

REFERENCE GUIDE

Infoblox REST API

NIOS 8.3

Contents

General Syntax and Options	5
Object Reference	5
Documentation	6
Common scenarios.....	7
Points to remember.....	7
Access.....	8
Authentication	8
Schema	10
Extensible Attributes	11
Extensible Attributes Inheritance	12
DNS-related scenarios	13
Zones.....	13
The most common record types	17
Retrieve all the records in a zone	27
Retrieve non-system generated records	29
Search for a record based on some criteria	29
Search for network objects based on regular expressions	31
Aliases for a host	32
Change the IP address of a host	34
Add or remove IP addresses from a host.....	34
Add a host with the next available IP address from a network.....	35
Add a host with a fixed address.....	38
Add Extensible Attributes to an object.....	39
TTL	39
Zone transfers.....	41
Response Policy Zones.....	42
DNS Traffic Control.....	50
IP address management related scenarios	53
Add a network or container.....	53
Search for a network	55
Get next available address from a subnet.....	56
Get all the addresses based on a search criteria	56
Zone associations on a network	58
Add a fixed address	59

Search for a fixed address by MAC	60
Get information about an IP address	60
Search for any object with an IP address	61
Search for an IP address	61
Delete an IP address	62
Network Views	62
DHCP related scenarios	65
DHCP ranges	65
Search for a DHCP range by Extensible Attributes	66
DHCP Leases	67
Grid management related scenarios	68
Grid DNS properties	68
Members	69
Services	72
vDiscovery	73
Permissions	74
Export a database backup	76
Export a CSV file	77
Export results of a WAPI call	78
Import a file (Certificate)	78
Import a CSV file	79
Sample paging request	80
Multiple Object Body Feature using /request	83
Get records of multiple record types	83
Delete a host record	85
Get next available IP address	85
Pre-provision a Grid member	86
Enable DHCP service on a member	87
Get permissions for an object	87
Sample Codes	88
Python	88
Powershell	90
Java	91
Using Java Unirest	91
Using Java OkHttpClient	92

C# (CSharp)	93
References	93

Introduction

The Infoblox WAPI is an interface based on REST (REpresentational State Transfer), also called a RESTful web API. These are not dependent on any programming language.

It uses HTTP methods for operations and supports input and output in JSON and XML. It supports the primary or most-commonly-used HTTP verbs: POST, GET, PUT, and DELETE. These correspond to create, read, update, and delete (or CRUD) operations, respectively.

All API calls are encrypted using SSL/TLS and authenticated using HTTP basic authentication.

General Syntax and Options

All WAPI requests consist of three parts; URL, Arguments and Data (body).

URL

The first part of the URL consists of the host IP address (or FQDN) to which all the calls will be directed. The second part of the URL identifies the requests as a WAPI request and specifies the expected version of WAPI.

The URL syntax is *<grid_IP>wapi/v<major.minor>*, e.g. *x.x.x.x/wapi/v2.4/*. The current version of the API is 2.7. If you need to use a patch release, you can specify it as *vx.x.x* (like *v2.6.1*)

The Infoblox WAPI has a versioning scheme that is independent of the NIOS versioning scheme. A current WAPI version is backward compatible with WAPI releases that have the same major WAPI version or with designated earlier major versions.

The last part of the URL identifies the resource, such as a network, on which the request operates.

Arguments

Query arguments (after '?') can be used to specify general options and method specific options and data for the request. All options start with the character *_* (underscore).

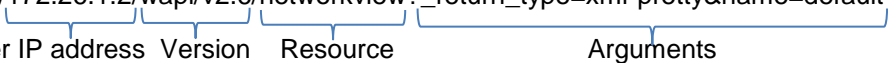
The general options are *_schema*, *_return_type*, *_return_fields* and *_method*.

Data (Body)

Contains data that is dependent on the method. Only, PUT, and POST methods can have a Body on input. All methods have Body on output.

Example

GET *https://172.26.1.2/wapi/v2.6/networkview?_return_type=xml-pretty&name=default*



Object Reference

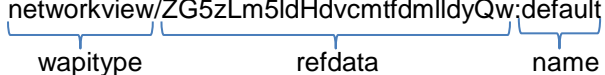
Every WAPI Object is referenced by a unique Object Reference. WAPI returns this reference when an object is created, modified, deleted or read. This reference is used to identify the object for the same operations.

An object reference is a string with the following format, without spaces:

wapitype / refdata [: name1 [{ / nameN }...]]

Example

networkview/ZG5zLm5ldHdvcmtfdmllldyQw:default



Documentation

For more information, you can navigate to the WAPI documentation available at https://<grid_IP>/wapidoc

Infoblox WAPI 2.6 documentation

Table Of Contents

- Infoblox WAPI documentation
 - Introduction
 - Notation
 - Transport and Authentication
 - Backward Compatibility
 - General Syntax and Options
 - Naming and Values
 - Object Reference
 - Function Calls
 - Extensible Attributes
 - Use Flags
 - Data Formats
 - XML Format
 - Error Handling
- Methods
 - GET
 - POST
 - PUT
 - DELETE
- Objects
- Struts
- Additional Information and Examples

Next topic
ad_auth_service: Active Directory Authentication Service object.

Quick search

Infoblox WAPI documentation

Introduction

The Infoblox WAPI is an interface based on REST (REpresentational State Transfer), also called a RESTful web API. It uses HTTP methods for operations and supports input and output in JSON and XML.

Notation

The following conventions are used to describe syntax for WAPI methods and objects:

What	Description
objref	A reference to an object. This must be a reference returned from an earlier call. For more information, see Object Reference .
WAPI	Used as a generic start in an URL. In real calls, this needs to be replaced with <code>/wapi/v2.6</code> or similar syntax.
objtype	The name of an object type, such as <code>network</code> .
field	The name of a field, such as <code>comment</code> .
value	The value of an item, such as a field. The value must be quoted according to where it is used. For information, see Naming and Values .
[thing]	These brackets are used to signify an optional value.
a b	The symbol <code> </code> is used to indicate that either <code>a</code> or <code>b</code> can be used.
thing...	<code>...</code> is used at the end of an item to signify that it can be repeated multiple times. Items must be separated in accordance with where they are used, such as <code>*</code> in arguments.
{ }	These brackets are used to group information in descriptions.

Transport and Authentication

You can also find more information in the [Support Site](#) under NIOS in the Tech Docs section.

NIOS

► Release Notes

► Administrator Guide

► CLI Guide

► API Reference Guide

▼ WAPI Documentation

NIOS 8.2 RESTful API Documentaiton 2.7.0

NIOS 8.1 RESTful API Documentaiton 2.6.0 MGM

NIOS 8.1 RESTful API Documentaiton 2.6.0

NIOS 8.0.0 RESTful API Documentation

NIOS 8.0.0 Multi-Grid-Manager RESTful API Documentation

NIOS 7.3.4 WAPI (RESTful) Reference

NIOS 7.3.0 WAPI (RESTful) Reference

NIOS 7.2.0 WAPI (RESTful) Reference

NIOS 7.1.0 WAPI (RESTful) Reference

NIOS 7.0.0 WAPI (RESTful) Reference

NIOS 6.8.2 WAPI (RESTful) Reference

NIOS 6.8 WAPI (RESTful) Reference

NIOS 6.7 WAPI (RESTful) Reference

NIOS 6.12.18 WAPI (RESTful) Reference

NIOS 6.12.0 WAPI (RESTful) Reference

NIOS 6.10.0 WAPI (RESTful) Reference

Common scenarios

This section will cover the most common scenarios that you may encounter in your day-to-day activities while working with the Infoblox grid.

