API Module - Resource

Resources

ProVision's APIv1 system has been replaced by APIv2, and is now considered deprecated.

- Resources
  - get
  - add
  - update
  - delete
  - get resource link
  - get resource search

get

URL	/api/v1/api.php?target=resource&action=get
Description	Get a resource or resources
Returns	**Examples:** SUCCESSFUL: *{"success":1,"message":"Search successful","data":[{"id":"57","name":"2nd Email","slug":"6c-contact-email2","type":"field","parent_id":"1","category_id":null,"attr":[]}]}* ERROR: *{"success":0,"message":"Search failed"}*
Optional Parameters	**General Parameters:** **Limit Results by ID:** At most, one of the following:

General Parameters:

Name	Type	Notes/Example
name	STRING	Name of the resource. Example: 6Connect, Inc.
slug	STRING	The unique URL friendly name of the resource. Example: 6connect-inc
type	STRING	Type of resource (eg. *entry*, *field*, *category*, *dnsmodule*)
search	STRING	Search the resource system for the provided term. Performs a "LIKE" search to return similar results. Similar to GET Resource SEARCH call.
search_column	STRING	Column to perform a 'search' on.
permissions__true	STRING	Set the permissions that must be true. Typically only used for UI / Gadget permissions.
permissions__false	STRING	Set the permissions that must be false. Typically only used for UI / Gadget permissions.
getFromBackup	INTEGER	Use data from the resource_archive table instead of the standard resource table, used with "orig_id" identifying parameter.

Name	Type	Notes/Example
id	INTEGER	Get the resource which has this ID
orig_id	INTEGER	The resource id from the standard resource table, used in conjunction with 'getFromBackup'.
custom_id	INTEGER	The resource custom id provided by the user for the resource.
resource__in	ARRAY	Get any resource which has any of these IDs Syntax: &resource__in[]=1771&resource__in[]=14238 (Each resource id you wish to search over gets its own phrase.)
resource__not_in	ARRAY	Get all the resources which do not have any of these IDs Syntax: &resource__not_in[]=1771&resource__not_in[]=14238 (Each resource id you wish to exclude gets its own phrase.)

At most, one of the following:

Name	Type	Notes/Example
parent_id	INTEGER	Get the resources whose parent has this ID
parent__in	ARRAY	Get any resource whose parents have any of these IDs. Syntax: &parent__in[]=162&parent__in[]=299 (Each parent id you wish to search over gets its own phrase.)
parent__not_in	ARRAY	Get all resources whose parents do not have any of these IDs Syntax: &parent__not_in[]=1771&parent__not_in[]=14238 (Each parent id you wish to exclude gets its own phrase.)

At most, one of the following:

Name	Type	Notes/Example
category_id	INTEGER	Get the resources of the category that has this ID
category__in	ARRAY	Get any resources whose categories have any of these IDs. Syntax: &category__in[]=11002&category__in[]=11003 (Each category id you wish to search over gets its own phrase.)
category__not_in	ARRAY	Get the resources of all the categories that do not have any of these IDs Syntax: &parent__not_in[]=11002&parent__not_in[]=11003 (Each category id you wish to exclude gets its own phrase.)

Limit Results by Resource Link:

For resources for which exist a Resource Link, you may limit by resource link data:

<table>
<tr><th>Name</th><th>Type</th><th>Notes/Example</th></tr>
<tr><td>resource_link_type</td><td>STRING</td><td>The resource linkage relation name. Valid values include:

For type = dnsmodule:
<table><tr><th>Valid Value</th><th>Notes</th></tr><tr><td>dnsViewACL</td><td>Links a ACL and a Group. The View (Group) must be resource1, resource id and the ACL resource must be resource2 in the linkage table</td></tr><tr><td>dnsViewServer</td><td>Links a View (Group) with Server (DNS Connector), and the Group resource must be as resource1 and DNS Module as resource2. Used for attaching DNS servers to DNS Groups.</td></tr><tr><td>dnsZoneMaster</td><td>Links a DNS Zone resource with the Server that will be exported as Master. The DNS Zone must be in resource1 and the server as resource2. Used for Directly attaching zones to servers.</td></tr><tr><td>dnsZoneSlave</td><td>Links a DNS Zone resource with the Server that will be exported as Slave. The DNS Zone must be in resource1 and the server as resource2. Used for Directly attaching zones to servers.</td></tr><tr><td>dnsZoneServer</td><td>Links a DNS Zone resource with a Server resource. The DNS Zone must be in resource1 and the server as resource2. Used to directly attach servers to zones.</td></tr><tr><td>dnsZoneView</td><td>Links a DNS Zone resource with a Group. The DNS Zone must be resource1, the Group resource as resource 2.</td></tr></table></td></tr>
<tr><td>resource_link_column</td><td>INTEGER</td><td>The column to be used for the parameter in "resource_link_value". Valid integer values are:

'1' = to use resource1 in the first column

'2' = to use resource2 in the second column</td></tr>
<tr><td>resource_link_value</td><td>INTEGER</td><td>The resource id for the resource_linkage table to search (Example: "10697")</td></tr>
</table>

Limit Results by Attributes:

You can further limit the results based on attributes the resources may have:

Name	Type	Notes/Example
attributes	ARRAY	You can search on multiple attributes by including an array of attribute options: `var data = {` `"type: "entry",` `"attributes": [` `{` `"attr_key": "_section",` `"attr_value": "105",` `},` `{` `"attr_key": "address-mail-state",` `"attr_value": "CA",` `}` `],` `"resources_per_page: 10` `}`
attr_key	STRING	The name of the attribute. Example: network-fqdn
attr_value	STRING	The value of any attribute, or if attr_key is specified, the value of the attribute defined in attr_key.
attr_compare	STRING	If both attr_key and attr_value are given, the results are by default compared based on the value given as attr_value being equal to the value stored in the database. You can optionally change this by setting the STRING value of attr_compare to one of the following: • = *(default)* • != • • = • • = • LIKE • NOT LIKE • IN • NOT IN • BETWEEN • NOT BETWEEN When attr_compare is set to IN, NOT IN, BETWEEN, NOT BETWEEN, then attr_value must either be an array or a comma separated string.
attr_load	BOOL	Load resource attributes along with the resource entry

Result Ordering:

Name	Type	Notes/Example
order	STRING	Set the direction of the ordering of the results by ascending or descending order. Valid values are: • ASC *(default)* • DESC
orderby	STRING	The parameter to order results by. Valid values include: • none • id • name *(default)* • slug • type • parent_id • date • resource__in *(preserve order given in the resource__in array)*

Range Selection / Paging:

You can restrict the range of the resources returned.

Name	Type	Notes/Example
resources_per_page	INTEGER	How many resources to return per page, e.g.: '10' .
offset	INTEGER	How many resources to offset from the initial resource, to use as the first resource provided in the return list (the initial resource is 0, not 1).
paged	INTEGER	The page to return (starts at 1, not 0). This parameter is provided for convenience and is used to calculate the offset where: offset=(paged-1)*resources_per_page

Example URL	/api/v1/api.php?target=resource&action=get&id=7

add

URL	/api/v1/api.php?target=resource&action=add
Description	Add a resource.
Returns	**Examples:** */api/v1/api.php?target=resource&action=add&meta[name]=apitest&meta[type]=entry&meta[section]=firewall&fields[network-fqdn][]=www.example.com* SUCCESSFUL: {"success":1,"message":"Resource added","data":{"id":1077,"name":"apitest","slug":"apitest","type":"entry","parent_id":1,"category_id":"NULL","attr":{"_section":"70","network-fqdn":"www.example.com"},"section":{"id":"70","name":"Firewall","slug":"firewall","type":"section","parent_id":"1","category_id":null,"attr":{}}}} */api/v1/api.php?target=resource&action=add&meta[name]=apitest&meta[type]=entry&fields[network-fqdn][]=www.example.com* ERROR:{"success":0,"message":"Entries must be assigned to a section"}
Required Parameters	

Name	Type	Notes/Example
meta[name]	STRING	Name of the resource
meta[type]	STRING	Type of resource (entry, section, field, etc.)

