
| limit | Limit | | 50 | |

**Example Output**

```
{
  "data": {
    "44": "10.0.0.63",
    "33": "10.0.0.52",
    "5": "10.0.0.6"
  },
  "error_message": "",
  "error_code": null,
```

```
    "status": true
}
```

## device.ip_group_add - Add a Device IP Group

This method is used to add an IP group to a device.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| group_description | Group Description | Y | | |
| client_id | Client ID | | | 0 = Global |
| vlan_range_id | VLAN Range ID | | | Required if vlan_num is specified. |
| vlan_num | VLAN Number | | | Will be auto-assigned if not specified and vlan_range_id is non-zero. |
| hsrp | HSRP | | | Enable or disable HSRP on the group, disabled by default (1 = Enable, 0 = Disable) |

**Example Output**

```
{
  "data": {
    "vlan_num": 0,
    "group_id": "16",
    "vlan_range_id": 0,
    "group_description": "this is the group description",
    "client_id": 0
    "hsrp": 0
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.ip_group_delete - Delete a Device IP Group

This method is used to delete an IP group from a device.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| group_id | Group ID | Y | | |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.ip_group_list - List a Device IP Group

This method is used to add an list a device's IP groups.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| device_id | Device ID | | | |
| client_id | Client ID | | | |
| fac_id | Facility ID | | | |
| vlan_range_id | VLAN Range ID | | | |
| vlan_num | VLAN Number | | | |
| is_vlan | VLAN Flag | | | |
| is_hsrp | HSRP Flag | | | |
| empty | Empty VLANs | | | If true and is_vlan, return all "empty" VLANs |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "11": {
      "vlan_num": "9000",
      "vlan_range_id": "4",
      "group_id": "11",
      "vlan_type_id": "2",
      "group_description": "description",
      "client_id": "1010",
      "listed_company": "Ubersmith",
      "assignment_count": "0"
      "hsrp": "0"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.ip_group_update - Update a Device IP Group

This method is used to update a device's IP group.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| group_id | Group ID | Y | | |
| group_description | Group Description | | | |
| client_id | Client ID | | | 0 = Global |
| vlan_range_id | VLAN Range ID | | | Required if vlan_num is specified. |
| vlan_num | VLAN Number | | | Will be auto-assigned if not specified and vlan_range_id is non-zero. |
| hsrp | HSRP | | | Enable or disable HSRP on the group (1 = Enable, 0 = Disable) |

**Example Output**

```
{
  "data": {
    "vlan_num": 0,
    "vlan_range_id": "0",
    "group_id": "17",
    "group_description": "new desc",
    "client_id": "0"
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.ip_lookup - Look Up a Device IP

This method is used to look up the details of the assignment which contains the specified IP address.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| ip | IP Address | Y | | Required if vlan_num is specified. |
| client_id | Client ID | | | Required if addr_type is 'local' or 'all' |
| addr_type | IP Assignment Type | | global | Options are 'local', 'global', or 'all' |
| metadata | Metadata Flag | | 0 | Will be auto-assigned if not specified and vlan_range_id is non-zero. |

**Example Output**

```
{
  "data": {
    "assign_id": "111",
    "addr_readable": "11.11.11.4",
    "created_ts": "1280412247",
    "assign_description": "11.11.11.0/29",
    "updated_ts": "1312261651",
    "service_id": "259",
    "group_id": "0",
    "client": {
      "city": "Troy",
      "address": "353 Broadway",
      "company": "Ubersmith",
      "zip": "12018",
      "country": "US",
      "class_id": "1",
      "last": "Burns",
      "full_name": "Johnny Ubersmith",
      "listed_company": "Ubersmith",
      "fax": "+1.2043245672",
      "phone": "+1.2043245671",
      "first": "Johnny",
      "clientid": "1010",
      "active": "1",
      "balance": "187554.67",
```

```
      "email": "support@ubersmith.com",
      "state": "NY"
    },
    "client_id": "1010",
    "device_id": "25",
    "range": "11.11.11.4",
    "last_ip": "00000000000000000000000000b0b0b04",
    "portable": "0",
    "num_ips": "1",
    "addr_type": "1",
    "addr": "00000000000000000000000000b0b0b04/128"
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.ip_pool_list - List IP Pools

This method is used to list IP pools.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| pool_id | Pool ID | | | |
| fac_id | Facility ID | | | |
| pool_assign | Auto-assignment | | | Return only IP pools with the auto-assignment set to the specific flag: 1 = Yes, 0 = No |
| show_blocks | Show IP blocks | | | If specified, the return will also include IP blocks associated with each IP pool |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "1": {
      "pool_id": "1",
      "fac_id": "3",
      "pool_name": "Pool",
      "pool_assign": "1",
      "pool_description": "Pool description",
      "notify_address": "someone@ubersmith.com",
      "min_subnet": "0",
      "min_subnet_size": "ip4_28",
      "curr_subnet": "1",
      "max_util": "1",
      "curr_util": "1",
      "status": "2",
      "blocks": {
        "1" {
          "block_id": "4",
          "pool_id": "2",
          "addr": "00000000000000000000000000ffffffff/128",
          "priority": "1",
          "block_assign": "1",
          "created_ts": "1353361603",
          "updated_ts": "1353361603",
          "block_description": "brief description",
          "addr_readable": "255.255.255.255"
        }
      }
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.ip_unassign - Unassign a Device IP

This method is used to unassign a device's IP.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| device_id | Device ID | Y | | |
| assign_id | IP Assignment ID | | | |
| addr | IP Address | | | |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.list - List Devices

This method is used to list devices in the system.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| parent | Parent Device ID | | | If specified, only devices which are children of the specified device are returned. If the special value 0 is specified, only top-level devices are returned. |
| client_id | Client ID | | | |
| service_id | Service ID | | | |
| device | Device ID(s) | | | A single device ID or an array of device IDs |
| status | Device Status | | | Possible values:<br>up<br>warn<br>down |
| label | Device Label | | | |
| dev_desc | Device Description | | | |
| devtype_group_id | Device Type Group ID | | | Return only devices in the specified device type group. If the value 0 is specified, devices of ungrouped types will be returned. |
| type_id | Device Type ID | | | Return only devices of the specified type. Set to 0 to return devices without a type set. |
| rack_id | Rack ID | | | Return only devices in the specified rack. Set to 0 to return devices without a rack. |
| row_id | Row ID | | | Return only devices in the specified row. Set to 0 to return devices without a row. |
| cage_id | Cage ID | | | Return only devices in the specified cage. Set to 0 to return devices without a cage. |
| zone_id | Zone ID | | | Return only devices in the specified zone. Set to 0 to return devices without a zone. |
| fac_id | Facility ID | | | Return only devices in the specified facility. Set to 0 to return devices without a facility. |
| tag_id | Tag ID | | | Return only devices with the specified tag. |
| tag_ids | Tag IDs | | | Return only devices with the specified tags. An array of tag IDs |
| require_ip | Include IP Addresses | | 0 | Set to 1 to include IP Address information in the output. |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| metadata | Include Metadata | | 0 | Set to 1 to include metadata in the output. |
| modules | Device Modules | | | 1 = modules & module_details, 2 = module_details |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "22": {
      "row_id": "1",
      "packid": "1096",
      "company": "Ubersmith",
      "label": "us.ubersmith.com",
      "devtype_group_id": "2",
      "location": "dlfra.dflr.dflc.dflz.eqxtry:39",
      "rack_code": "dlfra",
      "rack_pos": "39",
      "parent": "0",
      "fac_name": "Troy",
      "power_status": null,
      "down": "3",
      "total": "5",
      "fac_code": "eqxtry",
      "last": "Uber",
      "type_id": "12",
      "up": "1",
      "zone_id": "1",
      "cage_name": "default Cage",
      "row_name": "Default Row",
      "rack_id": "1",
      "type": "Virtual IP",
      "cage_code": "dflc",
      "row_code": "dflr",
      "phone": "+1.2125555555",
      "first": "Johnny",
      "rack_units" : "1",
      "height": "45.000000",
      "width": "482.60000",
      "depth": "609.60000",
      "front_image_file_id": "19",
      "back_image_file_id": "0"
      "height_display_value": 1.75,
      "height_display_unit": "inches",
      "width_display_value": 19,
      "width_display_unit": "inches",
      "depth_display_value": 2,
      "depth_display_unit": "feet",
      "disabled": "1",
      "cage_id": "1",
      "desserv": "server.ubersmith.com",
      "clientid": "1010",
      "devtype_group": "Network",
      "fac_id": "1",
      "zone_code": "dflz",
      "dev_desc": "us.ubersmith.com",
      "warn": "0",
      "zone_name": "Default Zone",
      "rack_name": "Default Rack",
      "email": "support@ubersmith.com",
      "owner_type": "PACKAGE",
      "owner": "1096",
      "active": "1",
      "dev": "22"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.module_call - Call a Device Module Function

This method is used to execute a device module function.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| device_id | Device ID | Y | | |
| module_id | Module Name | Y | | |
| function | Function Name | Y | | |
| module_params | Parameters | | | Array |

**Example Output**

```
{
  "data": {
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.module_call_aggregate - Call an Aggregate Device Module Function

This method is used to execute an aggregate device module function.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| device_ids | Device ID Array | Y | | Array of Device IDs |
| module_id | Device Module Name | Y | | |
| function | Function Name | Y | | |
| module_params | Parameters | | | Array |

**Example Output**

```
{
  "data": {
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

### device.module_graph - Generate Device Module Graph

This method is used to generate a device module graph.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| device_id | Device ID | Y | | |
| module_id | Device Module Name | Y | | The device module identifier, it can either be the module ID or the module name. |
| module_params | Module Parameters | | | Array of module parameters (e.g. module_params[start]=1368721108). |

**Example Output**

```
This method returns PNG image data.
```

### device.monitor_add - Add a New Device Monitor

This method is used to add a new device monitor.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| device_id | Device ID | Y | | |
| protocol | Protocol | Y | | icmp = Ping<br>dns = DNS<br>tcp = TCP<br>ssl = SSL<br>module = Device Module (DE-only, see extra) |
| address | Monitored IP Address | | | |
| label | Monitor Label | | | Additional textual identifier for the monitor |
| notify | Notification Email Address | | | |
| notify_delay | Notification Delay | | 0 | Delay before sending notification in seconds<br>0 = Send notification immediately |
| notify_interval | Notification Interval | | 0 | Interval between notifications in seconds<br>0 = Do not re-notify |
| script_id | Script ID | | | Script to run for TCP & SSL monitors |
| port | Port | | | Port for TCP & SSL Monitors |
| loss | Packet Loss | | | Maximum Allowable Packet Loss % |
| extra | Extra Parameters | | | Array of extra Parameters (see extra[]) |
| extra[notify_downstream] | Notify Downstream Device Owners | | 0 | 0 = No<br>1= Yes |
| extra[module] | Module Name | | | Device Module Name |
| extra[] | Additional Parameters | | | Additional Parameters to store in the extra array, e.g. parameters required by device module monitors |

**Example Output**

```
{
  "data": "34",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.monitor_delete - Delete a Device Monitor

This method is used to delete a device monitor.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| mon_id | Monitor ID | | | |
| device_id | Device ID | | | |
| protocol | Protocol | | | icmp = Ping<br>dns = DNS<br>tcp = TCP<br>ssl = SSL<br>module = Device Module (DE-only, see extra) |
| script_id | Script ID | | | Script to run for TCP & SSL monitors |
| module | Module Name | | | Device Module Name |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.monitor_disable - Disable a Device Monitor

This method is used to disable a device monitor.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| mon_id | Monitor ID | | | |
| device_id | Device ID | | | |
| script_id | Script ID | | | |
| module | Module Name | | | |
| zone_id | Monitor ID | | | |
| devtype_group_id | Device Type Group ID | | | |
| client_id | Client ID | | | |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.monitor_enable - Enable a Device Monitor