Points to remember

- This document will cover the most common use cases. It will not cover all the APIs available. For any additional information, please refer to the WAPI documentation.
- For each of the use cases, it will cover the API call and corresponding cURL commands.
- The Sample Codes section contains sample Python, Powershell, Java and C# code.
- In the following WAPI Samples, the guide uses a base url referenced as <wapi_url>. Replace it with https://<grid_master_ip>/wapi/<wapi_version> or https://<grid_master_fqdn>/wapi/wapi_version. Example: <https://x.x.x.x/wapi/v2.7> or https://this.is.a.test.com/wapi/wapi_version
- In the sample cURL commands, replace the text in red with values specific to your environment.
- By default, all the fields are not returned during a GET request. You can use the `_return_fields` argument to get the desired data. You would have to explicitly mention the additional fields you require with a `_return_fields+=<requiredfield>`
- You can set `_return_as_object` argument to 1. This will change the format of the JSON output to make it easy to parse by 3rd party libraries. It is recommended to set this as it standardizes the JSON format and is less error prone.
- Set the `_max_results` option to 1 when you need to stop a search on the first match. This results in a huge performance and speed improvement in large environments.
- When a search result yields a large number of results, you can use paging.
 - To start a paging request, the initial search request must have `_paging` and `_return_as_object` set to 1, and `_max_results` set to the desired page size.
 - The returned results object will contain the `next_page_id` field and the result field set to the first page of results. Note that the `next_page_id` field only contains URL-safe characters so it can be used as is and no quotation characters are required for subsequent requests.
 - To get the subsequent results, you can re-send GET requests to the original object and set `_page_id` to the ID string returned in the previous page of results.
 - For a sample request, refer to Sample paging request
- When making POST, or PUT requests, it is easier to send the data as 'application/json' in the body of the message, since it lets you send complex data structures in the payload.
- Field and argument values must be quoted according to where they are used.
Examples:
 - URL args, x-www-form-urlencoded:
Use %xx encoding for "%", ",", "/", "?", ":", "@", "&", "=", "+", "\$", " ", and " " (a space)
 - JSON Data:
Use JSON quoting, as specified at <http://json.org>
 - XML Data:
Use XML quoting (& etc.) as needed for XML.
- In the following examples, '+' is encoded as %2B, ':' as %3A, '<' as 3C, '>' as 3E
- The following examples include a sample output for each API call. This data is only representative of the test environment. "...." indicates that there are more records returned than mentioned.
- By default, the output of POST, PUT, and DELETE contains the `_ref` of the concerned object.
- PUT and DELETE operations require `_ref` of the object you are working with.
 - In the following WAPI Samples, the text highlighted in green indicates an object reference specific to the test environment. Please replace it with the object reference specific to your environment. Example: In `{wapi_url}/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLmhvc3Qx:host.info.com/default` replace with `{wapi_url}/<_ref>`
 - You can get the `_ref` by performing a GET operation.
- If a field is of type BOOLEAN, do not quote the word. Example: "is_default":true

- Some fields are associated with a corresponding boolean flag value that has the prefix *use_*. For example, *ttl* is associated with the flag *use_ttl*. In an object, the value of this field will only take effect when its use flag is *true*. Otherwise, the value will be inherited from a higher level setting.
- While searching for network objects, you can filter the data using regular expressions. You would need to specify the *~* modifier to indicate you are querying with a regular expression.
 - A search argument can use the following modifiers

Modifier	Explanation
!	Negates the condition
:	Makes string matching case insensitive
~	Regular expression search. Expressions are unanchored
<	Less than or equal
>	Greater than or equal

- Only one of the following can be specified at a time: greater than, less than, and regular expressions.
- Depending on the attribute type, following are modifiers supported by extensible attributes:
 - integer and date support *!*, *<* and *>*.
 - All other types behave like strings and support *!*, *~* and *:*.
- When you need to update or create multiple records, you can store the data as fields in a CSV file and import it at once. This will be faster in comparison with updating or creating each object with a separate API call. The *Import a CSV file* section covers an example.
- The *request* object allows the control of WAPI through a single-entry point. The Multiple Object Feature enables to make multiple requests with one API call, thus simplifying an operation, and reducing the number of API calls to be made. You can find a few samples in the Multiple Object Body Feature using */request* section.

Access

WAPI uses HTTPS (HTTP over SSL/TLS) as the transport mechanism. You can access WAPIs using various methods.

1. Applications that allow you to make REST calls, like Insomnia and POSTMAN
2. Any programming language or utility, like curl and python (by importing the requests module)
3. Any third-party application, that you want to integrate with, that supports REST methods, like ServiceNow and HP Operations Orchestrator.

Authentication

The server certificate used for WAPI is the same certificate used by NIOS for the GUI.

WAPI supports only HTTP Basic Authentication. You can use the connection for issuing multiple requests. In the case of multiple requests, authentication is handled by supplying the cookie (ibapauth) that was returned after the initial authentication. The IBAPAUTH cookie is sent back by the server to avoid repeat authentication requests. It is sent as an HTTPONLY, SECURE cookie.

Example: set-cookie: ibapauth="ip=127.0.0.1,client=API,group=admin-group,ctime=1446631868,timeout=600,mtime=1446631868,su=1,auth=LOCAL,user=admin,NlxMltsoxDNvBWKrfOMy+uxUMWS3guCB4yU"

This cookie can be invalidated by sending a POST request to */wapi/v2.7/logout*

WAPI supports the same underlying authentication methods that NIOS supports for username and password. This also applies to the cookie timeout which is the same value as the Grid UI timeout. All WAPI users must have permissions that grant them access to the API.

You can login with any API call. This will set the cookie.

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
A sample API call (Get host records) to save the cookie generated to a file called cookies.txt	curl -k -u admin:Infoblox -c cookies.txt -X GET "https://127.0.0.1/wapi/v2.7/record:host?_return_as_object=1"		
	<pre>{ "result": [{ "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS50ZXN0Lmhvc3Qx: host1.test.com/default", "ipv4addrs": [{ "_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWVzcyQuX2RlZmF1bHQuY29tLnRlc3QuaG9zdDEuMTcyLjIjEuMjAwLg: 172.26.1.200/host1.test.com/default", "configure_for_dhcp": false, "host": "host1.test.com", "ipv4addr": "172.26.1.200" }], "name": "host1.test.com", "view": "default" }, { "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLnByaW50ZXI: printer.info.com/default", "ipv4addrs": [{ "_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWVzcyQuX2RlZmF1bHQuY29tLmluZm8ucHJpbnRlci4xNjAuMjAwMS4xMDAu: 160.20.1.100/printer.info.com/default", "configure_for_dhcp": false, "host": "printer.info.com", "ipv4addr": "160.20.1.100" }], "name": "printer.info.com", "view": "default" }] }</pre>		
A sample API call (Get zones) to show how to use the generated cookie from cookies.txt	curl -k -b cookies.txt -X GET "https://127.0.0.1/wapi/v2.7/zone_auth_return_as_object=1"		
	<pre>{ "result": [{ "_ref": "zone_auth/ZG5zLnpvbWUkLI9kZWZhdWx0LmNvbS50ZXN0: test.com/default", "fqdn": "test.com", "view": "default" }] }</pre>		
Modify the session timeout	PUT	<wapi_url>/grid/b25ILmNsdXN0ZXIkMA:Infoblox	<pre>{ "security_setting": { "session_timeout": 60000 } }</pre>
	<pre>curl -k -b cookies.txt -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/grid/b25ILmNsdXN0ZXIkMA:Infoblox?_return_as_object=1&_return_fields%2B=security_setting" -d '{"security_setting":{"session_timeout": 60000}}'</pre>		
	<pre>{ "result": { "_ref": "grid/b25ILmNsdXN0ZXIkMA:Infoblox", "security_setting": { "admin_access_items": [], "audit_log_rolling_enable": true, "http_redirect_enable": false, "lcd_input_enable": true, "login_banner_enable": true, </pre>		

	<pre> "login_banner_text": "Disconnect NOW if you have not been expressly authorized to use this system.", "remote_console_access_enable": false, "security_access_enable": false, "security_access_remote_console_enable": true, "session_timeout": 600000, "ssh_perm_enable": true, "support_access_enable": false, "support_access_info": "Not enabled" }}} </pre>	
Invalidate a cookie	POST	<wapi_url>/logout
	<pre>curl -k -b cookies.txt -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/logout"</pre>	

Schema

The `_schema` argument returns the WAPI schema.

If a WAPI schema is requested without specifying the type of network object, an object with information about requested version of the WAPI schema, list of supported objects in the requested version, and list of all supported versions will be returned.

If you specify a network object while requesting WAPI schema, along with the above mentioned information, you will get a list of cloud restrictions, list of fields of the object, list of object restrictions, requested object type, and actual requested version of the WAPI object schema.