Optional Parameters

Name	Type	Notes/Example
meta[parent_id]	INTEGER	ID of the parent resource
meta[category_id]	INTEGER	ID of the category

Required Parameters

(meta [type] = entry)

One of the following:

Name	Type	Notes/Example
meta[section_id]	INTEGER	ID of the section that the entry will be assigned to
meta[section]	STRING	Slug of the section that the entry will be assigned to

Optional Parameters

(meta [type] = entry)

Name	Type	Notes/Example
fields[]	ARRAY	Entry field values (for fields that have already been assigned to the section) can be populated when the entry is created. The format is field[field-slug][field-instance]. If the field instance is left blank, it will simply be the next value in the instance array. For example: *fields[network-fqdn][]=example.com&fields[network-fqdn][]=test.com* would be written in JSON as `var fields = {` `"network-fqdn": [` `"example.com",` `"test.com"` `]` `}` **Please note that the Field Slug might differ from the Field Name!** To find the correct slug to use in adding resources with field values go the Section of the Resource you are adding, click 'Edit', then click the name of the Fields you will be populating. An Edit Field box will pop up which displays the Field's slug. A field can be added to a section multiple times. The field instance is used to keep track of which field occurrence we are referring. In this example, the network-fqdn field had been added twice to the section so we were able to store two values for it.
meta [custom_id]	STRING	A custom ID for the entry. In the past this has been called the Resource Holder ID or Customer ID. Most recently it was implemented as a text field with the slug "6c-resourceholder-id." Now it is a fundamental part the entry type resources.

Required Parameters

(meta [type] = field)

Name	Type	Notes/Example
meta[field_type]	STRING	Type of field • text • textarea • radios • checkboxes • choicebox

Optional Parameters

(meta [type] = field)

Name	Type	Notes/Example
meta [help_block]	STRING	Fields can have a line of text under them with instructions
meta [options]	ARRAY	Fields of type radios, checkboxes, or choicebox can have multiple options. This could be multiple radio buttons or a choicebox (dropdown) with several options. For example: meta[type]=field&meta[name]=Colors&meta[field_type]=choicebox&meta[options][]=Blue&meta [options][]=Green Will create a choicebox with dropdown options of Blue and Green.

Required Parameters

(meta [type] = gadgets)

Name	Type	Notes/Example
gadgets[x][uuid]	INTEGER	x: The nth gadget being described in the call ('0' for the first gadget, '1' for the second, and so on). uuid: User-generated ID of the gadget to be created.
gadgets[x] [code]	STRING	x: The nth gadget being described in the call ('0' for the first gadget, '1' for the second, and so on). code: Slug of the gadget code to be created. List of valid Gadget codes: • Contact Info: "_contact_info" • Contacts: "_contacts" • DHCP Server: "_dhcp_server" • DNS: "_dns" • Document Storage: "_document_storage" • IPAM: "_ipam" • Peer Groups: "_peering_peer_groups" • Peering Sessions: "_peering_sessions" • VRFs: "_peering_vrfs" • Resource Linkage: "_resource_linkage" • Resource View: "_resource_view" • Reverse API Console: "_reverse_api" • Tech Info: "_tech_info"

Example - Adding the IPAM Gadget to a Section:

api.php?target=resource&action=add&meta[type]=section&meta[name]=TestSection_1&meta[parent_id]=1&gadgets[0][uuid] =uuid-586dbd260d6ef&gadgets[0][code]=_ipam

update

URL	/api/v1/api.php?target=resource&action=update
Description	Update a resource.
Returns	**Examples:** SUCCESSFUL: *{"success":1,"message":"Resource Updated","data":{"id":"1055","name":"87-child-1","slug":"87-child-1","type":" entry","parent_id":"87","category_id":"65","attr":{"_section":"70"},"section":{"id":"70","name":"Firewall","slug":"firewall","type":" section","parent_id":"1","category_id":null,"attr":{}}}}* ERROR: *{"success":0,"message":"No resource found with ID: 1079"}*

Required Parameters

Name	Type	Notes/Example
meta[id]	INTEGER	ID of resource
meta[type]	STRING	Type of resource (entry, section, field, ect)

Optional Parameters

(meta [type] = entry)

Name	Type	Notes/Example
fields[]	ARRAY	See "add" documentation

Optional Parameters

(meta [type] = section)