This method is used to enable a device monitor.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| mon_id | Monitor ID | | | |
| device_id | Device ID | | | |
| script_id | Script ID | | | |
| module | Module Name | | | |
| zone_id | Monitor ID | | | |
| devtype_group_id | Device Type Group ID | | | |
| client_id | Client ID | | | |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.monitor_get - Get a Device Monitor

This method is used to get a device monitor's details.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| mon_id | Monitor ID | Y | | |
| logs | Monitor Logs | | 0 | |
| order_by | Order By | | | Field to order monitor log results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return monitor log records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of monitor log rows to return |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "mon_id": "3",
    "dev": "1",
    "address": "192.168.0.1",
    "addr_id": "0",
    "label": "Login page",
    "protocol": "tcp",
    "port": "0",
    "notify": "",
    "notify_delay": "600",
    "notify_interval": "0",
    "notify_up": "0",
    "script_id": "2",
    "state": "0",
    "reenable_time": null,
    "loss": "75",
    "extra": {
      "module": "",
      "notify_downstream": "0",
      "instructions": ""
    },
    "disabled_by": "0",
    "last_change": "1368631817",
```

```
      "last_poll": "1368808806",
      "last_notified": "0",
      "name": "HTTP test",
      "log": {
        "4": {
          "entry_id": "4",
          "time": "1368631817",
          "dev_id": "1",
          "mon_id": "3",
          "state": "down",
          "untiltime": "1369236902",
          "msg": "HTTP test on address 192.168.0.1, port 0.\nCouldn't open socket: Failed to parse address \"192.168.0.1\" (0)"
        }
      }
    }
  }
}
```

## device.monitor_list - List Device Monitors

This method is used to list device monitors.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| device_id | Device ID | | | |
| mon_id | Monitor ID | | | |
| state | Monitor State | | | |
| not_state | Monitor State | | | |
| script_id | Script ID | | | |
| module | Device Module | | | |
| protocol | Monitoring Protocol | | | icmp = Ping<br>dns = DNS<br>tcp = TCP<br>ssl = SSL<br>module = Device Module (DE-only, see extra) |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "3": {
      "mon_id": "3",
      "dev": "2",
      "address": "10.0.0.1",
      "addr_id": "0",
      "protocol": "ICMP (Ping)",
      "port": "",
      "label": "Login page",
      "notify": "support@ubersmith.com",
      "notify_delay": "300",
      "notify_interval": "3600",
      "notify_up": "3",
      "script_id": "0",
      "state": "0",
      "reenable_time": null,
      "loss": "75",
      "extra": {
        "module": "",
        "notify_downstream": "0",
        "instructions": ""
      },
      "disabled_by": "0",
      "last_change": "1389645613",
      "last_poll": "1389727802",
      "last_notified": "1389726003",
```

```
        "label": "",
        "name": ""
    },
    "4": {
        "mon_id": "4",
        "dev": "2",
        "address": "www.ubersmith.com",
        "addr_id": "0",
        "protocol": "TCP",
        "port": "80",
        "notify": "support@ubersmith.com",
        "notify_delay": "600",
        "notify_interval": "0",
        "notify_up": "0",
        "script_id": "2",
        "state": "1",
        "reenable_time": null,
        "loss": "75",
        "extra": {
          "module": "",
          "notify_downstream": "0",
          "instructions": ""
        },
        "disabled_by": "0",
        "last_change": "1383856464",
        "last_poll": "1389727802",
        "last_notified": "0",
        "label": "Login page",
        "name": "HTTP test"
      }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.monitor_update - Update a Device Monitor

This method is used to update a device monitor.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| mon_id | Monitor ID | Y | | |
| protocol | Protocol | Y | | icmp = Ping<br>dns = DNS<br>tcp = TCP<br>ssl = SSL<br>module = Device Module (DE-only, see extra) |
| address | Monitored IP Address | | | |
| label | Monitor Label | | | Additional textual identifier for the monitor |
| notify | Notification Email Address | | | |
| notify_delay | Notification Delay | | 0 | Delay before sending notification in seconds<br>0 = Send notification immediately |
| notify_interval | Notification Interval | | 0 | Interval between notifications in seconds<br>0 = Do not re-notify |
| script_id | Script ID | | | Script to run for TCP & SSL monitors |
| port | Port | | | Port for TCP & SSL Monitors |
| loss | Packet Loss | | | Maximum Allowable Packet Loss % |
| extra | Extra Parameters | | | Array of extra Parameters (see extra[]) |
| extra[notify_downstream] | Notify Downstream Device Owners | | 0 | 0 = No<br>1= Yes |
| extra[module] | Module Name | | | Device Module Name |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| extra[] | Additional Parameters | | | Additional Parameters to store in the extra array, e.g. parameters required by device module monitors |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.raw_module_list - List Available Device Modules

This method returns a list of device modules.

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "dm_bandwidth": {
      "module": "dm_bandwidth",
      "title": "Bandwidth Monitoring"
    },
    "dm_environment": {
      "module": "dm_environment",
      "title": "Environmental Monitoring"
    },
  }
}
```

## device.reboot - Set a Device's Power State

This method is used to control the power state of a device.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| device_id | Device ID | Y | | |
| power_action | Power Action | Y | | Possible values:<br>on<br>off<br>reboot |
| outlet | Outlet Number | | | Control an individual port |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.reboot_graph - Get a Reboot Graph

This method is used to generate a graph of the power usage for a specified device.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| device_id | Device ID | Y | | |
| module_id | Device Module Name | Y | | |
| start_time | Start Date | | | In calendar format |
| end_time | End Date | | | In calendar format |
| title | Graph Title | | | |
| width | Graph Width | | | |
| height | Graph Height | | | |
| period | Graph Period | | | Possible values:<br>hour – Current Hour<br>day – Current Day<br>week – Current Week<br>month – Current Month<br>lastmonth – Previous Month<br>year – Current Year<br>lastyear – Previous Year<br>custom – Custom Period<br>If using the start_time and end_time parameters, period must be set to custom. |

## device.storage_delete - Delete a Device's Storage

This method is used to delete a device's storage.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| device_id | Device ID | Y | | |
| item | Item | Y | | |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": true
}
```

## device.storage_get - Get a Device's Storage

This method is used to get a device's storage.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| device_id | Device ID | Y | | |
| item | Item | Y | | |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": "passed"
}
```

## device.storage_list - List a Device's Storage

This method is used to list a device's storage.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| device_id | Device ID(s) | | | A single device ID or an array of device IDs |
| items | Item(s) | | | A single item or an array of items |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "8": {
      "dev_desc":"some device description",
      "baz":"val"
      "foo":"bar",
    },
    "38": {
      "dev_desc":"another device description",
      "baz":"rum"
      "foo":"beer",
    }
  }
}
```

## device.storage_set - Set a Device's Storage

This method is used to set a device's storage.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| device_id | Device ID | Y | | |
| item | Item | Y | | |
| value | Value | Y | | |
| encrypt | Encrypt | | | |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": true
}
```

## device.tag - Tag a Device

This method is used to tag a device.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| tag | Tag(s) to set | Y | | A single tag, or array of tags |
| device_id | Device ID | Y | | A single device ID, or array of device IDs |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.type_add - Add a new Device Type

This method is used to add a device type to the system. Device type id is returned on success.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| type | Device Type Name | Y | | Name for device type |
| type_{locale} | Localized Device Type Name | | | Localized (i18n) name for device type. Key name is dynamic (e.g.: type_en_US, type_pt_PT) |
| devtype_group_id | Device Type Group ID | | 0 | Optional device type group the new device type will be a member of |
| rackable_device | Rackable Device | | 1 | Flag to specify whether the height is measurable in rack units |
| rack_units | Rack Units | | 1 | Number of Rack Units which measures the height of the device. Only valid if rackable_device is true. |
| width | Width in mm | | 0 | Width of device type in millimeters. Only use if not specifying width_display_unit. |
| width_display_value | Width | | | Width of device type in units of width_display_unit |
| width_display_unit | Width Units | | | Dimensional units of width (INCHES,FEET,CENTIMETERS,METERS) |
| depth | Depth in mm | | 0 | Depth of device type in millimeters. Only use if not specifying depth_display_unit. |
| depth_display_value | Depth | | | Depth of device type in units of width_display_unit |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| depth_display_unit | Depth Units | | | Dimensional units of depth (INCHES,FEET,CENTIMETERS,METERS) |
| height | Height in mm | | 0 | Height of device type. Only used if type is not a rackable_device |
| height_display_value | Height | | | Height of device type in units of height_display_unit |
| height_display_unit | Height Units | | | Dimensional units of height (INCHES,FEET,CENTIMETERS,METERS). Defaults to MILLIMETERS. |
| editable_dimensions | Editable Dimensions | | 0 | Flag to enable/disable editing the dimensions on devices of this device type |
| client_access | Client Visible | | 0 | Flag to enable/disable visibility of this device type to clients adding new devices |
| metadata_applies[{variable}] | Metadata Applicability | | | Set whether the metadata specified by variable name applies to the device type. Accepted values are (-1: Default, 0: No, 1: Yes, 2: Required) |
| front_image_file_id | Front Image File ID | | 0 | File id to use for the device type front image |
| back_image_file_id | Back Image File ID | | 0 | File id to use for the device type back image |
| block_delete_if_connected | Block Delete if Connected | | 0 | Prevent device deletion if it has Connections |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": "3"
}
```

## device.type_delete - Delete a Device Type

This method is used to delete a device type in the system. Child devices must be migrated to a new type

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| type_id | Device Type ID | Y | | Device type ID |
| new_type_id | New Type ID | Y | | Migrate devices of old type to the new type specified by this ID. To leave devices with no new type provide an ID of 0. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.type_group_add - Add Device Type Group

This method is used to add a device type group in the system. Device type group id is returned on success.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| name | Device Type Group Name | Y | | Name for device type group |
| name_{locale} | Localized Device Type Group Name | | | Localized (i18n) name for device type group. Key name is dynamic (e.g.: name_en_US, name_pt_PT) |
| metadata_applies[ {variable}] | Metadata Applicability | | | Set whether metadata applies to the device type group. Accepted values are 0 - No, 1 - Yes, 2 - Required |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": "189"
}
```

## device.type_group_delete - Delete Device Type Group

This method is used to delete a device type group in the system.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| devtype_group_id | Device Type Group ID | Y | | Device type group ID to delete. |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": "true"
}
```

## device.type_group_list - List Device Type Groups

This method is used to list device type groups in the system.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| devtype_group_id | Device Type Group ID | | | Return only the specified device type group. |
| metadata | Include Metadata | | 0 | Set to 1 to include metadata in the output. |
| modules | Device Modules | | 0 | Set to 1 to include modules in the output. |

**Example Output**

```
{
  "status":true,
  "error_code":null,
  "error_message":"",
  "data": {
    "2": {
      "devtype_group_id":"2",
      "name":"Load Balancer",
      "priority":"2",
      "name_en_US":"Load Balancer"
    },
    "3": {
      "devtype_group_id":"3",
      "name":"PDU",
      "priority":"3",
      "name_en_US":"PDU"
    },
    "4": {
      "devtype_group_id":"4",
```