Operation	REST Method	API Call
	Sample cURL Command	
	Sample Output	
Get the overall WAPI Schema (supported versions and supported objects)	GET	<wapi_url>/?_schema
	<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/?_schema&_return_as_object=1"</pre>	
	<pre> {"result": {"requested_version": "2.7", "supported_objects": ["ad_auth_service", "admingroup", "adminrole", "zone_delegated", "zone_forward", "zone_rp", "zone_stub"], "supported_versions": ["1.0","1.1","1.2","1.2.1", "1.3","1.4","1.4.1", "2.6","2.6.1","2.7"] } } </pre>	
Get schema of an object (<i>record:host</i> can be replaced with any other object)	GET	<wapi_url>/record:host?_schema
	<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/record:host?_schema&_return_as_object=1"</pre>	

	<pre> {"result": {"cloud_additional_restrictions": ["function call"], "fields": [{"is_array": true, "name": "aliases", "standard_field": false, "supports": "rwu", "type": ["string"]}], {"is_array": false, "name": "allow_telnet", "standard_field": false, "supports": "rwu", "type": ["bool"]}], {"is_array": false, "name": "zone", "searchable_by": "=", "standard_field": false, "supports": "rs", "type": ["string"]} } }, "restrictions": [], "type": "record:host", "version": "2.7" } } </pre>
--	---

Extensible Attributes

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Get the definition of the Extensible Attributes	GET	<wapi_url>/extensibleattributedef	
	<pre> curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/extensibleattributedef?_return_as_object=1" </pre>		
	<pre> {"result": [{"_ref": "extensibleattributedef/b25ILmV4dGVuc2libGVfYXR0cmliZXNkZWYkLkK1aWxkaW5n:Building", "default_value": null, "name": "Building", "type": "STRING" }, {"_ref": "extensibleattributedef/b25ILmV4dGVuc2libGVfYXR0cmliZXNkZWYkLkK1aWxkaW5n:GU:ReportingSite", "default_value": "site1", "name": "ReportingSite", </pre>		

	<pre>"type": "ENUM" }}}</pre>		
Create "Owner" Extensible Attribute of type String	POST	<wapi_url>/extensibleattributedef	<pre>{ "name": "Owner", "type": "STRING" }</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/extensibleattributedef? _return_fields%2B=name,type&_return_as_object=1" -d '{"name": "Owner","type": "STRING"}'</pre>		
	<pre>{ "result": { "_ref": "extensibleattributedef/b25ILmV4dGVuc2libGVfYXR0cmliZXIc19kZWYkLk93bmVy:Owner", "default_value": null, "name": "Owner", "type": "STRING" } }</pre>		
Create a "Location" Extensible Attribute of type list with possible values Santa Clara, Tacoma, Bangalore, Mumbai	POST	<wapi_url>/extensibleattributedef	<pre>{ "name": "Location", "type": "ENUM", "list_values": [{ "value": "Santa Clara" }, { "value": "Tacoma" }, { "value": "Bangalore" }, { "value": "Mumbai" }] }</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/extensibleattributedef? _return_fields%2B=name,type&_return_as_object=1" -d '{"name": "Location","type": "ENUM","list_values": [{"value": "Santa Clara"}, {"value": "Tacoma"}, {"value": "Bangalore"}, {"value": "Mumbai"}]}'</pre>		
	<pre>{ "result": { "_ref": "extensibleattributedef/b25ILmV4dGVuc2libGVfYXR0cmliZXIc19kZWYkLkxvY2F0aW9u:Location", "default_value": null, "name": "Location", "type": "ENUM" } }</pre>		

Extensible Attributes Inheritance

If inheritance is enabled for an extensible attribute, the following fields are supported in the extensible attribute value object:

Field	Description/Supported Values	
value	Value of the EA	
inheritance_source	This is a read-only field. It is a reference to the object this EA is inheriting from.	
inheritance_operation	INHERIT DELETE OVERRIDE	
descendants_action	option_delete_ea	REMOVE RETAIN
	option_with_ea	CONVERT INHERIT RETAIN
	option_without_ea	INHERIT NOT_INHERIT

Please note that for extensible attributes inheritance to work correctly with Host Record object (*record:host*) *use_for_ea_inheritance* should be set to *true* for one of its IP host addresses: IPv4 Host address object (*record:host_ipv4addr*) or IPv6 Host address object (*record:host_ipv6addr*).

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Add an extensible attribute with inheritance	PUT	<code><wapi_url>/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS50ZXN0Lmhvc3Qx: host1.test.com/default</code>	<code>{"extattrs": {"Location": {"descendants_action": {"option_with_ea": "RETAIN", "option_without_ea": "NOT_INHERIT", "value": "Santa Clara"}}}}</code>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS50ZXN0Lmhvc3Qx: host1.test.com/default?_return_fields%2B=extattrs&_return_as_object=1" -d '{"extattrs":{"Location": {"descendants_action": {"option_with_ea": "RETAIN", "option_without_ea": "NOT_INHERIT", "value": "Santa Clara"}}}'</pre>		
	<pre>{"result": {"_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS50ZXN0Lmhvc3Qx: host1.test.com/default", "extattrs": {"Location": {"value": "Santa Clara"}}, "ipv4addrs": [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWwRkcmVzcyQuX2RlZmF1bHQuY29tLnRlc3QuaG9zdDEuMTcyLjI2LjEuMjAwLg:172.26.1.200/host1.test.com/default", "configure_for_dhcp": false, "host": "host1.test.com", "ipv4addr": "172.26.1.200"}], "name": "host1.test.com", "view": "default"}}</pre>		

DNS-related scenarios

Zones

Zone type	Operation	REST Method	API Call	Sample Body (for POST and PUT)
		Sample cURL Command		
		Sample Output		
Authoritative Zone	Fetch information	GET	<code><wapi_url>/zone_auth</code>	
		<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/zone_auth?_return_as_object=1"</pre>		
		<pre>{"result": [{"_ref": "zone_auth/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS50ZXN0:test.com/default",</pre>		

		<pre>"fqdn": "test.com", "view": "default" }}</pre>	
Create	POST	<wapi_url>/zone_auth	<pre>{"fqdn": "info.com"}</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/zone_auth?_return_fields%2B=fqdn&_return_as_object=1" -d '{"fqdn": "info.com"}'</pre>		
	<pre>{"result": {"_ref": "zone_auth/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZv:info.com/default", "fqdn": "info.com", "view": "default" }} </pre>		
Create IPv4 reverse mapping zone	POST	<wapi_url>/zone_auth	<pre>{"fqdn": "10.10.10.in-addr.arpa ", "zone_format":"IPv4"}</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/zone_auth?_return_fields%2B=fqdn&_return_as_object=1" -d '{"fqdn": "10.10.10.in-addr.arpa","zone_format":"IPv4"}'</pre>		
	<pre>{"result": {"_ref": "zone_auth/ZG5zLnpvbmUkLI9kZWZhdWx0LmFycGEuaW4tYWRkc4xMC4xMC4xMA:10.10.10.0%2F24/default", "fqdn": "10.10.10.0/24", "view": "default" }} </pre>		
Modify	PUT	<wapi_url>/zone_auth/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZv:info.com/default	<pre>{"grid_primary":[{"name":"infoblox.localdomain"}]}</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/zone_auth/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZv:info.com/default?_return_fields%2B=fqdn,grid_primary&_return_as_object=1" -d '{"grid_primary":[{"name":"infoblox.localdomain"}]}'</pre>		
	<pre>{"result": {"_ref": "zone_auth/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZv:info.com/default", "fqdn": "info.com", "grid_primary": [{"name": "infoblox.localdomain", "stealth": false }], "view": "default" }} </pre>		
Delete	DELETE	<wapi_url>/zone_auth/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZv:info.com/default	
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X DELETE "https://127.0.0.1/wapi/v2.7/zone_auth/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZv:info.com/default&?_return_as_object=1"</pre>		
	<pre>{"result": "zone_auth/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZv:info.com/default"}</pre>		