Name	Type	Notes/Example
fields []	ARRAY	The fields value should be all the fields that are assigned to the section. Giving an empty array as the fields value will remove all fields from the section. The format is: fields[position][key] The position value is the position that the field will appear in (0 is first). The position value must always be included. An example field format for an existing field could be: fields[0][id]=2 fields[0][slug]=asset-serial-number fields[0][help_block]=something fields[0][new]=false ■ Either the id or the slug is required, not both. ■ When the "new" parameter is not included, FALSE is assumed If you want to create a new field and assign it to the section, use a format like this: fields[10][name]=TextArea fields[10][field_type]=textarea fields[10][new]=true

delete

URL	/api/v1/api.php?target=resource&action=delete
Description	Delete a resource.
Returns	**Examples:** SUCCESSFUL: *{"success":1,"message":"Resource deleted."}* ERROR: *{"success":0,"message":"No resource found with ID: 57"}*
Required Parameters	

Name	Type	Notes/Example
id	INTEGER	ID of the resource

<table>
<tr><td>Optional Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Notes/Example</th></tr>
<tr><td>recursive</td><td>BOOL</td><td>When 1, deletes parent and child entries for the resource</td></tr>
</table>
A recursive delete will delete all resources, which are permitted to be deleted, from the bottom up.
Imagine the following hierarchy:
<pre>
            A
     B1   B2
 C11   C12   C21   C22
</pre>
If a recursive delete is performed on A, but C21 is not deletable, the following
resources would still be deleted: (B1, C11, C12, C22).
B2 would not be deleted because it depends on C21 and A would not be deleted because it depends on B2.
</td></tr>
<tr><td>Example URL</td><td>/api/v1/api.php?target=resource&action=delete&id=57</td></tr>
</table>

get resource link

Description	Get available resource links. If no resource links exist for the given resource, an empty object is returned.
URL	/api/v1/api.php?target=resource&action=getLink

Returns

Examples:

SUCCESSFUL:	{"success":1 ,"message":"Search successful", "data":{"meta":{"totalRecords":"3", "retrieved":3}, "0":{"id":"22", "resource_id1":"1292", "resource_id2":"1302", "relation":"dhcpPoolLink"}, "1":{"id":"2", "resource_id1":"1292", "resource_id2":"1452", "relation":"dhcpPoolLink"}, "2":{"id":"12", "resource_id1":"1422", "resource_id2":" 1482", "relation":"dhcpPoolLink"}}}
ERROR:	{"success":0, "message":"error message"}

Return Detail:

Name	Type	Description
id	INTEGER	Id of the resource linkage
resource_id1	INTEGER	The id of the parent resource
resource_id2	INTEGER	The id of the linked resource
relation	STRING	The relation type. Relation types include: contact, dhcpPoolLink, dnsViewACL, dnsViewServer, dnsZoneMaster, dnsZoneServer, dnsZoneView

Meta Attributes:

Name	Type	Description
totalRecords	INTEGER	How many records were found by this query, without pagination.
retrieved	INTEGER	How many records were returned by this query, with pagination.

Optional Attributes:

Name	Type	Description
resultsPerPage	INTEGER	How many records to include per page display.*
page	INTEGER	Which page to display, when used with "resultsPerPage"*
*Example pagination: api.php?target=resource&action=getLink&relation=dhcpPoolLink&resultsPerPage=100&page=2		

get resource search

Description	Search the resource system for the provided term. Performs a "LIKE" search to return similar results.
URL	/api/v1/api.php?target=resource&action=get&search=

Returns

Examples:

SUCCESSFUL:	{"success":1,"message":"Search successful","data":[{"id":"11011","name":"a6connectchildentry","slug":"a6connectchildentry","type":"entry","parent_id":"4210","category_id":null,"date":1499106555,"modified":1499106555,"attr":{"_section":"4214"},"section":{"id":"4214","name":"aQA Section","slug":"aqa-section","type":"section","parent_id":"1","category_id":null,"date":1498775688,"modified":1499106630,"attr":[]},"gadgets":[]}],"result_count":1,"found_count":1}
ERROR:	{"success":0, "message":"error message"}

Return Detail:

Name	Type	Description
id	INTEGER	Id of the resource linkage
name	STRING	The resource name
slug	STRING	The resource slug
type	STRING	The resource type.
parent_id	INTEGER	ID of the parent resource
category_id	INTEGER	ID of the resource category type
date	INTEGER	Resouce creation date
modified	INTEGER	Resource last modified date
attr	JSON	A JSON list of resource attributes

Required Attributes:

Name	Type	Description
search	STRING	The search term

Example URL

/api/v1/api.php?target=resource&action=get&search=6connect