Table Of Contents

Introduction 4
API Sections 4
API Authentication 4
Using the Ubersmith API 4
API Clients 5
uber - Ubersmith 6
uber.api_export - Export Data 6
uber.attachment_get - Get an attachment 6
uber.attachment_list - List Attachments 6
uber.check_login - Verify a login and password 7
uber.client_welcome_stats - Display Client Statistics 7
uber.comment_add - Add Comment 8
uber.comment_delete - Delete Comment 9
uber.comment_get - Get Comments 9
uber.comment_list - List Comments 10
uber.comment_update - Update Comment 10
uber.documentation - Download API Documentation 11
uber.event_list - Access the Event Log 11
uber.forgot_pass - Send a Password Reminder 12
uber.login_list - List User Logins 13
uber.mail_get - Get an Email From the Log 13
uber.mail_list - Access the Mail Log 14
uber.message_list - List Message Board Messages 15
uber.metadata_bulk_get - Bulk Get Metadata Values 16
uber.metadata_get - Get Metadata Values 16
uber.method_get - Get API Method Details 17
uber.method_list - List Available API Methods 18
uber.quick_stats - Get Quick System Stats 19
uber.quick_stats_detail - Get Detailed System Stats 19
uber.service_plan_get - Get Service Plan Details 20
uber.service_plan_list - List Service Plans 22
uber.user_exists - Check whether a Client Exists 24
uber.username_exists - Check Whether a Username Exists 24
client - Client 26
client.ach_add - Add a New Bank Account 26
client.ach_delete - Delete a Bank Account 26
client.ach_update - Update a Bank Account 27
client.add - Add a New Client 27
client.cc_add - Add a New Credit Card 29
client.cc_delete - Delete a Credit Card 30
client.cc_update - Update a Credit Card 30
client.comment_list - List a Client's Comments 31
client.contact_add - Add a New Contact 32
client.contact_delete - Delete a Contact 32
client.contact_get - Get Contact Details 33
client.contact_list - List a Client's Contacts 33
client.contact_update - Update a Contact 34
client.count - Count Active Clients 35
client.credit_add - Add an Account Credit 36
client.credit_comment_list - List a Credit's Comments 36
client.credit_deactivate - Deactivate an Account Credit 37

client.credit_list - List a Client's Credits 37
client.deactivate - Deactivate a Client 38
client.domain_add - Add a Domain 38
client.domain_list - List a Client's Domains 39
client.domain_lookup - Look Up a Domain 40
client.domain_register - Register a Domain 41
client.domain_transfer - Transfer a Domain 43
client.get - Get Client Details 46
client.invoice_charge - Charge an Invoice 47
client.invoice_count - Count Invoices 49
client.invoice_disregard - Disregard an Invoice 49
client.invoice_generate - Generate an Invoice 49
client.invoice_get - Get an Invoice 50
client.invoice_list - List a Client's Invoices 53
client.invoice_payments - List an Invoice's Payments 54
client.invoice_post_gw_payment - Record a Payment 54
client.latest_client - Get the Latest Client 55
client.list - List Clients 56
client.lookup - Look Up a Client 57
client.metadata_get - Get a Client's Metadata 58
client.metadata_single - Get a Client's Metadata Value 59
client.payment_method_list - List a Client's Payment Methods 59
client.reactivate - Reactivate a Client 60
client.renewal_list - List Services for Renewal 60
client.send_welcome - Send a Welcome Letter 61
client.service_add - Add a New Service 62
client.service_comment_list - List a Service's Comments 64
client.service_deactivate - Deactivate a Service 65
client.service_get - Get a Service 65
client.service_list - List a Client's Services 66
client.service_metadata_get - Get a Service's Metadata 68
client.service_metadata_single - Get a Service's Metadata Value 69
client.service_module_call - Call a Service Module Function 70
client.service_prorate - Prorate a Service 70
client.service_update - Update a Service 71
client.set_login - Set a Client's Login 73
client.update - Update a Client 73
device - Device 76
device.add - Add a New Device 76
device.comment_list - List a Device's Comments 76
device.cpanel_add - Add a cPanel Account 77
device.delete - Delete a Device 77
device.event_list - List Device Events 78
device.facility_list - List Device Facilities 79
device.get - Get a Device 81
device.hostname_get - Get a Device Hostname 82
device.ip_assign - Assign an IP to a Device 83
device.ip_assignment_add - Create a New IP Assignment 84
device.ip_assignment_delete - Delete a Device IP Assignment 85
device.ip_assignment_list - List Device IP Assignments 86
device.ip_assignment_update - Update a Device IP Assignment 87
device.ip_get_available - List Available Device IPs 88
device.ip_get_unassigned - Get Unassigned Device IPs 88
device.ip_group_add - Add a Device IP Group 89
device.ip_group_delete - Delete a Device IP Group 89