Forward Zone	Fetch information	GET	<wapi_url>/zone_forward	
		curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/zone_forward?_return_as_object=1"		
		<pre>{ "result": { "_ref": "zone_forward/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5mb3J3YXJkdGVzdA:forwardtest.com/default", "forward_to": { "address": "172.26.1.10", "name": "host.forwardtest.com" }, "fqdn": "forwardtest.com", "view": "default" } }</pre>		
	Create	POST	<wapi_url>/zone_forward	<pre>{ "fqdn": "infoblox.com", "forward_to": { "address": "10.10.10.11", "name": "host.infoblox.com" } }</pre>
		<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/zone_forward?_return_as_object=1" -d '{"fqdn": "infoblox.com", "forward_to": { "address": "10.10.10.11", "name": "host.infoblox.com" }}'</pre>		
		<pre>{ "result": { "_ref": "zone_forward/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZvYmxveA:infoblox.com/default", "forward_to": { "address": "10.10.10.11", "name": "host.infoblox.com" }, "fqdn": "infoblox.com", "view": "default" } }</pre>		
	Modify	PUT	<wapi_url>/zone_forward/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZvYmxveA:infoblox.com/default	<pre>{ "forwarding_servers": { "name": "infoblox.localdomain" } }</pre>
		<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/zone_forward/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZvYmxveA:infoblox.com/default?_return_fields%2B=forwarding_servers&_return_as_object=1" -d '{"forwarding_servers": { "name": "infoblox.localdomain" }}'</pre>		
		<pre>{ "result": { "_ref": "zone_forward/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZvYmxveA:infoblox.com/default", "forward_to": { "address": "10.10.10.11", "name": "host.infoblox.com" }, "forwarding_servers": { "forward_to": [], "forwarders_only": false, "name": "infoblox.localdomain", "use_override_forwarders": false }, "fqdn": "infoblox.com", "view": "default" } }</pre>		

Zone Delegation	Delete	DELETE	<wapi_url>/zone_forward/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZvYmxveA:infoblox.com/default	
		curl -k -u admin:Infoblox -H 'content-type: application/json' -X DELETE "https://127.0.0.1/wapi/v2.7/zone_forward/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZvYmxveA:infoblox.com/default?_return_as_object=1"		
		{ "result": "zone_forward/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZvYmxveA:infoblox.com/default" }		
	Fetch information	GET	<wapi_url>/zone_delegated	
		curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/zone_delegated?_return_as_object=1"		
		{ "result": [{"_ref": "zone_delegated/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZvLnRlc3Q:test.info.com/default", "delegate_to": [{"address": "172.26.1.100", "name": "host.test.info.com"}], "fqdn": "test.info.com", "view": "default"}] }		
	Create	POST	<wapi_url>/zone_delegated	{ "fqdn": "demo.info.com", "delegate_to": [{"address": "10.10.10.20", "name": "host.demo.info.com"}] }
		curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/zone_delegated?_return_fields%2B=fqdn,delegate_to&_return_as_object=1" -d '{"fqdn":"demo.info.com","delegate_to":[{"address":"10.10.10.20","name":"host.demo.info.com"}]}'		
		{ "result": {"_ref": "zone_delegated/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZvLmRlbW8:demo.info.com/default", "delegate_to": [{"address": "10.10.10.20", "name": "host.demo.info.com"}], "fqdn": "demo.info.com", "view": "default"} }		
	Modify	PUT	<wapi_url>/zone_delegated/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZvLmRlbW8:demo.info.com/default	{ "locked": true }
		curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/zone_delegated/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZvLmRlbW8:demo.info.com/default?_return_fields=locked&_return_as_object=1" -d '{"locked":true}'		
		{ "result": {"_ref": "zone_delegated/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZvLmRlbW8:demo.info.com/default", "locked": true} }		
	Delete	DELETE	<wapi_url>/zone_delegated/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZvLmRlbW8:demo.info.com/default	

		<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X DELETE "https://127.0.0.1/wapi/v2.7/zone_delegated/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZvLmRlbW8:demo.info.com/default?_return_as_object=1"</pre>
		<pre>{"result": "zone_delegated/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZvLmRlbW8:demo.info.com/default"}</pre>

The most common record types

Record type	Operation	REST Method	API Call	Sample Body (for POST and PUT)
		Sample cURL Command		
		Sample Output		
HOST	Fetch information	GET	<wapi_url>/record:host	
		<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/record:host?_return_as_object=1"</pre>		
		<pre>{"result": [{"_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS50ZXN0Lmhvc3Qx:host1.test.com/default", "ipv4addrs": [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RlYWRkcmVzcyQuX2RlZmF1bHQuY29tLnRlc3QuaG9zdDEuMTcyLjI2LjEuMjAwLg:172.26.1.200/host1.test.com/default", "configure_for_dhcp": false, "host": "host1.test.com", "ipv4addr": "172.26.1.200" }], "name": "host1.test.com", "view": "default" }, {"_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLnByaW50ZXI:printer.info.com/default", "ipv4addrs": [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RlYWRkcmVzcyQuX2RlZmF1bHQuY29tLmluZm8ucHJpbnRlci4xNjAuMjAwMDAu:160.20.1.100/printer.info.com/default", "configure_for_dhcp": false, "host": "printer.info.com", "ipv4addr": "160.20.1.100" }], "name": "printer.info.com", "view": "default" }]}</pre>		
	Create	POST	<wapi_url>/record:host	<pre>{"name":"host.info.com", "ipv4addrs":[{"ipv4addr":"10.10.10.20"}]}</pre>
		<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/record:host?_return_fields%2B=name,ipv4addrs&_return_as_object=1" -d '{"name":"host.info.com","ipv4addrs":[{"ipv4addr":"10.10.10.20"}]}'</pre>		
		<pre>{"result": {"_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLmRlbW8:demo.info.com/default", "ipv4addrs":</pre>		

		<pre> [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWwRkcmVzcyQuX2RlZmF1bHQuY29tLmluZm8uaG9zdC4xMC4xMC4xMC4yMC4:10.10.10.20/host.info.com/default", "configure_for_dhcp": false, "host": "host.info.com", "ipv4addr": "10.10.10.20" }], "name": "host.info.com", "view": "default" }] </pre>		
	Modify	PUT	 <wapi_url>/record:host/ZG5zLmhvc3QkLI9kZWZhZWx0LmNvbS5pbmZvLmhvc3Q:host.info.com/default 	{"name": "host1.info.com"}
		<pre> curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/record:host/ZG5zLmhvc3QkLI9kZWZhZWx0LmNvbS5pbmZvLmhvc3Q:host.info.com/default?_return_fields%2B=name&_return_as_object=1" -d '{"name": "host1.info.com"}' </pre>		
		<pre> {"result": {"_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhZWx0LmNvbS5pbmZvLmhvc3Qx:host1.info.com/default", "ipv4addrs": [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWwRkcmVzcyQuX2RlZmF1bHQuY29tLmluZm8uaG9zdDEuMTAuMTAuMTAuMjAu:10.10.10.20/host1.info.com/default", "configure_for_dhcp": false, "host": "host1.info.com", "ipv4addr": "10.10.10.20" }] , "name": "host1.info.com", "view": "default" } } </pre>		
	Delete	DELETE	 <wapi_url>/record:host/ZG5zLmhvc3QkLI9kZWZhZWx0LmNvbS5pbmZvLmhvc3Qx:host1.info.com/default 	
		<pre> curl -k -u admin:Infoblox -H 'content-type: application/json' -X DELETE "https://127.0.0.1/wapi/v2.7/record:host/ZG5zLmhvc3QkLI9kZWZhZWx0LmNvbS5pbmZvLmhvc3Qx:host1.info.com/default?_return_as_object=1" </pre>		
		<pre> {"result": "record:host/ZG5zLmhvc3QkLI9kZWZhZWx0LmNvbS5pbmZvLmhvc3Qx:host1.info.com/default"} </pre>		
A	Fetch information	GET	 <wapi_url>/record:a 	
		<pre> curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/record:a?_return_as_object=1" </pre>		
		<pre> {"result": [{"_ref": "record:a/ZG5zLmJpbmRfYSQuX2RlZmF1bHQuY29tLmluZm8sdGVzdC5ob3N0LDE3Mi4yNi4xLjEwMA:host.test.info.com/default", "ipv4addr": "172.26.1.100", "name": "host.test.info.com", "view": "default" }] } </pre>		
	Create	POST	 <wapi_url>/record:a 	<pre> {"name": "server.info.com", "ipv4addr": "10.10.10.2"} </pre>