```
            "name":"Switch",
            "priority":"4",
            "name_en_US":"Switch"
            "module_details": {
              "dm_bandwidth": {
                "device_module_id":"5",
                "title":"Bandwidth Monitoring",
                "priority":"1",
                "client_access":"0",
                "enabled":1,
                "config": {
                  "appliance_id":"3",
                  "limit_view":"0",
                  "outneg":"0",
                  "actions":"1",
                  "use_switch_ports":"1"
                }
              },
              "dm_snmp_monitor": {
                "device_module_id":"8",
                "title":"SNMP Monitor",
                "priority":"5",
                "client_access":"0",
                "enabled":0,
                "config": {
                  "oids":"1.3.6.1.2.1.10.132.6.3\n1.3.6.1.2.1.10.132.6.6\n1.3.6.1.2.1.10.132.4.0"
                }
              }
            }
          }
        },
        "0": {
          "devtype_group_id":0,
          "name":"Ungrouped"
        }
      }
    }
  }
}
```

## device.type_group_module_add - Add a Device Module to Device Type Group

This method is used to add a device module to a device type group.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| devtype_group_id | Device Type Group ID | Y | | ID of device type group to update |
| module | Device Module Name | Y | | Name of device module. Names are obtained from device.raw_module_list |
| enabled | Enabled | | | Enable or disable the module by setting the value to 1 or 0, respectively |
| priority | Module Priority | | | Set priority of the module (over similar device type module). 1 indicates highest priority, while 10 is the lowest. |
| client_access | Client Access | | | Set client access to None, View, or Edit using a value of 0, 1, or 2 |
| config[{variable}] | Module Config | | | Module configuration variable name is used to create a valid key, valid values vary per module. Be warned that modules are meant to be configured through the web admin and that making changes using this feature is unsupported unless directed to do so by Ubersmith staff. |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": "6"
}
```

## device.type_group_module_delete - Remove a Device Module from a Device Type Group

This method is used to delete a device module from a device type group.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| module_id | Device Module ID | Y | | ID of device module to remove from device type group |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.type_group_module_update - Configure a Device Module in a Device Type Group

This method is used to configure a device module in a device type group.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| module_id | Device Module ID | Y | | ID of device type group module to update |
| enabled | Enabled | | | Enable or disable the module by setting the value to 1 or 0, respectively |
| priority | Module Priority | | | Set priority of the module (over similar device type module). 1 indicates highest priority, while 10 is the lowest. |
| client_access | Client Access | | | Set client access to None, View, or Edit using a value of 0, 1, or 2 |
| config[{variable}] | Module Config | | | Module configuration variable name is used to create a valid key, valid values vary per module. Be warned that modules are meant to be configured through the web admin and that making changes using this feature is unsupported unless directed to do so by Ubersmith staff. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.type_group_update - Update a Device Type Group

This method is used to update a device type group.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| devtype_group_id | Device Type Group ID | Y | | ID of device type group to update |
| name | Device Type Group Name | | | Name for device type group |
| name_{locale} | Localized Device Type Group Name | | | Localized (i18n) name for device type group. Key name is dynamic (e.g.: name_en_US, name_pt_PT) |
| metadata_applies[{variable}] | Metadata Applicability | | | Set whether metadata applies to the device type group. Accepted values are 0 - No, 1 - Yes, 2 - Required |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.type_list - List Device Types

This method is used to list device types.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| devtype_group_id | Type Group ID | | | |
| type_id | Type ID | | | |
| metadata | Metadata | | | |
| modules | Device Modules | | | |

**Example Output**

```
{
  "data": {
    "5": {
      "devtype_group_id":"5",
      "name":"Servers",
      "priority":"4",
      "name_en_US":"Servers",
      "types": {
        "1": {
          "type_id":"1",
          "devtype_group_id":"5",
          "client_access":"1",
          "editable_dimensions":"0",
          "rackable_device":"1",
          "rack_units":"1",
          "height":"44.450000",
          "width":"482.600000",
          "depth":"609.600000",
          "front_image_file_id":"0",
          "back_image_file_id":"0",
          "block_delete_if_connected": "0",
          "type":"Server",
          "type_en_US":"Server",
          "height_display_value": 1.75,
          "height_display_unit": "inches",
          "width_display_value": 19,
          "width_display_unit": "inches",
          "depth_display_value": 2,
          "depth_display_unit": "feet"
        }
      }
    }
  },
  "error_message": "",
  "error_code": null,
```

```
  "status": true
}
```

## device.type_module_add - Add a Device Module to Device Type

This method is used to add a device module to a device type.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| devtype_id | Device Type ID | Y | | ID of device type to update |
| module | Device Module Name | Y | | Name of device module. Names are obtained from device.raw_module_list |
| enabled | Enabled | | | Enable or disable the module by setting the value to 1 or 0, respectively |
| priority | Module Priority | | | Set priority of the module (over similar device type module). 1 indicates highest priority, while 10 is the lowest. |
| client_access | Client Access | | | Set client access to None, View, or Edit using a value of 0, 1, or 2 |
| config[{variable}] | Module Config | | | Module configuration variable name is used to create a valid key, valid values vary per module. Be warned that modules are meant to be configured through the web admin and that making changes using this feature is unsupported unless directed to do so by Ubersmith staff. |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": "19"
}
```

## device.type_module_delete - Remove a Device Module from a Device Type

This method is used to delete a device module from a device type.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| module_id | Device Module ID | Y | | ID of device module to remove from device type |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.type_module_update - Configure a Device Module in a Device Type

This method is used to configure a device module in a device type.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| module_id | Device Module ID | Y | | ID of device module to remove from device type |
| enabled | Enabled | | | Enable or disable the module by setting the value to 1 or 0, respectively |
| priority | Module Priority | | | Set priority of the module (over similar device type module). 1 indicates highest priority, while 10 is the lowest. |
| client_access | Client Access | | | Set client access to None, View, or Edit using a value of 0, 1, or 2 |
| config[{variable}] | Module Config | | | Module configuration variable name is used to create a valid key, valid values vary per module. Be warned that modules are meant to be configured through the web admin and that making changes using this feature is unsupported unless directed to do so by Ubersmith staff. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.type_update - Edit a Device Type

This method is used to edit a device type in the system.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| type_id | Device Type ID | Y | | Device type ID |
| type | Device Type Name | | | Name for device type |
| type_{locale} | Localized Device Type Name | | | Localized (i18n) name for device type. Key name is dynamic (e.g.: type_en_US, type_pt_PT) |
| devtype_group_id | Device Type Group ID | | | The device type will be added to the specified device type group |
| rackable_device | Rackable Device | | | Flag to specify whether the height is measurable in rack units |
| rack_units | Rack Units | | | Number of Rack Units which measures the height of the device. Only valid if rackable_device is true. |
| width | Width in mm | | | Width of device type in millimeters. Only use if not specifying width_display_unit. |
| width_display_value | Width | | | Width of device type in units of width_display_unit |
| width_display_unit | Width Units | | | Dimensional units of width (INCHES,FEET,CENTIMETERS,METERS) |
| depth | Depth in mm | | | Depth of device type in millimeters. Only use if not specifying depth_display_unit. |
| depth_display_value | Depth | | | Depth of device type in units of width_display_unit |
| depth_display_unit | Depth Units | | | Dimensional units of depth (INCHES,FEET,CENTIMETERS,METERS) |
| height | Height in mm | | | Height of device type. Only used if type is not a rackable_device |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| height_display_value | Height | | | Height of device type in units of height_display_unit |
| height_display_unit | Height Units | | | Dimensional units of height (INCHES,FEET,CENTIMETERS,METERS). Defaults to MILLIMETERS. |
| editable_dimensions | Editable Dimensions | | | Flag to enable/disable editing the dimensions on devices of this device type |
| client_access | Client Visible | | | Flag to enable/disable visibility of this device type to clients adding new devices |
| metadata_applies[{variable}] | Metadata Applicability | | | Set whether the metadata specified by variable name applies to the device type. Accepted values are (-1: Default, 0: No, 1: Yes, 2: Required) |
| front_image_file_id | Front Image File ID | | | File id to use for the device type front image |
| back_image_file_id | Back Image File ID | | | File id to use for the device type back image |
| remove_front_image | Remove Front Image | | | Set to 1 to remove the device type front image |
| remove_back_image | Remove Back Image | | | Set to 1 to remove the device type back image |
| block_delete_if_connected | Block Delete if Connected | | | Prevent device deletion if it has Connections |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.untag - Untag a Device

This method is used to untag a device.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| tag | Tag(s) to remove | Y | | A single tag, or array of tags |
| device_id | Device ID | Y | | A single device ID, or array of device IDs |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

### device.update - Update a Device

This method is used to update a device.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| device_id | Device ID | Y | 0 | |
| dev_desc | Device Description | | | |
| label | Device Label | | | |
| parent | Parent Device ID | | 0 | |
| type_id | Device Type ID | | 0 | |
| rack_id | Rack ID | | 0 | |
| rack_pos | Position Within Rack | | | |
| height | Device Height | | 1 | |
| client_id | Client ID | | | |
| service_id | Service ID | | | |
| client_owned | Client Owns Physical Device | | 0 | 0 = Client does not own physical device, 1 = Client owns physical device |
| connection_node_type_id | Connection Node Type ID | | | |
| front_image_file_id | Front Image File ID | | | A file id to use for the device's front image |
| back_image_file_id | Back Image File ID | | | A file id to use for the device's back image |
| front_image | Attachment | | | An image file for the device's front image. When using cURL the value for the parameter should be @ where is the image you wish to attach. |
| back_image | Attachment | | | An image file for the device's back image. When using cURL the value for the parameter should be @ where is the image you wish to attach. |
| remove_front_image | Remove Front Image | | | Set to 1 to remove the device's front image |
| remove_back_image | Remove Back Image | | | Set to 1 to remove the device's back image |
| meta_* | Custom Data | | | Metadata items may be set by passing the values as additional parameters matching the metadata items API field names prefixed with 'meta_'. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## device.vlan_get_available - List Available VLANs

This method is used to list available VLAN numbers, optionally filtered by: - VLAN range id (vlan_range_id) - facility id (fac_id) - HSRP (is_hsrp) optionally limited by: - limit (limit). Returns an array of all available VLAN numbers with VLAN range ID and title.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| vlan_range_id | VLAN Range ID | | | |
| fac_id | Facility ID | | | |
| limit | Results Limit | | | |
| is_hsrp | HSRP Flag | | | |

# order - Order

Functions related to orders

## order.cancel - Cancel an Order

This method is used to cancel a specified order.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| order_id | Order ID | | | Required if hash is not specified |
| hash | Order Hash | | | Required if order_id is not specified |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## order.client_respond - Post a Client/Lead Order Response

This method is used to post a response on the specified order as a client or lead.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| order_id | Order ID | | | Required if hash is not specified |
| hash | Order Hash | | | Required if order_id is not specified |
| contact_id | Contact ID | | | Required if author is not specified |
| author | Author Email | | | Required if contact_id is not specified |
| subject | Subject | | | |
| cc | Cc | | | |
| body | Response | | (no message) | |
| attach | Attachments | | | |

**Example Output**

```
{
  "data": "17",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## order.coupon_add - Add Coupon

This method is used to add a new coupon.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| coupon_code | Coupon Code | | | The coupon code |
| description | Description | | | The coupon description |
| plan_id | Plan ID | | | The plan ID that this coupon applies to |
| start | Start | | | The date the coupon becomes active |
| expire | Expire | | | The expiration date of the coupon |
| recurring | Recurring | | | Set to 0 for one-time use coupons, 1 for recurring coupons |
| discount_value | Discount Value | | | The numeric value of the discount for this coupon |
| dollar | Dollar | | | Set to 0 if the discount_value is a percentage, 1 if it is a dollar amount |
| setup_discount_value | Setup Discount Value | | | The numeric value of the setup discount for this coupon |
| setup_dollar | Setup Dollar | | | Set to 0 if the setup_discount_value is a percentage, 1 if it is a dollar amount |
| max_uses | Max Uses | | | The maximum number of times the coupon can be used |
| class_id | Class ID | | | The class ID that this coupon applies to |
| active | Active | | | Set to 0 to mark coupon as inactive, 1 to make it active |
| vip | VIP | | | Set to 0 to make coupon available to all clients, 1 for existing clients only |
| description_ | Locale description | | | Coupon description for language specified by locale |
| opts | Service Plan Options | | | The values for the service plan options based on the plan_id provided. Format:<br><br>opts[spo_id][discount_type]=0<br>opts[spo_id][discount]=10<br>opts[spo_id][setup_discount_type]=1<br>opts[spo_id][setup_discount]=5.00 |