device.ip_group_list - List a Device IP Group 90
device.ip_group_update - Update a Device IP Group 91
device.ip_lookup - Look Up a Device IP 91
device.ip_unassign - Unassign a Device IP 92
device.list - List Devices 93
device.module_call - Call a Device Module Function 95
device.module_call_aggregate - Call an Aggregate Device Module Function 95
device.module_graph - Generate Device Module Graph 96
device.monitor_add - Add a New Device Monitor 96
device.monitor_delete - Delete a Device Monitor 97
device.monitor_disable - Disable a Device Monitor 97
device.monitor_enable - Enable a Device Monitor 98
device.monitor_list - List Device Monitors 98
device.monitor_update - Update a Device Monitor 100
device.reboot - Set a Device's Power State 100
device.reboot_graph - Get a Reboot Graph 101
device.tag - Tag a Device 101
device.type_list - List Device Types 102
device.untag - Untag a Device 102
device.update - Update a Device 103
device.vlan_get_available - List Available VLANs 104

order - Order 105

order.cancel - Cancel an Order 105
order.client_respond - Post a Client/Lead Order Response 105
order.coupon_get - Get Order Coupon Details 105
order.create - Create a New Order 106
order.get - Get Order Details 109
order.list - List Orders 110
order.process - Process an Order 111
order.queue_list - List Order Queues 112
order.respond - Post an Order Response 113
order.submit - Submit An Order 113
order.update - Update an Order 114

sales - Sales 115

sales.opportunity_add - Add an Opportunity 115
sales.opportunity_list - List Opportunities 115
sales.opportunity_stage_list - List Opportunity Stages 117
sales.opportunity_status_list - List Opportunity Statuses 117
sales.opportunity_type_list - List Opportunity Types 117
sales.opportunity_update - Update an Opportunity 118

support - Support 120

support.department_get - Get Ticket Departments 120
support.department_list - List Ticket Departments 120
support.ticket_count - Count Support Tickets 121
support.ticket_get - Get Support Ticket Details 122
support.ticket_list - Get a List of Tickets 123
support.ticket_merge - Merge Tickets 125
support.ticket_post_client_response - Post a Client Response to a Ticket 125
support.ticket_post_list - Get all Posts for a Ticket 126
support.ticket_post_staff_response - Post a Staff Response to a Ticket 126
support.ticket_submit - Submit a New Ticket 127
support.ticket_submit_outgoing - Create a New Outgoing Ticket 128
support.ticket_update - Update a Ticket 129

Introduction

The Ubersmith API 2.0 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 2.0 release of 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 return 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 are passed to the API as either GET parameters or via standard HTTP POST requests.

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 2.0.