		<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/record:aaaa?_return_fields%2b=name,ipv6addr&_return_as_object=1" -d '{"name":"server.info.com","ipv6addr":"fd60:e32:f1b9::2"}'</pre>		
		<pre>{ "result": { "_ref": "record:aaaa/ZG5zLmJpbmRfYWZhYSQuX2RlZmF1bHQuY29tLmluZm8sc2VydMvyLGZkNjA6ZTMzOmYxY jk6Ojl:server.info.com/default", "ipv6addr": "fd60:e32:f1b9::2", "name": "server.info.com", "view": "default" }} </pre>		
	Modify	PUT	<wapi_url>/record:aaaa/ZG5zLmJpbmRfYWZhYSQuX2RlZmF1bHQuY29tLmluZm8sc2VydMvyLGZkNjA6ZTMzOmYxYjk6Ojl:server.info.com/default	{ "name": "server1.info.com" }
		<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/record:aaaa/ZG5zLmJpbmRfYWZhYSQuX2RlZmF1bHQuY29tLmluZm8sc2Vyd mVydMvyLGZkNjA6ZTMzOmYxYjk6Ojl:server.info.com/default?_return_fields%2B=name&_return_as_object=1" - d '{"name":"server1.info.com"}'</pre>		
	Delete	<pre>{ "result": { "_ref": "record:aaaa/ZG5zLmJpbmRfYWZhYSQuX2RlZmF1bHQuY29tLmluZm8sc2VydMvyMSxmZDYwOmUzMjp mMWI5Ojoy:server1.info.com/default", "ipv6addr": "fd60:e32:f1b9::2", "name": "server1.info.com", "view": "default" }} </pre>		
		DELETE	<wapi_url>/record:aaaa/record:aaaa/ZG5zLmJpbmRfYWZhYSQuX2RlZmF1bHQuY29tLmluZm8sc2VydMvyMSxmZDYwOmUzMjp mMWI5Ojoy:server1.info.com/default	
CNAME	Fetch information	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X DELETE "https://127.0.0.1/wapi/v2.7/record:aaaa/ZG5zLmJpbmRfYWZhYSQuX2RlZmF1bHQuY29tLmluZm8sc2Vyd mVydMvyMSxmZDYwOmUzMjp mMWI5Ojoy:server1.info.com/default?_return_as_object=1"</pre>		
		<pre>{ "result": { "record:aaaa/ZG5zLmJpbmRfYWZhYSQuX2RlZmF1bHQuY29tLmluZm8sc2VydMvyMSxmZDYwOmUzMjp mMWI5Ojoy:server1.info.com/default" }} </pre>		
		GET	<wapi_url>/record:cname	
	Create	POST	<wapi_url>/record:cname	{ "name": "cnametest.demo.info.com", "canonical": "demo.info.com" }

		<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/record:cname?_return_fields%2B=name,canonical&_return_as_object=1" -d '{"name":"cnametest.demo.info.com","canonical":"demo.info.com"}'</pre>	
		<pre>{"result": {"_ref": "record:cname/ZG5zLmJpbmRfY25hbWUkLI9kZWZhdWx0LmNvbS5pbmZvLmRibW8uY25hbWV0ZXN0:cna metest.demo.info.com/default", "canonical": "demo.info.com", "name": "cnametest.demo.info.com", "view": "default" }} </pre>	
Modify	PUT	<wapi_url>/record:cname/ZG5zLmJpbmRfY25hbWUkLI9kZWZhdWx0LmNvbS5pbmZvLmRibW8uY25hbWV0ZXN0:cnametest.demo.info.com/default	<pre>{"name":"cname.demo.info.com "}</pre>
		<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/record:cname/ZG5zLmJpbmRfY25hbWUkLI9kZWZhdWx0LmNvbS5pbmZvLmRibW8uY25hbWV0ZXN0:cnametest.demo.info.com/default?_return_fields%2B=name&_return_as_object=1" -d '{"name":"cname.demo.info.com"}'</pre>	
		<pre>{"result": {"_ref": "record:cname/ZG5zLmJpbmRfY25hbWUkLI9kZWZhdWx0LmNvbS5pbmZvLmRibW8uY25hbWU:cname.de mo.info.com/default", "canonical": "demo.info.com", "name": "cname.demo.info.com", "view": "default" }} </pre>	
Delete	DELETE	<wapi_url>/record:cname/ZG5zLmJpbmRfY25hbWUkLI9kZWZhdWx0LmNvbS5pbmZvLmRibW8uY25hbWU:cname.demo.info.com/default	
		<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X DELETE "https://127.0.0.1/wapi/v2.7/record:cname/ZG5zLmJpbmRfY25hbWUkLI9kZWZhdWx0LmNvbS5pbmZvLmRibW8uY25hbWU:cname.demo.info.com/default?_return_as_object=1"</pre>	
		<pre>{"result": "record:cname/ZG5zLmJpbmRfY25hbWUkLI9kZWZhdWx0LmNvbS5pbmZvLmRibW8uY25hbWU:cname.de mo.info.com/default"} </pre>	
MX	Fetch information	GET	<wapi_url>/record:mx
		<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/record:mx?_return_as_object=1"</pre>	
		<pre>{"result": [{"_ref": "record:mx/ZG5zLmJpbmRfbXgkLI9kZWZhdWx0LmNvbS50ZXN0Lm1haWwuaG9zdDEudGVzdC5jb20uMQ: mail.test.com/default", "mail_exchanger": "host1.test.com", "name": "mail.test.com", "preference": 1, "view": "default" }]} </pre>	

	Create	POST	<wapi_url>/record:mx	{ "mail_exchanger": "exchange.info.com", "name": "mx.info.com", "preference": 1 }
		curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/record:mx?_return_fields%2B=mail_exchanger,name&_return_as_object=1" -d '{"mail_exchanger": "exchange.info.com", "name": "mx.info.com", "preference": 1}'		
		{ "result": { "_ref": "record:mx/ZG5zLmJpbmRfbXgkLI9kZWZhdWx0LmNvbS5pbmZvLm14LmV4Y2hhbmdlLmluZm8uY29tLjE:mx.info.com/default", "mail_exchanger": "exchange.info.com", "name": "mx.info.com", "preference": 1, "view": "default" } }		
	Modify	PUT	<wapi_url>/record:mx/ZG5zLmJpbmRfbXgkLI9kZWZhdWx0LmNvbS5pbmZvLm14LmV4Y2hhbmdlLmluZm8uY29tLjE:mx.info.com/default	{ "name": "mail.info.com" }
		curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/record:mx/ZG5zLmJpbmRfbXgkLI9kZWZhdWx0LmNvbS5pbmZvLm14LmV4Y2hhbmdlLmluZm8uY29tLjE:mx.info.com/default?_return_fields%2B=name&_return_as_object=1" -d '{"name": "mail.info.com"}'		
		{ "result": { "_ref": "record:mx/ZG5zLmJpbmRfbXgkLI9kZWZhdWx0LmNvbS5pbmZvLm1haWwuZXhjaGFuZ2UuaW5mby5jb20uMQ:mail.info.com/default", "mail_exchanger": "exchange.info.com", "name": "mail.info.com", "preference": 1, "view": "default" } }		
NS	Delete	DELETE	<wapi_url>/record:mx/ZG5zLmJpbmRfbXgkLI9kZWZhdWx0LmNvbS5pbmZvLm1haWwuZXhjaGFuZ2UuaW5mby5jb20uMQ:mail.info.com/default	
		curl -k -u admin:Infoblox -H 'content-type: application/json' -X DELETE "https://127.0.0.1/wapi/v2.7/record:mx/ZG5zLmJpbmRfbXgkLI9kZWZhdWx0LmNvbS5pbmZvLm1haWwuZXhjaGFuZ2UuaW5mby5jb20uMQ:mail.info.com/default?_return_as_object=1"		
		{ "result": "record:mx/ZG5zLmJpbmRfbXgkLI9kZWZhdWx0LmNvbS5pbmZvLm1haWwuZXhjaGFuZ2UuaW5mby5jb20uMQ:mail.info.com/default" }		
	Fetch information	GET	<wapi_url>/record:ns	
		curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/record:ns?_return_as_object=1"		
		{ "result": { { "_ref": "record:ns/ZG5zLmJpbmRfbnMkLI9kZWZhdWx0LmFycGEuaW4tYWwRkci4xMjcuMC4wLi5jbHVzdGVy:cluster/0.0.127.in-addr.arpa/default", "name": "0.0.127.in-addr.arpa", "nameserver": "cluster", "view": "default" } }		