**Example Output**

```
{
  "data": "225",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## order.coupon_client_add - Add Coupon Client

This method is used to add client(s) to a coupon.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| coupon_id | Coupon ID | Y | | The coupon id to add the client(s) to |
| client_id | Client ID | | | Specify a single client ID or multiple client IDs to add. Format: <br><br> client_id=X -OR- client_id[0]=X client_id[1]=Y |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## order.coupon_client_delete - Delete Coupon Client

This method is used to delete client(s) from a coupon.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| coupon_id | Coupon ID | Y | | The coupon id to delete the client from |
| client_id | Client ID | | | Specify a single client ID or multiple client IDs to delete. Format: <br><br> client_id=X -OR- client_id[0]=X client_id[1]=Y |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## order.coupon_get - Get Order Coupon Details

This method is used to get the details of a specified order coupon.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| coupon_id | Coupon ID | | | Required if coupon_code is not specified |
| coupon_code | Coupon Code | | | Required if coupon_id is not specified |
| brand_id | Brand ID | | | Optional, used only if coupon_code is specified |

**Example Output**

```
{
  "data": {
    "coupon": {
      "total_uses": "3",
      "coupon_id": "1",
      "setup_dollar": "0",
      "setup_discount_value": "10",
      "class_id": "1",
      "description_en_GB": "Promotional Sale",
      "start": "1224820800",
      "coupon_code": "CK91F8",
      "vip": "0",
      "description_en_US": "Promotional Sale",
      "discount_value": "45",
      "description": "Promotional Sale",
      "max_uses": "0",
      "recurring": "0",
      "expire": "0",
      "active": "1",
      "dollar": "0",
      "plan_id": "0"
    },
    "options": {
      "31": {
        "setup_discount_type": "0",
        "coupon_id": "1",
        "discount": "100.00",
        "setup_discount": "0.00",
        "discount_type": "0",
        "spo_id": "31"
      }
    },
    "clients": [

    ]
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## order.coupon_list - Coupon List

This method is used to retrieve list of coupons.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| plan_id | Service Plan ID | | | Find the coupons for this service plan. |
| start | Coupon Start Date | | | Find the coupons that are valid on or after this timestamp. |
| expire | Coupon Expiration Date | | | Find the coupons that expire on or before this timestamp. |
| class_id | Class ID | | | The class_id that the coupons apply to. |
| active | Active | | | Set to 1 to find active coupons, 0 to find inactive coupons, or omit to find all. |
| include_options | Include Options | | | Set to 1 to include service plan options in the output. |
| include_clients | Include Clients | | | Set to 1 to include clients enabled for the coupon in the output. |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
"data": {
"123": {
"coupon_id": "123",
"coupon_code": "CK91F8",
"description": "Coupon description",
"plan_id": "342",
"start": "1380122574",
"expire": "1380122574",
"recurring": "1",
"discount_value": "10",
"dollar": "0",
"setup_discount_value": "5",
"setup_dollar": "1",
"max_uses": "2",
"total_uses": "1",
"class_id": "1",
"active": "1",
"vip": "0",
"description_en_US": "Coupon description en_US",
"description_zh_CN": "Coupon description zh_CN",
"description_fr_FR": "Coupon description fr_FR",
"description_fr_CA": "Coupon description fr_CA",
"description_pt_PT": "Coupon description pt_PT",
"description_ja_JP": "Coupon description ja_JP",
"description_el_GR": "Coupon description el_GR",
"description_de_DE": "Coupon description de_DE",
"description_tr_TR": "Coupon description tr_TR",
"description_sv_SE": "Coupon description sv_SE",
"description_zh_TW": "Coupon description zh_TW",
"description_en_GB": "Coupon description en_GB",
"description_es_ES": "Coupon description es_ES",
"options": {
"0": {
"coupon_id": "123",
"spo_id": "31",
"discount_type": "0",
"discount": "100.00",
"setup_discount_type": "0",
"setup_discount": "100.00"
},
"1": {
"coupon_id": "123",
"spo_id": "34",
"discount_type": "0",
```

```
"discount": "100.00",
"setup_discount_type": "0",
"setup_discount": "100.00"
}
},
"clients": {
"43": {
"clientid": "43"
},
"48": {
"clientid": "48"
}
}
}
}
}
```

## order.coupon_update - Update Coupon Details

This method is used to update the details for a coupon.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| coupon_id | Coupon ID | Y | | The id of the coupon to update |
| coupon_code | Coupon Code | | | The coupon code |
| description | Description | | | The coupon description |
| plan_id | Plan ID | | | The plan ID that this coupon applies to, set to 0 to remove |
| start | Start | | | The date the coupon becomes active |
| expire | Expire | | | The expiration date of the coupon |
| recurring | Recurring | | | Set to 0 for one-time use coupons, 1 for recurring coupons |
| discount_value | Discount Value | | | The numeric value of the discount for this coupon |
| dollar | Dollar | | | Set to 0 if the discount_value is a percentage, 1 if it is a dollar amount |
| setup_discount_value | Setup Discount Value | | | The numeric value of the setup discount for this coupon |
| setup_dollar | Setup Dollar | | | Set to 0 if the setup_discount_value is a percentage, 1 if it is a dollar amount |
| max_uses | Max Uses | | | The maximum number of times the coupon can be used |
| class_id | Class ID | | | The class_id that this coupon applies to |
| active | Active | | | Set to 0 to mark coupon as inactive, 1 to make it active |
| vip | VIP | | | Set to 0 to make coupon available to all clients, 1 for existing clients only |
| description_ | Locale description | | | Coupon description for language specified by locale |
| opts | Service Plan Options | | | The values for the service plan options based on the plan_id provided. Format:<br><br>opts[spo_id][discount_type]=0<br>opts[spo_id][discount]=10<br>opts[spo_id][setup_discount_type]=1<br>opts[spo_id][setup_discount]=5.00 |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## order.create - Create a New Order

This method is used to create a new order.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| order_queue_id | Order Queue ID | Y | | |
| ts | Timestamp | | current time | |
| order_form_id | Order Form ID | | 0 | |
| order_status | Step ID | | Leads Step ID | |
| priority | Order Priority | | 1 | |
| owner | Admin ID | | | |
| client_id | Client ID | | | |
| requested_completion | Requested Completion | | | |
| planned_completion | Planned Completion | | | |
| listed_company | Listed Company | | | |
| total | Order Total | | | |
| hash | Order Hash | | automatic | |
| activity | Activity Timestamp | | | |
| progress | Progress | | | |
| info | Order Information | | | |
| signature | Signature Data | | | |

**Example Output**

```
{
  "data": {
    "data_changed": false,
    "step": {
      "data": {
        "name": "Leads",
        "priority": "1",
        "order_queue_id": "8",
        "order_step_id": "30"
      },
      "queue": {
        "data": {
          "name": "Order Queue New",
          "sendlink_body": "",
          "sendlink_sub": "Newest Queue",
          "class_id": "1",
          "i18n": {
            "6": {
              "name": "",
              "sendlink_body": "",
              "sendlink_sub": "",
              "lang_id": "6",
              "id": "75",
              "order_queue_id": "8"
            },
            "1": {
              "name": "Order Queue New",
              "sendlink_body": "",
              "sendlink_sub": "Newest Queue",
              "lang_id": "1",
              "id": "74",
              "order_queue_id": "8"
            }
          },
          "order_queue_id": "8",
          "sendlink_from": "support@ubersmith.com",
          "active": "1"
        },
        "order_queue_fields": [
```

```
          "class_id",
          "active",
          "name",
          "sendlink_from",
          "sendlink_sub",
          "sendlink_body"
        ],
        "i18n_column": [
          "name",
          "sendlink_sub",
          "sendlink_body"
        ],
        "order_queue_id": "8"
      },
      "order_step_id": "30"
    },
    "data": {
      "method": "order.create",
      "hash": "286fe-6a593",
      "signature": "",
      "progress": [

      ],
      "activity": 1312304629,
      "total": "0.00",
      "priority": 1,
      "info": {
        "coupon_ids": [

        ],
        "coupon_credits": [

        ],
        "taxes": [

        ]
      },
      "order_status": "30",
      "opportunity_id": "0",
      "requested_completion": "1323950400",
      "planned_completion": "1323950400",
      "client_id": "",
      "listed_company": "",
      "order_form_id": 0,
      "order_queue_id": "8",
      "opportunity_id": 0,
      "ts": 1312304629,
      "owner": ""
    },
    "order_id": "1281",
    "form": {
      "data": null,
      "id": 0,
      "queue": null
    },
    "summary": "\n\t\t\t<table cellpadding=\"3\">\n\t\t\t\t<tr>\n\t\t\t\t\t<td colspan=\"2\">Order Summary</td>\n\t\t\t\t\t<td
class=\"DetailLabel\" width=\"15%\" align=\"right\" style=\"border-bottom:1px solid
#cccccc;\">Price/Period</td>\n\t\t\t\t\t<td> </td>\n\t\t\t\t\t<td class=\"DetailLabel\" width=\"15%\" align=\"right\"
style=\"border-bottom:1px solid #cccccc;\">Setup</td>\n\t\t\t\t</tr>\n\t\t\t\t<tr>\n\t\t\t\t\t<td>Total
Cost</td>\n\t\t\t\t\t<td align=\"right\">$</td>\n\t\t\t\t\t<td class=\"DetailLabel\" align=\"right\" style=\"border-top:1px
solid #cccccc;\">0.00</td>\n\t\t\t\t\t<td> </td>\n\t\t\t\t\t<td class=\"DetailLabel\" align=\"right\"
style=\"border-top:1px solid #cccccc;\">0.00</td>\n\t\t\t\t</tr>\n\t\t\t</table>",
    "queue": {
      "data": {
        "name": "Order Queue New",
        "sendlink_body": "",
        "sendlink_sub": "Newest Queue",
        "class_id": "1",
        "i18n": {
          "6": {
            "name": "",
            "sendlink_body": "",
            "sendlink_sub": "",
            "lang_id": "6",
            "id": "75",
            "order_queue_id": "8"
          },
          "1": {
            "name": "Order Queue New",
```

```
          "sendlink_body": "",
          "sendlink_sub": "Newest Queue",
          "lang_id": "1",
          "id": "74",
          "order_queue_id": "8"
        }
      },
      "order_queue_id": "8",
      "sendlink_from": "support@ubersmith.com",
      "active": "1"
    },
    "order_queue_fields": [
      "class_id",
      "active",
      "name",
      "sendlink_from",
      "sendlink_sub",
      "sendlink_body"
    ],
    "i18n_column": [
      "name",
      "sendlink_sub",
      "sendlink_body"
    ],
    "order_queue_id": "8"
  }
},
"error_message": "",
"error_code": null,
"status": true
}
```

## order.get - Get Order Details

This method is used to get the details of a specified order.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| order_id | Order ID | | | Required if hash is not specified |
| hash | Order Hash | | | Required if order_id is not specified |

**Example Output**

```
{
  "data": {
    "hash": "286fe-6a593",
    "order_id": "1281",
    "signature": "",
    "progress": [

    ],
    "activity": "1312304629",
    "total": "0.00",
    "priority": "1",
    "info": {
      "coupon_ids": null,
      "coupon_credits": null,
      "taxes": null
    },
    "order_status": "30",
    "client_id": "0",
    "opportunity_id": "0",
    "requested_completion": "1323950400",
    "planned_completion": "1323950400",
    "listed_company": "",
    "order_form_id": "0",
    "order_queue_id": "8",
    "opportunity_id": "0",
    "ts": "1312304629",
    "owner": ""
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## order.list - List Orders