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.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.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
id	ID of ticket, comment, etc.	Y		

Example Output

```
{
  "data": {
    "1946": {
      "size": "10276",
      "attach_id": "1946",
      "conttype": "application/pdf",
      "filename": "Invoice-1251-411.pdf"
    }
  },
  "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",
    "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_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 or opportunity
ref_id	Related ID	Y		ID number of client, service, device, credit or opportunity
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 or opportunity
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 or opportunity
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 or opportunity
ref_id	Related ID	Y		ID number of client, service, device, credit or opportunity

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 or opportunity
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.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.

Parameter	Name	Required	Default	Description
ledger	Ledger events only		0	Return all events or just those that appear in the ledger event log.
offset	Offset		0	If specified, return records starting from this offset (0-based).
limit	Limit		100	
order_by	Order By		time	
direction	Direction		asc	

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.forgot_pass - Send a Password Reminder

This method is used to send a password reminder, the reminder will be send 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.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 ",
    "time": "1312231187",
    "recipient": "\"Testes Client\" ",
    "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.
offset	Offset		0	If specified, return records starting from this offset (0-based).
limit	Limit		100	
order_by	Order By		time	
direction	Direction		asc	

Parameter	Name	Required	Default	Description
all	Include Message Body		0	Include full email body.

Example Output

```
{
  "data": {
    "99": {
      "header": "From: uber-test ",
      "time": "1242317909",
      "recipient": "Test Client ",
      "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	

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 or package.

Parameter	Name	Required	Default	Description
variable	Metadata Variable Name	Y		
field_type	Field Type	Y	pack	Possible values: client, pack

Example Output

```
{
  "data": {
    "1092": "",
    "1039": ""
  }
  "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.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.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

Parameter	Name	Required	Default	Description
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.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",
    "api_include": "1",
    "period": "1",
    "metadata": {
      "11": {
        "required": "0",
        "size": "0",
        "prefix": "Included Transfer Unit",
        "metagroup_id": "4",
        "editable": "0",
        "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"
      }
    },
    "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": "",
    "hide_upgrades": "0",
    "service_host": "",
    "qbaccount": "Sales - Software",
    "welcomesub": "",
    "active": "1",
    "plan_id": "11"
  },
  "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			

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",
      "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.",
      "hide_upgrades": "0",
      "service_host": "",
      "qbaccount": "Sales - Software",
      "welcomesub": "Welcome",
      "active": "1",
      "plan_id": "33"
    }
  },
  "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
				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			

Example Output

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

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
user_login	Client Login	Y		Client's login name or ID.
billing_info_id	Bank Account ID			Account to update. If not specified, the system will check for an account with the same last 4 digits.
ach_acct	Account Number	Y		
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			

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 'daysdue' days after generation) 1 = Static Due Date (invoice is due on 'datedue' day of the month, only applicable to monthly invoicing)
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.
daysdue	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.
active	Client Status		1	1 = Client, 2 = Lead
late_fee_scheme_id	Late Fee Schedule		Default late fee schedule	

Parameter	Name	Required	Default	Description
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. 0 = Low 1 = Normal 2 = High 3 = 911
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. 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. 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_'.

Example Output

```
{
  "data": "1264",
  "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		
cc_expire	Expiration Date			MMYY format

Parameter	Name	Required	Default	Description
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			

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_update - Update a Credit Card

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

Parameter	Name	Required	Default	Description
user_login	Client Login	Y		Client's login name or ID.
billing_info_id	Credit Card ID			Credit card to update. If not specified, the system will check for an account with the same last 4 digits.
cc_num	Credit Card Number	Y		
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			

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		

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).

Example Output

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

client.contact_delete - Delete a Contact

This method is used to delete a contact from a client's account.

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

Example Output

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

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			

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": [

    ],
    "password": "{ssha1}5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8",
    "login": "j_uber",
    "email": "tommy@willsmythe.com"
  },
  "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	Y		Client's login name or ID.

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"
    }
  },
  "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 be left blank and the contact will not be able to log in.
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).

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 active clients in the system.

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
user_login	Client Login	Y		Client's login name or 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:
				check = Received Check
				cash = Received Cash
				comp = Company Issued
				cred = Credit card charge
				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.