			MTAuMi5zZXJ2ZXluaW5mby5jb20:2.10.10.10.in-addr.arpa/default	
		curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/record:ptr/ZG5zLmJpbmRfcHRyJC5fZGVmYXVsdC5hcnBhLmluLWFKZHluMTAuMTAuMTAuMi5zZXJ2ZXluaW5mby5jb20:2.10.10.10.in-addr.arpa/default?_return_fields%2B=ptrdname&_return_as_object=1" -d '{"ptrdname":"server.info.com"}'		
		{"result": {"_ref": "record:ptr/ZG5zLmJpbmRfcHRyJC5fZGVmYXVsdC5hcnBhLmluLWFKZHluMTAuMTAuMTAuMi5zZXJ2ZXluaW5mby5jb20:2.10.10.10.in-addr.arpa/default", "ptrdname": "server.info.com", "view": "default" }}		
	Delete	DELETE	<wapi_url>/record:ptr/ZG5zLmJpbmRfcHRyJC5fZGVmYXVsdC5hcnBhLmluLWFKZHluMTAuMTAuMTAuMi5zZXJ2ZXluaW5mby5jb20:2.10.10.10.in-addr.arpa/default	
		curl -k -u admin:Infoblox -H 'content-type: application/json' -X DELETE "https://127.0.0.1/wapi/v2.7/record:ptr/ZG5zLmJpbmRfcHRyJC5fZGVmYXVsdC5hcnBhLmluLWFKZHluMTAuMTAuMTAuMi5zZXJ2ZXluaW5mby5jb20:2.10.10.10.in-addr.arpa/default?_return_as_object=1"		
		{"result": "record:ptr/ZG5zLmJpbmRfcHRyJC5fZGVmYXVsdC5hcnBhLmluLWFKZHluMTAuMTAuMTAuMi5zZXJ2ZXluaW5mby5jb20:2.10.10.10.in-addr.arpa/default"}		
TXT	Fetch information	GET	<wapi_url>/record:txt	
		curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/record:txt?_return_as_object=1"		
		{"result": [{"_ref": "record:txt/ZG5zLmJpbmRfdHh0JC5fZGVmYXVsdC5jb20udGVzdC5tYWlsLiJUaGlzIiAiYSIglm1haWwIICJzZXJ2ZXli:mail.test.com/default", "name": "mail.test.com", "text": "This a mail server", "view": "default" }}]		
	Create	POST	<wapi_url>/record:txt	{"name":"server.info.com", "text":"This a host server"}
		curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/record:txt?_return_fields%2B=name,text&_return_as_object=1" -d '{"name":"server.info.com","text":"This a host server"}'		
		{"result": {"_ref": "record:txt/ZG5zLmJpbmRfdHh0JC5fZGVmYXVsdC5jb20uaW5mby5zZXJ2ZXlulRoXMiIlCjhlIiAiaG9zdCglbnNlcnZlciI:server.info.com/default", "name": "server.info.com", "text": "This a host server", "view": "default" }}		
	Modify	PUT	<wapi_url>/record:txt/ZG5zLmJpbmRfdHh0JC5fZGVmYXVsdC5jb20uaW5mby5zZXJ2ZXlulRoX	{"text":"This is a test server"}

			MilCJhliAiaG9zdCiglNlcnZcil:server.info.com/default	
		<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/record:txt/ZG5zLmJpbmRfdHh0JC5fZGVmYXVsdC5jb20uaW5mby5zZXJ2ZXlullRoaXMilCJhliAiaG9zdCiglNlcnZcil:server.info.com/default?_return_fields%2B=text&_return_as_object=1" -d '{"text":"This a test server"}'</pre>		
		<pre>{ "result": { "_ref": "record:txt/ZG5zLmJpbmRfdHh0JC5fZGVmYXVsdC5jb20uaW5mby5zZXJ2ZXlullRoaXMilCJhliAidGVzdCiglNlcnZcil:server.info.com/default", "name": "server.info.com", "text": "This a test server", "view": "default" } }</pre>		
	Delete	DELETE	<wapi_url>/record:txt/ZG5zLmJpbmRfdHh0JC5fZGVmYXVsdC5jb20uaW5mby5zZXJ2ZXlullRoaXMilCJhliAiaG9zdCiglNlcnZcil:server.info.com/default	
<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X DELETE "https://127.0.0.1/wapi/v2.7/record:txt/ZG5zLmJpbmRfdHh0JC5fZGVmYXVsdC5jb20uaW5mby5zZXJ2ZXlullRoaXMilCJhliAidGVzdCiglNlcnZcil:server.info.com/default?_return_as_object=1"</pre>				
<pre>{ "result": "record:txt/ZG5zLmJpbmRfdHh0JC5fZGVmYXVsdC5jb20uaW5mby5zZXJ2ZXlullRoaXMilCJhliAidGVzdCiglNlcnZcil:server.info.com/default" }</pre>				
SRV	Fetch information	GET	<wapi_url>/record:srv	
		<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/record:srv?_return_as_object=1"</pre>		
		<pre>{ "result": { { "_ref": "record:srv/ZG5zLmJpbmRfc3J2JC5fZGVmYXVsdC5jb20udGVzdC9fdGNwLI9zc2gvMS8xMC8yMi90ZXN0LmNvbQ:ssh_tcp.test.com/default", "name": "_ssh_tcp.test.com", "port": 22, "priority": 1, "target": "test.com", "view": "default", "weight": 10 } } }</pre>		
	Create	POST	<wapi_url>/record:srv	<pre>{ "name": "_ssh_tcp.info.com", "port": 22, "priority": 10, "target": "info.com", "weight": 10 }</pre>
<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/record:srv?_return_fields%2B=name,target&_return_as_object=1" -d '{"name": "_ssh_tcp.info.com","port": 22,"priority": 10,"target": "info.com","weight": 10}'</pre>				
<pre>{ "result": { "_ref": "record:srv/ZG5zLmJpbmRfc3J2JC5fZGVmYXVsdC5jb20uaW5mby9fdGNwLI9zc2gvMTAvMTAvMjlvW5mb" } }</pre>				

		<pre>y5jb20:_ssh._tcp.info.com/default", "name": "_ssh._tcp.info.com", "port": 22, "priority": 10, "target": "info.com", "view": "default", "weight": 10 }}</pre>		
	Modify	PUT	wapi_url /record:srv/ZG5zLmJpbmRfc3J2JC5fZGVmYXVsdC5jb20uaW5mby9fdGNwLI9zc2gvMTAvMTAvMjlvaW5mby5jb20:_ssh._tcp.info.com/default	{"priority": 20}
		<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/record:srv/ZG5zLmJpbmRfc3J2JC5fZGVmYXVsdC5jb20uaW5mby9fdGNwLI9zc2gvMTAvMTAvMjlvaW5mby5jb20:_ssh._tcp.info.com/default?_return_fields%2B=priority&_return_as_object=1" -d '{"priority": 20}'</pre>		
		<pre>{"result": {"_ref": "record:srv/ZG5zLmJpbmRfc3J2JC5fZGVmYXVsdC5jb20uaW5mby9fdGNwLI9zc2gvMjAvMTAvMjlvaW5mby5jb20:_ssh._tcp.info.com/default", "name": "_ssh._tcp.info.com", "port": 22, "priority": 20, "target": "info.com", "view": "default", "weight": 10 }} }</pre>		
	Delete	DELETE	wapi_url /record:srv/ZG5zLmJpbmRfc3J2JC5fZGVmYXVsdC5jb20uaW5mby9fdGNwLI9zc2gvMjAvMTAvMjlvaW5mby5jb20:_ssh._tcp.info.com/default	
		<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X DELETE "https://127.0.0.1/wapi/v2.7/record:srv/ZG5zLmJpbmRfc3J2JC5fZGVmYXVsdC5jb20uaW5mby9fdGNwLI9zc2gvMjAvMTAvMjlvaW5mby5jb20:_ssh._tcp.info.com/default?_return_as_object=1"</pre>		
		<pre>{"result": "record:srv/ZG5zLmJpbmRfc3J2JC5fZGVmYXVsdC5jb20uaW5mby9fdGNwLI9zc2gvMjAvMTAvMjlvaW5mby5jb20:_ssh._tcp.info.com/default"} }</pre>		

Retrieve all the records in a zone

The API function allrecords returns the following types of records within a zone:

- record:a
- record:aaaa
- record:cname
- record:dname
- record:host
- record:host_ipv4addr
- record:host_ipv6addr
- record:mx
- record:naptr