This method is used to get a list of orders.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| order_step_id | Order Step ID | | | |
| order_queue_id | Order Queue ID | | | Ignored if order_step_id is specified. |
| brand_id | Brand ID | | system default | Ignored if order_step_id or order_queue_id is specified. |
| step_name | Order Step Name | | | Ignored if order_step_id is specified. Also ignored if step_status is provided. |
| step_status | Order Step Status | | | Ignored if order_step_id is specified. Overrides step_name. 0 = Normal, 1 = Leads, 2 = Complete, 3 = Canceled |
| min_ts | Minimum Timestamp | | | If specified only orders received after this timestamp are returned. |
| max_ts | Maximum Timestamp | | | If specified only orders received before this timestamp are returned. |
| client_id | Client ID | | | |
| opportunity_id | Opportunity ID | | | |
| order_by | Order By | | ts | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "1280": {
      "hash": "67eda-f6ce3",
      "order_id": "1280",
      "signature": "",
      "progress": [

      ],
      "activity": "1312304618",
      "total": "0.00",
      "priority": "1",
      "info": {
        "coupon_ids": null,
        "coupon_credits": null,
        "taxes": null
      },
      "order_status": "30",
      "client_id": "0",
      "opportunity_id": "0",
      "requested_completion": "1323950400",
      "planned_completion": "1323950400",
      "listed_company": "",
      "order_form_id": "0",
      "order_queue_id": "8",
      "opportunity_id": "0",
      "ts": "1312304618",
      "owner": ""
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## order.module_get - Order Module Details

This method is used to retrieve details for a single order module.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| order_module_na me | Order Module Name | Y | | The name of the order module to retrieve |

**Example Output**

```
{
"data": {
"module_name": {
"short_name": "module_name",
"long_name": "Friendly Module Name",
"config_items": {
"item_name": {
"label": "Item Label",
"type": "select",
"options": {
"value": "Option Value Name",
"another_value": "Another Option Value Name"
},
"default": "value"
}
}
}
},
"error_message": "",
"error_code": null,
"status": true
}
```

## order.module_list - Order Modules

This method is used to retrieve a list of available order modules.

**Example Output**

```
{
"data": {
"module_name": "Friendly Module Name"
},
"error_message": "",
"error_code": null,
"status": true
}
```

## order.post_list - Get all posts for an Order

This method is used to return all the posts for the specified order.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| order_id | Order ID | Y | | |
| private | Show only client-viewable posts | | 0 | Set to 1 to hide posts which are not client viewable. |
| direction | Direction | | asc | Set to desc to return posts in descending order (newest first) |
| offset | Offset | | | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return, only used if offset is specified |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": {
    "128": {
      "ticket": "128",
      "order_id": "1490",
      "contact_id": "250",
      "author": "Johnny Ubersmith <uber@ubersmith.com>",
      "recipient": "luke@ubersmith.com",
      "subject": "Subject",
      "attachment": "0",
      "timestamp": "1368823390",
      "type": "Staff Followup",
      "source": "Staff",
      "assignment": "1",
      "activity": "0",
      "cc": "",
      "body": "body",
      "priority": "1",
      "q_id": "0",
      "weight": null,
      "rem_assign": "0",
      "public_comment": "0",
      "company_assignment": "0",
      "activity_type": "none",
      "headers": null
    }
  }
}
```

## order.process - Process an Order

This method is used to process an order.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| order_action_id | Order Action ID | Y | | The action to process |
| order_id | Order ID | | | Required if hash is not specified |
| hash | Order Hash | | | Required if order_id is not specified |
| skip | Skip Action | | 0 | Set to 1 to skip the specified action rather than processing it. |
| undo_skip | Undo Skip | | 0 | Set to 1 to revert skipped status of an action, allowing further processing. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## order.queue_action_add - Add Order Action

This method is used to add a new order action.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| order_step_id | Order Step ID | Y | | The ID of the order step the new action will belong to |
| name | Name | | | The name of the order action |
| module | Module Name | Y | | The name of the order module associated with the order action |
| priority | Priority | | | The priority of the order action |
| automatic | Automatic | | | Set to 1 for automatic |
| optional | Optional | | | Set to 1 if optional |
| client_viewable | Client Viewable | | | Set to 1 if client viewable |
| prereqs | Prerequisite Order Actions | | | The other order action IDs in this step that are prerequisites for this action. Set to value to 1. Format:<br><br>prereqs[order_action_id]=1 |
| config | Config | | | The config values for the order module associated with this action. Format:<br><br>config[field]=value |
| i18n | Internationalization Options | | | The internationalized config for this order step. Substitute lang_id with the numeric language id or language locale, e.g. en_US. Options are:<br><br>i18n[lang_id][name] |

**Example Output**

```
{
"data": "123",
"error_message": "",
"error_code": null,
"status": true
}
```

## order.queue_action_delete - Delete an order action

This method is used to delete the given order action ID.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| order_action_id | Order Action ID | Y | | The ID of the order action to delete |

**Example Output**

```
{
"data": true,
"error_message": "",
"error_code": null,
"status": true
}
```

## order.queue_action_get - Order Queue Action Details

This method is used to retrieve details for a single order queue action.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| order_action_id | Order Action ID | Y | | The ID of the order queue action to retrieve |

**Example Output**

```
{
"data": {
"order_action_id": "45",
"order_step_id": "25",
"name": "Action Name",
"module": "order_module",
"priority": "1",
"automatic": "1",
"optional": "0",
"client_viewable": "0",
"prereqs": {
"44": "44"
},
"config": {
"order_module_config_field_name": "value"
},
"i18n": {
"1": {
"id": "376",
"order_action_id": "111",
"lang_id": "1",
"name": "Translated Action Name"
}
},
"module_config_items": {
"field1": {
"label": "Field1 Label",
"type": "text",
"size": "10"
"default: "0"
},
"field2": {
"label": "Field2 Label",
"type": "select",
"options": {
"email": "me@test.com",
"username": "myusername"
},
"default": "email"
}
}
},
"error_message": "",
"error_code": null,
"status": true
}
```

## order.queue_action_list - Order Queue Actions

This method is used to retrieve the order actions for an order step ID.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| order_step_id | Order Step ID | Y | | The ID of the order step to retrieve the actions for |

**Example Output**

```
{
"data": {
"44": {
"order_action_id": "44",
"order_step_id": "25",
"name": "Action Name",
"module": "order_module",
"priority": "1",
"automatic": "1",
"optional": "0",
"client_viewable": "0",
"prereqs": {
},
"config": {
}
},
"45": {
"order_action_id": "45",
"order_step_id": "25",
"name": "Action Name",
"module": "order_module",
"priority": "1",
"automatic": "1",
"optional": "0",
"client_viewable": "0",
"prereqs": {
"44": "44"
},
"config": {
"order_module_config_field_name": "value"
}
}
},
"error_message": "",
"error_code": null,
"status": true
}
```

## order.queue_action_update - Update Order Action Details

This method is used to update the details for an order action ID.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| order_action_id | Order Action ID | Y | | The ID of the order action to update |
| name | Name | | | The name of the order action |
| module | Module Name | | | The name of the order module associated with the order action |
| priority | Priority | | | The priority of the order action |
| automatic | Automatic | | | Set to 1 for automatic |
| optional | Optional | | | Set to 1 if optional |
| client_viewable | Client Viewable | | | Set to 1 if client viewable |
| prereqs | Prerequisite Order Actions | | | The other order action IDs in this step that are prerequisites for this action. Set to 1 to add or 0 to remove. Format: prereqs[order_action_id]=1 |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| config | Config | | | The config values for the order module associated with this action. Format:<br><br>config[field]=value |
| i18n | Internationalization Options | | | The internationalized config for this order step. Substitute lang_id with the numeric language id or language locale, e.g. en_US. Options are:<br><br>i18n[lang_id][name] |

**Example Output**

```
{
"data": true,
"error_message": "",
"error_code": null,
"status": true
}
```

## order.queue_add - Add Order Queue

This method is used to add an order queue.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| brand_id | Brand ID | | | The brand id for the order queue |
| name | Name | | | The name of the order queue |
| allow_quotes | Allow Quotes | | | Set to 1 to allow quotes |
| sendlink_from | Send Link From | | | The email address of the order email sender |
| sendlink_sub | Send Link Subject | | | The subject of the order email |
| sendlink_body | Send Link Body | | | The body of the order email |
| i18n | Internationalization Options | | | The internationalized config for this order queue. Substitute lang_id with the numeric language id or language locale, e.g. en_US. Options are:<br><br>i18n[lang_id][name]<br>i18n[lang_id][sendlink_sub]<br>i18n[lang_id][sendlink_body] |

**Example Output**

```
{
  "data": "123",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## order.queue_get - Order Queue Details

This method is used to retrieve details for a single order queue.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| order_queue_id | Order Queue ID | Y | | The ID of the order queue to retrieve |

**Example Output**

```
{
"data": {
"order_queue_id": "7",
"class_id": "1",
"active": "1",
"name": "Order Queue Name",
"allow_quotes": "0",
"sendlink_from": "username@email.com",
"sendlink_sub": "Subject",
"sendlink_body": "The body of the email message.",
"i18n": {
"1": {
"id": "53",
"order_queue_id": "7",
"lang_id": "1",
"name": "Translated Order Queue Name",
"sendlink_sub": "Translated Subject",
"sendlink_body": "Translated body of the email message."
}
},
"steps": {
"25": {
"order_step_id": "25",
"name": "Step Name",
"priority": "1",
"actions": {
"44": {
"order_action_id": "44",
"name": "Action Name",
"module": "order_module",
"priority": "1",
"automatic": "1",
"optional": "0",
"client_viewable": "0",
"prereqs": {
},
"config": {
}
},
"45": {
"order_action_id": "45",
"name": "Action Name",
"module": "order_module",
"priority": "1",
"automatic": "1",
"optional": "0",
"client_viewable": "0",
"prereqs": {
"44": "44"
},
"config": {
"order_module_config_field_name": "value"
}
}
}
}
},
"error_message": "",
"error_code": null,
"status": true
}
```

## order.queue_list - List Order Queues

This method is used to get a list of order queues.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| brand_id | Brand ID | Y | | |

**Example Output**

```
{
  "data": {
    "7": {
      "steps": {
        "25": {
          "name": "Leads",
          "count": 0
        },
        "26": {
          "name": "Processing",
          "count": "6"
        },
        "27": {
          "name": "Complete",
          "count": "1"
        },
        "28": {
          "name": "Canceled",
          "count": 0
        }
      },
      "name": "Main Order Queue",
      "order_queue_id": 7
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## order.queue_step_add - Add Order Step

This method is used to add an order step to the given order queue ID.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| order_queue_id | Order Queue ID | Y | | The ID of the order queue this step should be added to |
| name | Name | | | The name of the order step |
| priority | Priority | | | The priority of the order step |
| i18n | Internationalization Options | | | The internationalized config for this order step. Substitute lang_id with the numeric language id or language locale, e.g. en_US. Options are:<br><br>i18n[lang_id][name] |

**Example Output**

```
{
"data": "123",
"error_message": "",
"error_code": null,
"status": true
}
```

## order.queue_step_delete - Delete an order step

This method is used to delete the given order step ID.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| order_step_id | Order Step ID | Y | | The ID of the order step to delete |

**Example Output**

```
{
"data": true,
"error_message": "",
"error_code": null,
"status": true
}
```

## order.queue_step_get - Order Queue Step Details

This method is used to retrieve details for a single order queue step.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| order_step_id | Order Step ID | Y | | The ID of the order queue step to retrieve |

**Example Output**