Example Output

```
{
  "data": "54",
  "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		

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_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		
csv	CSV Output Flag		0	Set to 1 to return details in CSV format.

Example Output

```
{
  "data": {
    "55": {
      "payment_number": "",
      "comments": "",
      "order_id": "0",
      "credit_id": "55",
      "amount": "-10.00",
      "date": "1311739200",
      "reason": "Courtesy",
      "clientid": "1263",
      "active": "1",
      "balance": "-10.00",
      "payment_type": "comp"
    }
  },
  "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.
password	Password			Password for domain.
registrar	Domain Registrar	Y		Supported registrar, possible values: enom opensrs manual

Example Output

```
{
  "data": true,
  "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 1 = All Domains

Parameter	Name	Required	Default	Description
order_by	Order By		name	Field to order results by, available fields: name registered expires registrar

direction	Direction		asc	asc = Ascending
				desc = Descending

limit	Limit		0	

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: owner 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: owner 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			

Parameter	Name	Required	Default	Description
billing_phone	Billing Telephone Number			
billing_fax	Billing Fax Number			
technical_info	Technical Information		custom	Possible Values: owner custom
technical_fname	Technical First Name			
technical_lname	Technical Last Name			
technical_company	Technical Company			
technical_email	Technical Email			
technical_address1	Technical Address 1			
technical_address2	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

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	
admin_info	Administrator's Information		custom	Possible Values: owner 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			

Parameter	Name	Required	Default	Description
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: owner 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: owner custom
technical_fname	Technical First Name			
technical_lname	Technical Last Name			
technical_company	Technical Company			

Parameter	Name	Required	Default	Description
technical_email	Technical Email			
technical_address1	Technical Address 1			
technical_address2	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.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.
csv	CSV Output Flag		0	Set to 1 to return details in CSV format.

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":"a:1:{s:4:\"cbms\";s:6:\"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":"0",
      "late_fee_scheme_id":"0",
      "acctmgr":"0",
      "name_convention":"",
      "inv_balance":"187,559.67"
  }
}
```

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.
				cc = Credit Card
				ach = Bank Account

billing_info_id	Payment Method ID			
paid_in_full	Paid In Full			
package	Packages			

Parameter	Name	Required	Default	Description
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			
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:
				json = JSON (default)
				pdf = PDF
				html = HTML
				xml = XML

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 1 = Paid 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
				desc = Descending Order

limit	Limit			
offset	Offset		0	If specified, return records starting from this offset (0-based).

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"
    }
  },
  "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: paypal worldpay 2checkout 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. 1 = Inactive 0 = Active 2 = All
active	Status			1 = Clients 2 = Leads 0 = Deactivated
qblistid	QuickBooks ID			If specified, only the client which matches the specified QuickBooks ID will be returned.
order_by	Order By		client_id	Field to order results by

direction	Direction		asc	asc = Ascending Order
				desc = Descending Order

limit	Limit			
offset	Offset		0	If specified, return records starting from this offset (0-based).

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": "0",
      "sales": "0",
      "discount_type": "0",
      "grace_due": "0",
      "metadata": "",
      "tier_sales": "0",
      "commission_rate": "0.00",
      "class_id": "1",
      "priority": "1",
      "datesend": "18",
      "last": "Smith",
      "late_fee_scheme_id": "0",
      "password_changed": "1310566873",
      "latest_inv": "1276016884",
      "charge_days": "0",
      "permnote": null,
      "datepay": "16",
      "name_convention": "0",
      "acctmgr": "0",
      "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 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.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		

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"
    }
  },
  "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.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",
      "cp_host": "0",
      "desserv": "Pre-start (not BiA)",
      "clientid": "1081",
      "hide_upgrades": "0",
      "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	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.

Parameter	Name	Required	Default	Description
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. 0 = By Period 1 = By Month 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 2 = Pending 3 = Suspended 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: 0 = percentage 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.
start	Start Date		current date	When the service should start.

Parameter	Name	Required	Default	Description
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. If the service is using cPanel, this is the cPanel username of the account. If not, it will only be used as a storage field.
pass	Service Password			A password for the service. If the service is using cPanel, this is the cPanel password of the account. If not, it will only be used as a storage field.
server	Service Domain or Server Name			A domain or server name for the service. If the service is using cPanel, this is the domain name of the account. If not, it will only be used as a storage field.
ip_address	Service IP Address			An IP address for the service. It is NOT used in the cPanel integration. This is only a storage field.
options	Service Options Array			An array of options which should be enabled for the service.
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_'.
zone_id	Zone ID			Assigned location for reporting purposes.