- record:ptr
- record:srv
- record:txt
- sharedrecord:a
- sharedrecord:aaaa
- sharedrecord:mx
- sharedrecord:srv
- sharedrecord:txt

For the other types of records, it returns “UNSUPPORTED”

Operation	REST Method	API Call
	Sample cURL Command	
	Sample Output	
Retrieve all records in a zone	GET	<wapi_url>/allrecords?zone=info.com
	curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/allrecords?zone=info.com&_return_as_object=1"	
	<pre>{ "result": [{ "_ref": "allrecords/ZG5zLnpvbmVfc2VhcmNoX2luZGV4JGRucy5iaW5kX3NvYSQuX2RlZmF1bHQyY29tLmluZm8:", "comment": "Auto-created by Add Zone", "name": "", "type": "UNSUPPORTED", "view": "default", "zone": "info.com" }, { "_ref": "allrecords/ZG5zLnpvbmVfc2VhcmNoX2luZGV4JGRucy5iaW5kX25zJC5fZGVmYXVsdC5jb20uaW5mb2Jsb3gubG9jYWxkb21haW4:", "comment": "Auto-created by Add Zone", "name": "", "type": "UNSUPPORTED", "view": "default", "zone": "info.com" }, { "_ref": "allrecords/ZG5zLnpvbmVfc2VhcmNoX2luZGV4JGRucy5ob3N0JC5fZGVmYXVsdC5jb20uaW5mb2Jsb3N0:host", "comment": "", "name": "host", "type": "record:host_ipv4addr", "view": "default", "zone": "info.com" }, { "_ref": "allrecords/ZG5zLnpvbmVfc2VhcmNoX2luZGV4JGRucy5iaW5kX2EkL9kZWZhdWx0LmNvbS5pbmZvLHNlcnZlcwMC4xMC4xMC4y:server", "comment": "", "name": "server", "type": "record:a", "view": "default", "zone": "info.com" }], "_ref": "allrecords/ZG5zLnpvbmVfc2VhcmNoX2luZGV4JGRucy5iaW5kX2FhYWVhZm9kZWZhdWx0LmNvbS5pbmZvLHNlcnZlcwMC4xMC4xMC4y:server" }</pre>	

	<pre> mZDYwOmUzMjpmMWI5Ojoy:server", "comment": "", "name": "server", "type": "record:aaaa", "view": "default", "zone": "info.com" }, {"_ref": "allrecords/ZG5zLn timerVfc2VhcmNoX2luZGV4JGRucy5ob3N0JC5fZGVmYXVsdC5jb20uaW5mby50ZXN0:test", "comment": "", "name": "test", "type": "record:host_ipv4addr", "view": "default", "zone": "info.com" }}} </pre>
--	--

Retrieve non-system generated records

Records have a creator field. The possible values for this field are SYSTEM, STATIC and DYNAMIC. You can either fetch STATIC and DYNAMIC records separately, or combine them in a Multiple Object Body request as shown in the section Multiple Object Body Feature using */request*

Run a GET request on the records by passing creator as the parameter.

Operation	REST Method	API Call
	Sample cURL Command	
	Sample Output	
Retrieve non-system generated records	GET	<wapi_url>/record:a?creator=STATIC
	<pre> curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/record:a?creator=STATIC&_return_fields%2B=creator&_return_as_object=1" </pre>	
	<pre> {"result": [{"_ref": "record:a/ZG5zLn timerVfc2VhcmNoX2luZGV4JGRucy5ob3N0JC5fZGVmYXVsdC5jb20uaW5mby50ZXN0:test", "creator": "STATIC", "ipv4addr": "10.10.10.2", "name": "server.info.com", "view": "default" }, {"_ref": "record:a/ZG5zLn timerVfc2VhcmNoX2luZGV4JGRucy5ob3N0JC5fZGVmYXVsdC5jb20uaW5mby50ZXN0:test", "creator": "STATIC", "ipv4addr": "10.10.10.21", "name": "print.demo.com", "view": "default" }] } </pre>	

Search for a record based on some criteria

Run a GET request on host/A records by passing required criteria as the parameter

Operation	REST Method	API Call
	Sample cURL Command	
	Sample Output	
Retrieve host by object reference	GET	<wapi_url>/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS50ZXN0Lmhvc3Qx:host1.test.com/default
	curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS50ZXN0Lmhvc3Qx:host1.test.com/default?_return_as_object=1"	
	<pre>{ "result": { "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS50ZXN0Lmhvc3Qx:host1.test.com/default", "ipv4addrs": [{ "_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRkcmVzcyQuX2RlZmF1bHQyY29tLnRlc3QuaG9zdDEuMTcyLjI2LjEuMjAwLg:172.26.1.200/host1.test.com/default", "configure_for_dhcp": false, "host": "host1.test.com", "ipv4addr": "172.26.1.200" }], "name": "host1.test.com", "view": "default" } }</pre>	
Search host by name	GET	<wapi_url>/record:host?name=host1.info.com
	curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/record:host?name=host1.info.com&_return_as_object=1"	
	<pre>{ "result": [{ "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLmhvc3Qx:host1.info.com/default", "ipv4addrs": [{ "_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRkcmVzcyQuX2RlZmF1bHQyY29tLmluZm8uaG9zdDEuMTAuMTAuMTAuMjA:10.10.10.20/host1.info.com/default", "configure_for_dhcp": false, "host": "host1.info.com", "ipv4addr": "10.10.10.20" }], "name": "host1.info.com", "view": "default" }] }</pre>	
Search host by EA (Location is set to Santa Clara)	GET	<wapi_url>/record:host?_return_fields%2B=extattrs*&Location:=Santa%20Clara
	curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/record:host?_return_fields%2B=extattrs*&Location%3A=Santa%20Clara&_return_as_object=1"	
	<pre>{ "result": [{ "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS50ZXN0Lmhvc3Qx:host1.test.com/default", "extattrs": { "Location": { "value": "Santa Clara" } }, "ipv4addrs": [{ "_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRkcmVzcyQuX2RlZmF1bHQyY29tLnRlc3QuaG9zdDEuMTcyLjI2LjEuMjAwLg:172.26.1.200/host1.test.com/default", "configure_for_dhcp": false, "host": "host1.test.com", "ipv4addr": "172.26.1.200" }], "name": "host1.test.com", "view": "default" }] }</pre>	

	<pre>"record:host_ipv4addr/ZG5zLmhvc3RfYWVWRkcmVzcyQuX2RlZmF1bHQyY29tLnRlc3QuaG9zdDEuMTcyLjI2LjEuMjAwLg:172.26.1.200/host1.test.com/default", "configure_for_dhcp": false, "host": "host1.test.com", "ipv4addr": "172.26.1.200" }, "name": "host1.test.com", "view": "default" }}</pre>	
Get A record using object reference	GET	<a href="http://<wapi_url>/record:a/ZG5zLmJpbmRfYSQuX2RlZmF1bHQyY29tLmluZm8sc2VydmVyLDEwLjEwLjEwLjI:server.info.com/default"> <wapi_url>/record:a/ZG5zLmJpbmRfYSQuX2RlZmF1bHQyY29tLmluZm8sc2VydmVyLDEwLjEwLjEwLjI:server.info.com/default
	<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/record:a/ZG5zLmJpbmRfYSQuX2RlZmF1bHQyY29tLmluZm8sc2VydmVyLDEwLjEwLjEwLjI:server.info.com/default?_return_as_object=1"</pre>	
	<pre>{"result": {"_ref": "record:a/ZG5zLmJpbmRfYSQuX2RlZmF1bHQyY29tLmluZm8sc2VydmVyLDEwLjEwLjEwLjI:server.info.com/default", "ipv4addr": "10.10.10.2", "name": "server.info.com", "view": "default" } }</pre>	

The third example uses the `_return_fields+` to specify the Extensive Attribute(EA) as an extra attribute. Also, prepend `**` to the front of the EA name.

Search for network objects based on regular expressions

You can use regular expressions to search for most network objects.

The following example filters network objects based on a regular expression.

The first example lists all A records that contain *info.com* in their name. The second example lists all zones that start with an “i” and contain “lo”. The regular expression for this will be `^i.*lo.*$`

Remember to specify the `~` modifier to indicate you are querying with a regular expression.