```
{
"data": {
"order_step_id": "25",
"order_queue_id": "7",
"name": "Step Name",
"priority": "1",
"i18n": {
"1": {
"id": "53",
"order_step_id": "25",
"lang_id": "1",
"name": "Translated Order Step Name"
}
},
"actions": {
"44": {
"order_action_id": "44",
"name": "Action Name",
"module": "order_module",
"priority": "1",
"automatic": "1",
"optional": "0",
"client_viewable": "0",
"prereqs": {
},
"config": {
}
}
"45": {
"order_action_id": "45",
"name": "Action Name",
"module": "order_module",
"priority": "1",
"automatic": "1",
"optional": "0",
"client_viewable": "0",
"prereqs": {
"44": "44"
},
"config": {
"order_module_config_field_name": "value"
}
}
}
},
"error_message": "",
"error_code": null,
"status": true
}
```

## order.queue_step_list - Order Queue Steps

This method is used to retrieve the order steps for an order queue ID.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| order_queue_id | Order Queue ID | Y | | The ID of the order queue to retrieve the steps for |

**Example Output**

```
{
"data": {
"25": {
"order_step_id": "25",
"order_queue_id": "7",
"name": "Order Step Name",
"priority": "1"
}
},
"error_message": "",
"error_code": null,
"status": true
}
```

## order.queue_step_update - Update Order Step Details

This method is used to update the details for an order step ID.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| order_step_id | Order Step ID | Y | | The ID of the order step to delete |
| name | Name | | | The name of the order step |
| priority | Priority | | | The priority of the order step |
| i18n | Internationalization Options | | | The internationalized config for this order step. Substitute lang_id with the numeric language id or language locale, e.g. en_US. Options are:<br><br>i18n[lang_id][name] |

**Example Output**

```
{
"data": true,
"error_message": "",
"error_code": null,
"status": true
}
```

## order.queue_update - Update Order Queue

This method is used to update an order queue.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| order_queue_id | Order Queue ID | Y | | The ID of the order queue to update |
| active | Active | | | Set to 1 to set the order queue as active |
| name | Name | | | The name of the order queue |
| allow_quotes | Allow Quotes | | | Set to 1 to allow quotes |
| sendlink_from | Send Link From | | | The email address of the order email sender |
| sendlink_sub | Send Link Subject | | | The subject of the order email |
| sendlink_body | Send Link Body | | | The body of the order email |
| i18n | Internationalization Options | | | The internationalized config for this order queue. Substitute lang_id with the numeric language id or language locale, e.g. en_US. Options are:<br><br>i18n[lang_id][name]<br>i18n[lang_id][sendlink_sub]<br>i18n[lang_id][sendlink_body] |

**Example Output**

```
{
"data": true,
"error_message": "",
"error_code": null,
"status": true
}
```

## order.respond - Post an Order Response

This method is used to post a response on the specified order.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| order_id | Order ID | | | Required if hash is not specified |
| hash | Order Hash | | | Required if order_id is not specified |
| recipient | Recipient | | | Order Email Address |
| subject | Subject | | Your order with | |
| followup | Response | | (no message) | |
| from | From | | API | |
| comment | Comment | | 0 | Set to 1 to post a comment |
| cc | Cc | | | |
| bcc | Bcc | | | |
| user_id | User ID | | | |
| attach[] | Attachments | | | Attachment files as multipart/form-data. Array keys are disregarded but may be supplied. |

**Example Output**

```
{
  "data": "18",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## order.submit - Submit An Order

This method is used to submit the specified order.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| order_id | Order ID | | | Required if hash is not specified |
| hash | Order Hash | | | Required if order_id is not specified |

**Example Output**

```
{
  "data": "\n\t\t\t<p>\n\t\t\t\tThank you for your order, for future reference your order number is: 1260<br />\n\t\t\t\tYour
order is now being processed.\n\t\t\t<\/p>",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## order.update - Update an Order

This method is used to update the details of an existing order.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| order_id | Order ID | | | |
| hash | Order Hash | | | |
| info | Order Information | | | |
| step_id | Order Step ID | | | If specified, overrides step_name |
| step_name | Order Step Name | | | Ignored if order_step_id is specified. |
| priority | Order Priority | | 1 | |
| owner | Admin ID | | | |
| client_id | Client ID | | | |
| requested_completion | Requested Completion | | | |
| planned_completion | Planned Completion | | | |
| listed_company | Listed Company | | | |
| total | Order Total | | | |
| signature | Signature Data | | | |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## sales - Sales

Functions related to sales opportunities

### sales.opportunity_add - Add an Opportunity

This method is used to add a new opportunity to the Sales Manager.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| user_login | Client/Lead Login Name or ID | | | |
| client_id | Client/Lead ID | | | |
| contact_id | Contact ID | | | |
| status | Opportunity Status | | | |
| opportunity_type_id | Opportunity Type | Y | | Use sales.opportunity_type_list to look up the IDs for defined types. |
| opportunity_stage_id | Opportunity Stage | Y | | Use sales.opportunity_stage_list to look up the IDs for defined stages. |
| owner | Owner ID | | | |
| closure_ts | Closure Timestamp | | | |
| closure_pct | Closure Percentage | | | |
| price_min | Minimum Price | | | |
| price_max | Maximum Price | | | |
| name | Opportunity Name | | | |
| last_action | Last Action | | | |
| next_step | Next Step | | | |
| description | Description | | | |

**Example Output**

```
{
  "data": "11",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## sales.opportunity_list - List Opportunities

This method is used to retrieve a list of opportunities from the Sales Manager.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| opportunity_id | Opportunity ID | | | |
| brand_id | Brand ID | | | |
| user_login | Client/Lead Login Name or ID | | | |
| client_id | Client/Lead ID | | | |
| contact_id | Contact ID | | | |
| status | Opportunity Status | | | |
| opportunity_type_id | Opportunity Type | Y | | Use sales.opportunity_type_list to look up the IDs for defined types. |
| opportunity_stage_id | Opportunity Stage | Y | | Use sales.opportunity_stage_list to look up the IDs for defined stages. |
| owner | Owner ID | | | |
| min_value | Minimum Value | | | |
| max_value | Maximum Value | | | |
| min_activity | Minimum Activity Timestamp | | | |
| price_min | Minimum Price | | | |
| price_max | Maximum Price | | | |
| begin | Begin Timestamp | | | |
| end | End Timestamp | | | |
| order_by | Order By | | ts | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return. |

**Example Output**

```
{
  "data": {
    "1": {
      "data": {
        "last_action": "",
        "name": "Server Farm",
        "opportunity_type_id": "1",
        "stage": "Contacting",
        "activity": "1244483396",
        "contact_id": "54",
        "price_max": "0.00",
        "closure_ts": "0",
        "owner_name": "Administrator",
        "type": "Blade Servers",
        "next_step": "",
        "value": "0.000000",
        "client_id": "1054",
        "listed_company": "Client",
        "description": "",
        "price_min": "0.00",
        "closure_pct": "0",
        "status": "1",
        "ts": "1244483396",
        "opportunity_id": "1",
        "owner": "1",
        "opportunity_stage_id": "1"
      }
    }
```

```
    },
    "error_message": "",
    "error_code": null,
    "status": true
}
```

## sales.opportunity_stage_list - List Opportunity Stages

This method is used to retrieve a list of opportunity stages from the Sales Manager.

**Example Output**

```
{
    "data": {
        "1": "Contact",
        "2": "Meeting",
        "3": "Negotiation",
        "4": "Buying",
        "5": "Closed"
    },
    "error_message": "",
    "error_code": null,
    "status": true
}
```

## sales.opportunity_status_list - List Opportunity Statuses

This method is used to retrieve a list of opportunity statuses from the Sales Manager.

**Example Output**

```
{
    "data": {
        "0": "Deactivated",
        "1": "Open",
        "2": "Won",
        "4": "Lost"
    },
    "error_message": "",
    "error_code": null,
    "status": true
}
```

## sales.opportunity_type_list - List Opportunity Types

This method is used to retrieve a list of opportunity types from the Sales Manager.

**Example Output**

```
{
    "data": {
        "1": "Colocation",
        "2": "Blade Servers"
    },
    "error_message": "",
    "error_code": null,
    "status": true
}
```

## sales.opportunity_update - Update an Opportunity

This method is used to update an opportunity in the Sales Manager.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| opportunity_id | Opportunity ID | Y | | |
| contact_id | Contact ID | | | |
| status | Opportunity Status | | | |
| opportunity_type_id | Opportunity Type | Y | | Use sales.opportunity_type_list to look up the IDs for defined types. |
| opportunity_stage_id | Opportunity Stage | Y | | Use sales.opportunity_stage_list to look up the IDs for defined stages. |
| owner | Owner ID | | | |
| closure_ts | Closure Timestamp | | | |
| closure_pct | Closure Percentage | | | |
| price_min | Minimum Price | | | |
| price_max | Maximum Price | | | |
| name | Opportunity Name | | | |
| last_action | Last Action | | | |
| next_step | Next Step | | | |
| description | Description | | | |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

# support - Support

Functions related to support departments and tickets

## support.classification_list - List Classifications

This method is used to return a list of the classifications in the system.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| q_id | Department ID | | | If specified, include department applicability data |
| applicable | Applicable | | | When used with q_id, include only applicable classifications |
| client_selectable | Client Selectable | | | When used with q_id, include only client selectable classifications |
| order_by | Order By | | lft | Field to order results by. |
| direction | Direction | | asc | |
| limit | Limit | | | |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |

**Example Output**

```
{
  "data" : {
    "10" : {
      "active" : "1",
      "applicable" : "1",
      "classification_id" : "10",
      "client_selectable" : "0",
      "lft" : "1",
      "name" : "Shipment Received",
      "parent_id" : "0",
      "rgt" : "2"
    },
  },
  "error_code" : null,
  "error_message" : "",
  "status" : true
}
```

## support.department_get - Get Ticket Departments

This method is used to return the details for a specific department.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| queue | Department ID | Y | | |

**Example Output**

```
{
  "data": {
    "q_id": "2",
    "name": "Billing",
    "subject_match": "1",
    "q_order": "2",
    "purge_after": "0",
    "name_en_US": "Billing",
    "reply_address": "billing@ubersmith.com",
    "reply_template": "",
    "unassign_closed": "0",
    "send_autoreply": "no",
    "users": {
      "1": {
        "name": "Administrator",
        "prefer_lang": "1",
        "last_login": "1312306226",
        "config": "a:3:{i:1;a:2:{s:5:\"email\";s:17:\"support@ubersmith.com\";s:9:\"signature\";s:31:\"Thank
you,\r\n\r\nDefault Signature\";}i:2;a:2:{s:5:\"email\";s:17:\"support@ubersmith.com\";s:9:\"signature\";s:27:\"Thank
You\r\n\r\nTest Signature\";}i:6;a:2:{s:5:\"email\";s:17:\"support@ubersmith.com\";s:9:\"signature\";s:0:\"\";}}",
        "username": "admin",
        "signature": "",
```

```
            "current_login": "1312309730",
            "password_changed": "1269009589",
            "id": "1",
            "name_convention": "",
            "password_timeout": "0",
            "access": "a:20:{s:6:\"queues\";a:5:{i:1;i:1;i:2;i:1;i:3;i:1;i:4;i:1;i:5;i:1;}s:9:\"devicemgr\";s:4:\"full\";s:14:\"rep
orts_access\";s:4:\"full\";s:8:\"ordermgr\";s:4:\"full\";s:3:\"api\";b:1;s:10:\"all_brands\";s:4:\"full\";s:10:\"supportmgr\";s
:4:\"full\";s:15:\"global_settings\";s:4:\"full\";s:11:\"cm_settings\";s:4:\"full\";s:11:\"sm_settings\";s:4:\"full\";s:11:\"dm
_settings\";s:4:\"full\";s:11:\"ci_settings\";s:4:\"full\";s:17:\"salesmgr_settings\";s:4:\"full\";s:11:\"om_settings\";s:4:\"f
ull\";s:19:\"client_contact_info\";s:4:\"full\";s:15:\"client_services\";s:4:\"full\";s:14:\"client_billing\";s:4:\"full\";s:15
:\"client_acctmgmt\";s:4:\"full\";s:15:\"client_comments\";s:4:\"full\";s:14:\"client_reports\";s:4:\"full\";}",
            "email": "support@ubersmith.com"
        }
    },
    "time_required": "0",
    "name_en_GB": "",
    "support_staff_address": "support@ubersmith.com",
    "post_script": "",
    "submit_emergency": "0"
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## support.department_list - List Ticket Departments

This method is used to return a list of the ticket departments in the system.

**Example Output**