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		

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.

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,
    "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": "",
    "taxes": [

    ],
    "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",
    "cp_host": "0",
    "desserv": "Price Adjustment",
    "clientid": "1005",
    "hide_upgrades": "0",
    "password": "",
    "options": [

    ],
    "plan_id": "44",
    "created": "1312229764",
    "dev": "0",
    "active": "1",
    "ip_address": "",
    "balance": "0.00"
  },
  "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	Y		
parentpack	Parent Service ID			
period	Period			
pack_type_select	Service Selection		0	0 = all active recurring services and unpaid one-time services 1 = active recurring services 2 = active one-time services 3 = all services 4 = all active services
metadata	Include Metadata			Set to 1 to include service metadata in output.
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.

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,
      "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",
      "cp_host": "0",
      "desserv": "DELETE",
      "clientid": "1011",
      "hide_upgrades": "0",
      "password": "",
      "plan_id": "27",
      "created": "1265309281",
      "active": "1",
      "ip_address": "",
      "balance": "0.00"
    }
  },
  "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_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.

Example Output

```
{
  "data": "0.00",
  "error_message": "",
  "error_code": null,
  "status": 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. 0 = By Period 1 = By Month 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 2 = Pending 3 = Suspended 4 = Cancelled

Parameter	Name	Required	Default	Description
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: 0 = percentage 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.
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.
start	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. If the service is using cPanel, this is the cPanel username of the account. If not, it will only be used as a storage field.
pass	Service Password			A password for the service. If the service is using cPanel, this is the cPanel password of the account. If not, it will only be used as a storage field.
server	Service Domain or Server Name			A domain or server name for the service. If the service is using cPanel, this is the domain name of the account. If not, it will only be used as a storage field.
ip_address	Service IP Address			An IP address for the service. It is NOT used in the cPanel integration. This is only a storage field.
cp_host	cPanel Host ID			
options	Service Options Array			An array of options which should be enabled for the service.

Parameter	Name	Required	Default	Description
taxes	Taxes			An array of tax ids which should be enabled for the service.
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.

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.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		

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			
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.
grace_due	Due Date Method		0	Method used to calculate invoice due date: 0 = Grace Period (invoice is due 'daysdue' days after generation) 1 = Static Due Date (invoice is due on 'datedue' day of the month, only applicable to monthly invoicing)
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. If datedue is less than or equal to datesend, the invoice will be due on that day of the following month.
daysdue	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: 0 = percentage 1 = dollar value
referred	Referred By			This field stores whoever may have referred the client.
active	Client Status		1	1 = Client, 2 = Lead
late_fee_scheme_id	Late Fee Schedule		Default late fee schedule	
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.

Parameter	Name	Required	Default	Description
priority	Ticket Priority			This variable takes the integer that corresponds to the desired level of priority given to a client's support tickets. 0 = Low 1 = Normal 2 = High 3 = 911