```
{
  "data": {
    "1": {
      "subject_match": "1",
      "q_order": "1",
      "name": "General Support",
      "q_id": "1",
      "purge_after": "0",
      "name_en_US": "General Support",
      "name_en_GB": "General Support",
      "name_zh_CN": "General Support",
      "reply_address": "support@ubersmith.com",
      "reply_template": "this is a support response.\r\n\r\n##ticketlink##",
      "unassign_closed": "1",
      "client": true,
      "send_autoreply": "no",
      "time_required": "0",
      "support_staff_address": "",
      "post_script": "",
      "submit_emergency": "1"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## support.resolution_list - List Resolutions

This method is used to return a list of the resolutions in the system.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| q_id | Department ID | | | If specified, include department applicability data |
| applicable | Applicable | | | When used with q_id, include only applicable resolutions |
| order_by | Order By | | lft | Field to order results by. |
| direction | Direction | | asc | |
| limit | Limit | | | |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |

**Example Output**

```
{
  "data" : {
    "1":{
      "ticket_resolution_id":"1",
      "resolution_type":"1",
      "priority":"8",
      "code":"FIXED",
      "name":"Fixed",
      "applicable":"1"
    },
    "2":{
      "ticket_resolution_id":"2",
      "resolution_type":"2",
      "priority":"12",
      "code":"INVL",
      "name":"Invalid",
      "applicable":"1"
    },
    "3":{
      "ticket_resolution_id":"3",
      "resolution_type":"3",
      "priority":"13",
      "code":"DUPL",
      "name":"Duplicate",
      "applicable":"1"
    }
  },
  "error_code" : null,
  "error_message" : "",
  "status" : true
}
```

## support.ticket_count - Count Support Tickets

This method is used to count the number of support tickets in the system.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| client_id | Client Login | | | Client's login name or ID. |
| queue | Ticket Department ID | | | |
| type | Ticket Type | | | Possible types:<br>Open<br>On Hold<br>Closed<br>Deleted<br>ClientAll (Open, On Hold & Closed) |
| priority | Ticket Priority | | | Possible priorities:<br>0 = Low<br>1 = Normal<br>2 = High<br>3 = 911 |
| assigned | Assigned admin ID | | | |
| contact_id | Contact ID | | | |
| service_id | Service ID | | | |
| opportunity_id | Opportunity ID | | | |
| device_id | Device ID | | | |
| order_id | Order ID | | | |
| client_viewable | Client Viewable | | 1 | If specified, count only records that match the client_viewable setting. Possible values:<br>0: Limit results to tickets that are not client viewable<br>1: Limit results to tickets that are client viewable (default)<br>2: Include all tickets in results |

**Example Output**

```
{
  "data": "218",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## support.ticket_get - Get Support Ticket Details

This method is used to get the details of a specified support ticket.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| ticket_id | Ticket ID | Y | | |

**Example Output**

```
{
  "data": {
    "cc": "",
    "client_replies": "2",
    "q_id": "1",
    "activity_type_name": "Client Followup",
    "author": "Johnny Ubersmith <uber@ubersmith.com>",
    "timestamp": "1272986228",
    "assignment": "0",
    "admin_name": "John Admin Doe",
    "admin_email": "jdoe@ubersmith.com",
    "admin_username": "jdoe",
    "order_id": "1129",
    "metadata": {
      "billable": ""
    },
    "weight": "0",
    "client_activity_type": "11",
    "activity": "1294173475",
    "contact_id": "0",
    "priority": "2",
    "subject": "test ticket",
    "recipient": "",
    "client_activity": "1294173475",
    "service_id": "1271",
    "ticket_id": "5141",
    "priority_name": "High",
    "priority_id": "2",
    "q_name": "General Support",
    "device_id": "14",
    "client_id": "1005",
    "type": "1",
    "type_id": "1",
    "type_name": "Open",
    "reply_to": "",
    "client_activity_type_name": "Client Followup",
    "admin_name": null,
    "source_id": "1",
    "source_name": "client",
    "source": "1",
    "opportunity_id": "0",
    "replies": "2",
    "activity_type": "11"
    "notification_id": "83"
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## support.ticket_list - Get a List of Tickets

This method is used to return a list of tickets.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| client_id | Client Login | | | Client's login name or ID. |
| queue | Ticket Department ID | | | |
| type | Ticket Type | | | Possible types:<br>Open,<br>On Hold,<br>Closed,<br>Deleted,<br>ClientAll (meta-type meaning any of: Open, On Hold, or Closed),<br>Any numeric type as listed by the 'support.ticket_type_list' method is also supported. |
| priority | Ticket Priority | | | Possible priorities:<br>0 = Low<br>1 = Normal<br>2 = High<br>3 = 911 |
| impact | Ticket Impact | | | Possible impacts:<br>0 = Minor/Localized<br>1 = Moderate/Limited<br>2 = Significant/Large<br>3 = Extensive/Widespread |
| assigned | Assigned admin ID | | | |
| contact_id | Contact ID | | | |
| service_id | Service ID | | | |
| opportunity_id | Opportunity ID | | | |
| classification | Classification | | | Include only tickets with this classification or a sub-classifications |
| classification_id | Classification (Exact) | | | Include only tickets with this classification |
| device_id | Device ID | | | |
| order_id | Order ID | | | |
| begin | Beginning Timestamp | | | If specified, return only records that fall after this timestamp. |
| end | End Timestamp | | | If specified, return only records that fall before this timestamp. |
| activity_begin | Activity Begin Timestamp | | | If specified, return only records that fall after this timestamp. |
| activity_end | Activity End Timestamp | | | If specified, return only records that fall before this timestamp. |
| client_activity_begin | Client Activity Begin Timestamp | | | If specified, return only records that fall after this timestamp. |
| client_activity_end | Client Activity End Timestamp | | | If specified, return only records that fall before this timestamp. |
| client_viewable | Client Viewable | | 1 | If specified, return only records that match the client_viewable setting. Possible values:<br>0: Limit results to tickets that are not client viewable<br>1: Limit results to tickets that are client viewable (default)<br>2: Include all tickets in results |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| metadata | Include Metadata | | 0 | Possible priorities:<br>0 or blank: Include fields with show_in_list flag set<br> 1: Include all metadata fields<br> array: Specify individual fields to include by entering "metadata[field_name1]=1 metadata[field_name2]=1 etc..." |
| | Metadata Filters | | | Filter list by custom field values, ie. "myfield1=somevalue myfield2=somevalue etc..." |
| order_by | Order By | | | Field to order results by. Possible values: ticket_id ,contact_id, author, recipient, subject, timestamp, type, source, assignment, activity, cc, priority, q_id, q_name, weight, activity_type |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "5048": {
      "cc": "",
      "client_replies": "1",
      "q_id": "1",
      "author": "Johnny Ubersmith <uber@ubersmith.com>",
      "timestamp": "1266257592",
      "assignment": "0",
      "admin_name": "John Admin Doe",
      "admin_email": "jdoe@ubersmith.com",
      "admin_username": "jdoe",
      "order_id": "0",
      "weight": "0",
      "client_activity_type": "0",
      "activity": "1266257592",
      "contact_id": "0",
      "priority": "High",
      "priority_name": "High",
      "priority_id": "2",
      "subject": "Server Question",
      "recipient": "support@ubersmith.com",
      "client_activity": "1266257592",
      "service_id": "238",
      "ticket_id": "5048",
      "device_id": "0",
      "client_id": "1005",
      "type": "Open",
      "type_id": "1",
      "type_name": "Open",
      "listed_company": "Ubersmith, Inc.",
      "reply_to": "",
      "source_id": "1",
      "source_name": "client",
      "source": "client",
      "opportunity_id": "0",
      "replies": "1",
      "activity_type": "None",
      "q_name": "General Support"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## support.ticket_merge - Merge Tickets

This method is used to merge two tickets.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| ticket_id | Ticket ID (Main) | Y | | |
| merge_ticket_id | Ticket ID (Sub) | Y | | |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## support.ticket_post_client_response - Post a Client Response to a Ticket

This method is used to post a client response to a ticket, in the same way as if the client had responded via the client interface.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| ticket_id | Ticket ID | Y | | |
| subject | Response Subject | | | |
| body | Response Body | | | |
| contact_id | Contact ID | | | |
| author | Client Email Address | | | |
| cc | CC Address(es) | | | One or more (comma-separated) email addresses to receive a carbon copy of the response. |
| timestamp | Timestamp | | current time | |
| attach[x] | Attachment | | | Up to 5 attachments may be specified as attach[0], attach[1], etc. When using cURL the value for each attach[x] parameter should be @ where is the file you wish to attach. |

**Example Output**

```
{
  "data": "433",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## support.ticket_post_list - Get all Posts for a Ticket

This method is used to return all the posts for the specified ticket.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| ticket_id | Ticket ID | | | Either this or ticket_post_id must be provided. |
| ticket_post_id | Ticket Post ID | | | Either this or ticket_id must be provided. |
| private | Show only client-viewable posts | | 0 | Set to 1 to hide posts which are not client viewable. |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | Set to desc to return posts in descending order (newest first) |
| offset | Offset | | | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Only used if offset is specified |

**Example Output**

```
{
  "data": {
    "432": {
      "cc": "",
      "ticket_post_id": "432",
      "body": "body",
      "author": "Johnny Ubersmith <uber@ubersmith.com>",
      "assignment": "0",
      "timestamp": "1312293776",
      "contact_id": "261",
      "subject": "Subject",
      "recipient": "luke@ubersmith.com",
      "ticket_id": "10092",
      "type": "21",
      "type_id": "21",
      "type_name": "Example Custom Ticket Type",
      "reply_to": "",
      "attachments": "1",
      "source": "2",
      "source_id": "2",
      "source_name": "admin",
      "headers": "",
      "hidden": "0",
      "duration": "3600"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## support.ticket_post_staff_response - Post a Staff Response to a Ticket

This method is used to post a staff response to a ticket.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| ticket_id | Ticket ID | Y | | |
| subject | Response Subject | | Ticket Subject | |
| body | Response Body | | | |
| comment | Comment Only | | 0 | Set to 1 to post a comment |
| recipient | Recipient Email | | | Not required for comments |
| user_id | Admin ID | | | |
| from | Author's Name and/or Email | | Admin's email address | |
| cc | CC Address(es) | | | One or more (comma-separated) email addresses to receive a carbon copy of the response. |
| bcc | BCC Address(es) | | | One or more (comma-separated) email addresses to receive a blind carbon copy of the response. |
| time_spent | Time Spent | | 0 | Time spent in minutes |
| timestamp | Timestamp | | current time | |
| attach[x] | Attachment | | | Up to 5 attachments may be specified as attach[0], attach[1], etc. When using cURL the value for each attach[x] parameter should be @ where is the file you wish to attach. |