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. 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. 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_'.

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			
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		

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.cpanel_add - Add a cPanel Account

This method is used to add a cPanel account.

Parameter	Name	Required	Default	Description
user_login	Client's Login Name or ID	Y		
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.
hostid	Host ID		from service plan	
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		

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			

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,
        "owner_type": null,
        "owner": "0",
        "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
rack_id	Rack ID			
row_id	Row ID			
cage_id	Cage ID			
zone_id	Zone ID			
fac_id	Facility ID			

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			
tags	Device Tags			

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,
    "height": "1",
    "cage_id": 0,
    "assignments": [

    ],
    "disabled": "0",
    "desserv": null,
    "clientid": null,
    "fac_id": 0,
    "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",
    "timeout": "30",
    "username": "root",
    "class_id": "1",
    "accesskey": "",
    "id": "1",
    "version": "11.28.60",
    "type": "whm",
    "host": "host.example.com",
    "reseller": "0",
    "active": "1"
  },
  "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": "0000000000000000000000000a000021",
    "portable": "0",
    "desserv": "",
    "addr_type_readable": "Global",
    "dev_desc": "ubersmith.com",
    "num_ips": "1",
    "addr_type": "1",
    "addr": "0000000000000000000000000a000021/128"
  },
  "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
device_id	Device ID	Y		String or Array, CIDR or range notation.
addr	Address(es) to Assign	Y		
assign_description	Description			
client_id	Client ID			

Parameter	Name	Required	Default	Description
service_id	Service ID			
addr_type	Address Type		1	1 = Global 2 = Local 4 = Gateway
portable	Portable		0	0 = Not Portable 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": "0000000000000000000000000a000024",
        "portable": "0",
        "desserv": "Bargain Dedicated Server",
        "addr_type_readable": "Global",
        "dev_desc": "ubersmith.com",
        "num_ips": "1",
        "addr_type": "1",
        "addr": "0000000000000000000000000a000024/128"
      }
    }
  },
  "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			

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": "0000000000000000000000000a000024",
      "portable": "0",
      "desserv": "Bargain Dedicated Server",
      "addr_type_readable": "Global",
      "dev_desc": "ubersmith.com",
      "num_ips": "1",
      "addr_type": "1",
      "addr": "0000000000000000000000000a000024/128"
    }
  },
  "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 2 = Local 4 = Gateway
portable	Portable		0	0 = Not Portable 1 = Portable

Example Output

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

device.ip_get_available - List Available Device IPs

This method is used to list a device's available IPs.

Parameter	Name	Required	Default	Description
device_id	Device ID			
client_id	Client ID			
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"
    },
    "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"
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

device.ip_get_unassigned - Get Unassigned Device IPs

This method is used to get a device's unassigned IPs.

Parameter	Name	Required	Default	Description
portable	Portable		0	0 = Not Portable 1 = Portable
fac_id	Facility ID			
pool_id	IP Address Pool ID			
block_id	IP Address Block ID			
cidr	CIDR Assignment Size		32	
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	0 = Global
vlan_range_id	VLAN Range ID		0	Required if vlan_num is specified.
vlan_num	VLAN Number		0	Will be auto-assigned if not specified and vlan_range_id is non-zero.

Example Output