**Example Output**

```
{
  "data": "434",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## support.ticket_submit - Submit a New Ticket

This method is used to create a new support ticket.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| body | Ticket Body | Y | | |
| subject | Ticket Subject | Y | | |
| author | Author Email Address | | | |
| name | Author Name | | | |
| cc | Cc Address(es) | | | One or more (comma-separated) email addresses to receive a carbon copy of the response. |
| bcc | Bcc Address(es) | | | One or more (comma-separated) email addresses to receive a blind carbon copy of the response. |
| priority | Ticket Priority | | 1 | 0 = Low, 1= Normal, 2 = High, 3 = 911 |
| impact | Ticket Impact | | 1 | Possible impacts:<br>0 = No Impact<br>1 = Minor/Localized<br>2 = Moderate/Limited<br>3 = Significant/Large<br>4 = Extensive/Widespread |
| queue | Ticket Department ID | | system default | |

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| classification_id | Classification ID | | | |
| ticket_resolution_id | Resolution ID | | | |
| client_id | Client ID | | 0 | |
| contact_id | Contact ID | | 0 | |
| brand_id | Brand ID | | | |
| service_id | Associate Service | | | The input should be a service ID |
| device_id | Associate Device | | | The input should be a device ID |
| timestamp | Timestamp | | current time | |
| meta_* | Custom Data | | | Metadata items may be set by passing the values as additional parameters matching the metadata items API field names prefixed with 'meta_'. |
| attach[x] | Attachment | | | Up to 5 attachments may be specified as attach[0], attach[1], etc. When using cURL the value for each attach[x] parameter should be @ where is the file you wish to attach. |
| client_viewable | Client Viewable | | 1 | Set to 0 to mark ticket as not client-viewable, 1 to mark as viewable |

**Example Output**

```
{
  "data": "10093",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## support.ticket_submit_outgoing - Create a New Outgoing Ticket

This method is used to create a new outgoing support ticket.

| Parameter | Name | Required | Default | Description |
|-----------|------|----------|---------|-------------|
| body | Ticket Body | Y | | |
| subject | Ticket Subject | | | |
| recipient | Recipient Email | Y | | |
| user_id | Admin ID | | | |
| author | Author Email Address | | | |
| cc | Cc Address(es) | | | One or more (comma-separated) email addresses to receive a carbon copy of the response. |
| bcc | Bcc Address(es) | | | One or more (comma-separated) email addresses to receive a blind carbon copy of the response. |
| priority | Ticket Priority | | 1 | 0 = Low, 1= Normal, 2 = High, 3 = 911 |
| impact | Ticket Impact | | 1 | Possible impacts:<br>0 = No Impact<br>1 = Minor/Localized<br>2 = Moderate/Limited<br>3 = Significant/Large<br>4 = Extensive/Widespread |
| queue | Ticket Department ID | | system default | |
| classification_id | Classification ID | | | |
| client_id | Client ID | | 0 | |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| contact_id | Contact ID | | 0 | |
| brand_id | Brand ID | | | |
| service_id | Associate Service | | | The input should be a service ID |
| device_id | Associate Device | | | The input should be a device ID |
| timestamp | Timestamp | | current time | |
| client_viewable | Client Viewable | | | 0 = Not Client Viewable, 1 = Client Viewable |
| meta_* | Custom Data | | | Metadata items may be set by passing the values as additional parameters matching the metadata items API field names prefixed with 'meta_'. |
| attach[x] | Attachment | | | Up to 5 attachments may be specified as attach[0], attach[1], etc. When using cURL the value for each attach[x] parameter should be @ where is the file you wish to attach. |

**Example Output**

```
{
  "data": "10094",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## support.ticket_time_update - Update Time Spent on a Ticket

This method is used to update time spent on a ticket.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| ticket_post_id | Ticket Post ID | Y | | |
| time_spent | Time Spent | Y | | Time spent in minutes |
| user_id | User ID | | | |
| billable | Billable | | | If billing based on ticket posts, set to the value of the billing tier |

**Example Output**

```
{
  "status": true,
  "error_code": null,
  "error_message": "",
  "data": true
}
```

## support.ticket_timer_add - Add a New Ticket Timer

This method is used to add a new ticket timer to a ticket.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| ticket_id | Ticket ID | Y | | |
| ts | Timestamp | Y | | Time when the ticket will be changed. |
| type | Ticket Type | | | Possible types:<br>1 = Open,<br>2 = On Hold,<br>3 = Closed,<br>4 = Deleted<br>Either the name or the numeric value of the type is supported. The numeric value of types listed under the 'support.ticket_type_list' method is also supported. |
| priority | Ticket Priority | | | Possible priorities:<br>0 = Low<br>1 = Normal<br>2 = High<br>3 = 911<br>Either the name or the numeric value of the priority is supported. |
| q_id | Ticket Department ID | | | |
| assignment | Assign to Admin | | | Either the admin ID or username can be provided. |
| auto_remove | Auto Remove on Reply | | | Determines whether the ticket timer is removed when a staff member responds to the ticket. Possible values:<br>1 = Yes<br>0 = No |
| respond | Update Type | | | Type of response to add to the ticket once the ticket timer expires.<br>Possible types:<br>21 = Staff Followup<br>22 = Staff Comment (Client Viewable)<br>23 = Staff Comment (Private)<br>Only the numeric value of the type is supported. |
| respond_from | Update From | | | From email address to set on the response added to the ticket once the ticket timer expires. |
| respond_subject | Update Subject | | | Subject of the response to add to the ticket once the ticket timer expires. |
| respond_body | Update Body | | | Body of the response to add to the ticket once the ticket timer expires. |
| notify_email | Notification To | | | Email address the 'ticket timer has expired' notification email will be sent to. |
| notify_subject | Notification Subject | | | Subject of the 'ticket timer has expired' notification email. |
| notify_body | Notification Body | | | Body of the 'ticket timer has expired' notification email. |

**Example Output**

```
{
  "data": "16",
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## support.ticket_timer_delete - Delete a Ticket Timer

This method is used to delete a ticket timer.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| ticket_timer_id | Ticket Timer ID | Y | | |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## support.ticket_timer_list - Get a List of Ticket Timers

This method is used to return a list of ticket timers.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| ticket_id | Ticket ID | Y | | |
| type | Ticket Type | | | Possible types:<br>1 = Open,<br>2 = On Hold,<br>3 = Closed,<br>4 = Deleted<br>ClientAll (meta-type meaning any of: Open, On Hold, or Closed)<br>Either the name or the numeric value of the type is supported. The numeric value of types listed under the 'support.ticket_type_list' method is also supported. |
| priority | Ticket Priority | | | Possible priorities:<br>0 = Low<br>1 = Normal<br>2 = High<br>3 = 911<br>Either the name or the numeric value of the priority is supported. |
| q_id | Ticket Department ID | | | |
| assignment | Assign to Admin | | | Either the admin ID or username can be provided. |
| auto_remove | Auto Remove on Reply | | | Determines whether the ticket timer is removed when a staff member responds to the ticket. Possible values:<br>1 = Yes<br>0 = No |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "16": {
      "ticket_timer_id": "16",
      "ticket_id": "64",
      "ts": "1387040400",
      "type": "1",
      "priority": "1",
      "q_id": "4",
```

```
            "assignment": "2",
            "auto_remove": "1",
            "respond": "21",
            "respond_from": "Ubersmith <support@ubersmith.com>",
            "respond_subject": "Ticket updated",
            "respond_body": "The department on ticket ##ticket_id## has been changed to ##department##.",
            "notify_email": "Ubersmith <support@ubersmith.com>",
            "notify_subject": "Ticket timer expired",
            "notify_body": "The pending timer on ticket ##ticket_id## has expired."
        }
    },
    "error_message": "",
    "error_code": null,
    "status": true
}
```

## support.ticket_timer_update - Update a Ticket Timer

This method is used to update a ticket timer.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| ticket_timer_id | Ticket Timer ID | Y | | |
| ts | Timestamp | | | Time when the ticket will be changed. |
| type | Ticket Type | | | Possible types:<br>1 = Open,<br>2 = On Hold,<br>3 = Closed,<br>4 = Deleted<br>Either the name or the numeric value of the type is supported. The numeric value of types listed under the 'support.ticket_type_list' method is also supported. |
| priority | Ticket Priority | | | Possible priorities:<br>0 = Low<br>1 = Normal<br>2 = High<br>3 = 911<br>Either the name or the numeric value of the priority is supported. |
| q_id | Ticket Department ID | | | |
| assignment | Assign to Admin | | | Either the admin ID or username can be provided. |
| auto_remove | Auto Remove on Reply | | | Determines whether the ticket timer is removed when a staff member responds to the ticket. Possible values:<br>1 = Yes<br>0 = No |
| respond | Update Type | | | Type of response to add to the ticket once the ticket timer expires.<br>Possible types:<br>21 = Staff Followup<br>22 = Staff Comment (Client Viewable)<br>23 = Staff Comment (Private)<br>Only the numeric value of the type is supported. |
| respond_from | Update From | | | From email address to set on the response added to the ticket once the ticket timer expires. |
| respond_subject | Update Subject | | | Subject of the response to add to the ticket once the ticket timer expires. |
| respond_body | Update Body | | | Body of the response to add to the ticket once the ticket timer expires. |
| notify_email | Notification To | | | Email address the 'ticket timer has expired' notification email will be sent to. |

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| notify_subject | Notification Subject | | | Subject of the 'ticket timer has expired' notification email. |
| notify_body | Notification Body | | | Body of the 'ticket timer has expired' notification email. |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## support.ticket_type_list - Get a List of Ticket Types

This method is used to return a list of ticket types. If the q_id parameter is specified the output will include a client_selectable element indicating whether clients are allowed to select each type in the specific department.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| queue | Ticket Department ID | | | |
| order_by | Order By | | | Field to order results by |
| direction | Direction | | asc | asc = Ascending Order<br>desc = Descending Order |
| offset | Offset | | 0 | If specified, return records starting from this offset (0-based). |
| limit | Limit | | | Maximum number of rows to return |

**Example Output**

```
{
  "data": {
    "1":{
      "ticket_type_id":"1",
      "status":"1",
      "priority":"1",
      "on_reply":"0",
      "reply_type":"1",
      "name":"Open"
    },
    "2":{
      "ticket_type_id":"2",
      "status":"2",
      "priority":"3",
      "on_reply":"0",
      "reply_type":"1",
      "name":"On Hold"
    },
    "3":{
      "ticket_type_id":"3",
      "status":"3",
      "priority":"4",
      "on_reply":"0",
      "reply_type":"1",
      "name":"Closed"
    },
    "4":{
      "ticket_type_id":"4",
      "status":"4",
      "priority":"5",
      "on_reply":"0",
      "reply_type":"1",
      "name":"Deleted"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

## support.ticket_update - Update a Ticket

This method is used to update the details of a ticket.

| Parameter | Name | Required | Default | Description |
|---|---|---|---|---|
| ticket_id | Ticket ID | Y | | |
| assignment | Assigned User ID | | | |
| queue | Ticket Department ID | | | |
| type | Ticket Type | | | 1 or open = Open, 2 or onhold = On Hold, 3 or closed = Closed, 4 or deleted = Deleted. Any numeric type as listed by the 'support.ticket_type_list' method is also supported.) |
| subject | Ticket Subject | | | |
| author | Ticket Author Email Address | | | |
| cc | Cc Address(es) | | | One or more (comma-separated) email addresses to receive a carbon copy of the response. |
| bcc | Bcc Address(es) | | | One or more (comma-separated) email addresses to receive a blind carbon copy of the response. |
| priority | Ticket Priority | | | 0 = Low, 1= Normal, 2 = High, 3 = 911 |
| impact | Ticket Impact | | | Possible impacts:<br>0 = No Impact<br>1 = Minor/Localized<br>2 = Moderate/Limited<br>3 = Significant/Large<br>4 = Extensive/Widespread |
| classification_id | Classification ID | | | |
| ticket_resolution_id | Resolution ID | | | |
| weight | Ticket Rating | | | Number of stars 0-5 |
| client_id | Associate Client | | | The input should be a client ID |
| contact_id | Associate Contact | | | The input should be a contact ID |
| service_id | Associate Service | | | The input should be a service ID |
| device_id | Associate Device | | | The input should be a device ID |
| opportunity_id | Opportunity ID | | | The input should be an opportunity ID |
| order_id | Order ID | | | The input should be an order ID |
| meta_* | Custom Data | | | Metadata items may be set by passing the values as additional parameters matching the metadata items API field names prefixed with 'meta_'. |
| client_viewable | Client Viewable | | 1 | Set to 0 to mark ticket as not client-viewable, 1 to mark as viewable |

**Example Output**

```
{
  "data": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```