```
{
  "data": {
    "vlan_num": 0,
    "group_id": "16",
    "vlan_range_id": 0,
    "group_description": "this is the group description",
    "client_id": 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			
order_by	Order By			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
empty	Empty VLANs			If true and is_vlan, return all "empty" VLANs

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"
    }
  },
  "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	0 = Global
vlan_range_id	VLAN Range ID		0	Required if vlan_num is specified.
vlan_num	VLAN Number		0	Will be auto-assigned if not specified and vlan_range_id is non-zero.

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": "0000000000000000000000000b0b0b04",
    "portable": "0",
    "num_ips": "1",
    "addr_type": "1",
    "addr": "0000000000000000000000000b0b0b04/128"
  },
  "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			
status	Device Status			Possible values: up warn down
label	Device Label			
dev_desc	Device Description			

Parameter	Name	Required	Default	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.
require_ip	Include IP Addresses		0	Set to 1 to include IP Address information in the output.
metadata	Include Metadata		0	Set to 1 to include metadata in the output.
order_by	Order By			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": {
    "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",
      "height": "2",
      "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		

Example Output

```
This method will return data for a PNG format image.
```

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
				dns = DNS
				tcp = TCP
				ssl = SSL
				module = Device Module (DE-only, see extra)

address	Monitored IP Address			
notify	Notification Email Address			

Parameter	Name	Required	Default	Description
notify_delay	Notification Delay		0	Delay before sending notification in seconds 0 = Send notification immediately
notify_interval	Notification Interval		0	Interval between notifications in seconds 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 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
				dns = DNS
				tcp = TCP
				ssl = SSL
				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_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	Y		icmp = Ping
				dns = DNS
				tcp = TCP
				ssl = SSL
				module = Device Module (DE-only, see extra)

order_by	Order By			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": {
    "1": {
      "name": "HTTP test",
      "notify": "support@ubersmith.com",
      "addr_id": "0",
      "address": "ubersmith.com",
      "port": "0",
      "mon_id": "1",
      "reenable_time": null,
      "notify_delay": "300",
      "protocol": "icmp",
      "notify_interval": "0",
      "script_id": "2",
      "loss": "75",
      "disabled_by": "0",
      "extra": {
        "load_max": "",
        "io_min_bps": "",
        "notify_downstream": "0",
        "io_min_ops": "",
        "ram_max": "",
        "module": "dm_servermetrics",
        "iowait_max_pct": "",
        "cpu_max_pct": "",
        "virt_max": "",
        "ram_min": "",
        "iowait_min_pct": "",
        "cpu_min_pct": "",
        "notify_address": "support@ubersmith.com",
        "io_max_bps": "100000",
        "hdd_max_pct": "",
        "virt_min": "",
        "instructions": "testing",
        "io_max_ops": ""
      },
      "state": "-1",
      "dev": "1"
    }
  },
  "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
				dns = DNS
				tcp = TCP
				ssl = SSL
				module = Device Module (DE-only, see extra)

address	Monitored IP Address			
notify	Notification Email Address			
notify_delay	Notification Delay		0	Delay before sending notification in seconds 0 = Send notification immediately
notify_interval	Notification Interval		0	Interval between notifications in seconds 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 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": true,
  "error_message": "",
  "error_code": null,
  "status": true
}
```

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: on off 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: hour – Current Hour day – Current Day week – Current Week month – Current Month lastmonth – Previous Month year – Current Year lastyear – Previous Year custom – Custom Period If using the start_time and end_time parameters, period must be set to custom.

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_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			

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",
          "type":"Server",
          "type_en_US":"Server"
        }
      }
    }
  },
  "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			

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_'.

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) 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			

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_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

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.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			
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",
    "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\n\t\t\t\t\n\t\t\t\t\tOrder Summary                      Price/Period
      Setup
      0.00                    \n\t\t\t\t\n\t\t\t\t\n\t\t\t\t\tTotal Cost\n\t\t\t\t    0.00              $   \n\t\t\t\t\t
\n\t\t\t\t\t                  \n\t\t\t\t\t                 ",\n\t\t\t\n\t\t\t\t\t      \n\t\t\t\t\t
\n\t\t\t\t",
  "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

Parameter	Name	Required	Default	Description
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",
    "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.

Parameter	Name	Required	Default	Description
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.
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	Direction (asc,desc)
offset	Offset		0	If specified, return records starting from this offset (0-based).
limit	Limit			Limit

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",
      "listed_company": "",
      "order_form_id": "0",
      "order_queue_id": "8",
      "opportunity_id": "0",
      "ts": "1312304618",
      "owner": ""
    }
  },
  "error_message": "",
  "error_code": null,
  "status": true
}
```

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.

Example Output

```
{
  "data": true,
  "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": "Cancelled",
          "count": 0
        }
      },
      "name": "Main Order Queue"
    },
  },
  "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			

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

\n\t\t\t\tThank you for your order, for future reference your order number is: 1260
\n\t\t\t\tYour order is now being processed.\n\t\t\t",
  "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			
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

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
				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": {
    "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.department_get - Get Ticket Departments

This method is used to return the details for a specific department.

Parameter	Name	Required	Default	Description
dept_id	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.ticket_count - Count Support Tickets

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

Parameter	Name	Required	Default	Description
user_login	Client's Login Name or ID			
queue	Ticket Department ID			
type	Ticket Type			Possible types: Open On Hold Closed Deleted ClientAll (Open, On Hold & Closed)
priority	Ticket Priority			Possible priorities: 0 = Low 1 = Normal 2 = High 3 = 911
assigned	Assigned admin ID			

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 ",
    "timestamp": "1272986228",
    "assignment": "0",
    "order_id": "1129",
    "metadata": {
      "billable": ""
    },
    "weight": "0",
    "client_activity_type": "11",
    "activity": "1294173475",
    "contact_id": "0",
    "priority": "3",
    "subject": "test emergency",
    "recipient": "",
    "client_activity": "1294173475",
    "service_id": "1271",
    "ticket_id": "5141",
    "priority_name": "911",
    "qname": "General Support",
    "device_id": "14",
    "client_id": "1005",
    "type": "1",
    "type_name": "Open",
    "reply_to": "",
    "client_activity_type_name": "Client Followup",
    "admin_name": null,
    "source": "1",
    "opportunity_id": "0",
    "replies": "2",
    "activity_type": "11"
  },
  "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 ID			
queue	Ticket Department ID			
type	Ticket Type			Possible types: Open On Hold Closed Deleted ClientAll (Open, On Hold & Closed)
priority	Ticket Priority			Possible priorities: 0 = Low 1 = Normal 2 = High 3 = 911
assigned	Assigned admin ID			
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, weight, activity_type
direction	Direction		asc	
limit	Limit			
offset	Offset		0	If specified, return records starting from this offset (0-based).

Example Output

```
{
  "data": {
    "5048": {
      "cc": "",
      "client_replies": "1",
      "q_id": "1",
      "author": "Johnny Ubersmith ",
      "timestamp": "1266257592",
      "assignment": "0",
      "order_id": "0",
      "weight": "0",
      "client_activity_type": "0",
      "activity": "1266257592",
      "contact_id": "0",
      "priority": "911",
      "subject": "Server Question",
      "recipient": "support@ubersmith.com",
      "client_activity": "1266257592",
      "service_id": "238",
      "ticket_id": "5048",
      "device_id": "0",
      "client_id": "1005",
      "type": "Open",
      "listed_company": "ubersmith, inc.",
      "reply_to": "",
      "source": "client",
      "opportunity_id": "0",
      "replies": "1",
      "activity_type": "None"
    }
  },
  "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	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			Only used if offset is specified

Example Output

```
{
  "data": {
    "432": {
      "cc": "",
      "ticket_post_id": "432",
      "body": "body",
      "author": "Johnny Ubersmith ",
      "assignment": "0",
      "timestamp": "1312293776",
      "contact_id": "261",
      "subject": "Subject",
      "recipient": "luke@ubersmith.com",
      "ticket_id": "10092",
      "type": "21",
      "reply_to": "",
      "attachments": "1",
      "source": "2",
      "headers": "",
      "hidden": "0"
    }
  },
  "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.
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

Parameter	Name	Required	Default	Description
body	Ticket Body			
subject	Ticket Subject			
email	Author Email Address			
name	Author Name			
priority	Ticket Priority		1	
queue	Ticket Department ID		system default	
client_id	Client ID		0	
contact_id	Contact ID		0	
brand_id	Brand 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.

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			Not required for comments
userid	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.

Parameter	Name	Required	Default	Description
bcc	Bcc Address(es)			One or more (comma-separated) email addresses to receive a blind carbon copy of the response.
priority	Ticket Priority		1	
queue	Ticket Department ID		system default	
client_id	Client ID		0	
contact_id	Contact ID		0	
brand_id	Brand 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.

Example Output

```
{
  "data": "10094",
  "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			
priority	Ticket Priority			0 = Low, 1= Normal, 2 = High, 3 = 911
dept_id	Ticket Department ID			
status	Ticket Status			
type	Ticket Type			1 or open = Open, 2 or onhold = On Hold, 3 or closed = Closed, 4 or deleted = Deleted
subject	Ticket Subject			
cc	Carbon Copy			An email address to CC

Parameter	Name	Required	Default	Description
device_id	Associate Device			The input should be a device ID
client_id	Associate Client			The input should be a client ID
service_id	Associate Service			The input should be a service 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_'.

Example Output

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