Operation	REST Method	API Call
	Sample cURL Command	
	Sample Output	
Search for A records that contain info.com in the name	GET	<a href="http://<wapi_url>/record:a?name~=info.com&_return_fields=name"> <wapi_url>/record:a?name~=info.com&_return_fields=name
	<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/record:a?name~=info.com&_return_fields=name&_return_as_object=1"</pre>	
	<pre>{"result": [{"_ref": "record:a/ZG5zLmJpbmRfYSQuX2RlZmF1bHQyY29tLmluZm8sc2VydmVyLDEwLjEwLjEwLjI:server.info.com/default", "name": "server.info.com" }, {"_ref": "record:a/ZG5zLmJpbmRfYSQuX2RlZmF1bHQyY29tLmluZm8sdGVzdC5ob3N0LDE3Mi4yNi4xLjEwMA:host.test.info.com/default", "name": "host.test.info.com" } }]</pre>	

	<pre>"name": "host.test.info.com" }}}</pre>	
Search for zones that start with an "i" and contain "lo"	GET	<wapi_url>/zone_auth?fqdn=~^i.*lo.*\$&_return_fields=fqdn
	<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/zone_auth?fqdn=~^i.*lo.*\$&_return_fields=fqdn&_return_as_object=1"</pre>	
	<pre>{"result": [{"_ref": "zone_auth/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZvYmxveA:infoblox.com/default", "fqdn": "infoblox.com" }, {"_ref": "zone_auth/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZvcm1ibG94:informblox.com/default", "fqdn": "informblox.com" }, {"_ref": "zone_auth/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmdsbw:inglo.com/default", "fqdn": "inglo.com" }]}</pre>	

Aliases for a host

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Get all the aliases on a host	GET	<wapi_url>/record:host?_return_fields%2B=aliases&name=printer.info.com	
	<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/record:host?_return_fields%2B=aliases&name=printer.info.com&_return_as_object=1"</pre>		
	<pre>{"result": [{"_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLnByaW50ZXI:printer.info.com/default", "aliases": ["scanner.info.com"], "ipv4addrs": [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWVWRkcmVzcyQuX2RlZmF1bHQuY29tLmluZm8ucHJpbmRlci4xNjAuMjAuMS4xMD Au:160.20.1.100/printer.info.com/default", "configure_for_dhcp": false, "host": "printer.info.com", "ipv4addr": "160.20.1.100" }], "name": "printer.info.com", "view": "default" }]}</pre>		
Add a host with aliases	POST	<wapi_url>/record:host	<pre>{"name": "wapialias.info.com", "ipv4addrs": [{"ipv4addr": "172.26.1.2"}], "aliases": ["remote", "pointer"]}</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/record:host?_return_fields%2B=aliases&_return_as_object=1" -d '{"name": "wapialias.info.com", "ipv4addrs": [{"ipv4addr": "172.26.1.2"}], "aliases": ["remote", "pointer"]}'</pre>		

	<pre>{ "result": { "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLndhcGlhbGlhcw:wapialias.info.com/default", "aliases": ["remote.info.com", "pointer.info.com"], "ipv4addrs": { "_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRkcmVzcyQuX2RlZmF1bHQyY29tLmluZm8ud2FwaWFsaWFzLjE3Mi4yNi4xLjIu:172.26.1.2/wapialias.info.com/default", "configure_for_dhcp": false, "host": "wapialias.info.com", "ipv4addr": "172.26.1.2" }, "name": "wapialias.info.com", "view": "default" } }</pre>		
Modify aliases on a host	PUT	wapi_url/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLndhcGlhbGlhcw:wapialias.info.com/default	{"aliases": ["local"]}
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLndhcGlhbGlhcw:wapialias.info.com/default?_return_fields%2B=aliases&_return_as_object=1" -d '{"aliases": ["local"]}'</pre> <pre>{ "result": { "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLndhcGlhbGlhcw:wapialias.info.com/default", "aliases": ["local.info.com"], "ipv4addrs": { "_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRkcmVzcyQuX2RlZmF1bHQyY29tLmluZm8ud2FwaWFsaWFzLjE3Mi4yNi4xLjIu:172.26.1.2/wapialias.info.com/default", "configure_for_dhcp": false, "host": "wapialias.info.com", "ipv4addr": "172.26.1.2" }, "name": "wapialias.info.com", "view": "default" } }</pre>		
Remove aliases from a host	PUT	wapi_url/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLndhcGlhbGlhcw:wapialias.info.com/default	{"aliases": []}
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLndhcGlhbGlhcw:wapialias.info.com/default?_return_fields%2B=aliases&_return_as_object=1" -d '{"aliases": []}'</pre> <pre>{ "result": { "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLndhcGlhbGlhcw:wapialias.info.com/default", "ipv4addrs": { "_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRkcmVzcyQuX2RlZmF1bHQyY29tLmluZm8ud2FwaWFsaWFzLjE3Mi4yNi4xLjIu:172.26.1.2/wapialias.info.com/default", "configure_for_dhcp": false, "host": "wapialias.info.com", "ipv4addr": "172.26.1.2" }, "name": "wapialias.info.com", "view": "default" } }</pre>		

	<pre>"name": "wapialias.info.com", "view": "default" }}</pre>
--	---

Change the IP address of a host

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Change IP address of a host	PUT	<wapi_url>/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS50ZXN0Lmhvc3Qx:host1.test.com/default	<pre>{"ipv4addrs": [{"ipv4addr": "172.26.1.21"}]}</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS50ZXN0Lmhvc3Qx:host1.test.com/default?_return_fields%2B=ipv4addrs&_return_as_object=1" -d '{"ipv4addrs":[{"ipv4addr": "172.26.1.21"}]}'</pre>		
	<pre>{"result": { "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS50ZXN0Lmhvc3Qx:host1.test.com/default", "ipv4addrs": [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRRkcmVzcyQuX2RlZmF1bHQuY29tLnRlc3QuaG9zdDEuMTcyLjIjEuMjEu:172.26.1.21/host1.test.com/default", "configure_for_dhcp": false, "host": "host1.test.com", "ipv4addr": "172.26.1.21" }], "name": "host1.test.com", "view": "default" }}</pre>		

Add or remove IP addresses from a host

To add or remove IP address from a host record without altering the original list, use *ipv4addrs+* or *ipv4addrs-*

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Add IP address to a host record	PUT	<wapi_url>/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS50ZXN0Lmhvc3Qx:host1.test.com/default	<pre>{"ipv4addrs+": [{"ipv4addr": "192.168.1.21"}, {"ipv4addr": "10.10.10.21"}]}</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS50ZXN0Lmhvc3Qx:host1.test.com/default?_return_fields%2B=ipv4addrs&_return_as_object=1" -d '{"ipv4addrs+": [{"ipv4addr": "192.168.1.21"}, {"ipv4addr": "10.10.10.21"}]}'</pre>		
	<pre>{"result": { "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS50ZXN0Lmhvc3Qx:host1.test.com/default", "ipv4addrs": [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRRkcmVzcyQuX2RlZmF1bHQuY29tLnRlc3QuaG9zdDEuMTcyLjIjEuMjEu:192.168.1.21/host1.test.com/default", "configure_for_dhcp": false, "host": "host1.test.com", "ipv4addr": "192.168.1.21" }, [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRRkcmVzcyQuX2RlZmF1bHQuY29tLnRlc3QuaG9zdDEuMTcyLjIjEuMjEu:10.10.10.21/host1.test.com/default", "configure_for_dhcp": false, "host": "host1.test.com", "ipv4addr": "10.10.10.21" }], "name": "host1.test.com", "view": "default" }}</pre>		

	<pre> "record:host_ipv4addr/ZG5zLmhvc3RfYWVWRkcmVzcyQuX2RlZmF1bHQyY29tLnRlc3QuaG9zdDEuMTcyLjI2LjEuMjEu:172.26.1.21/host1.test.com/default", "configure_for_dhcp": false, "host": "host1.test.com", "ipv4addr": "172.26.1.21" }, {"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWVWRkcmVzcyQuX2RlZmF1bHQyY29tLnRlc3QuaG9zdDEuMTcyLjE2OC4xLjlxLg:192.168.1.21/host1.test.com/default", "configure_for_dhcp": false, "host": "host1.test.com", "ipv4addr": "192.168.1.21" }, {"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWVWRkcmVzcyQuX2RlZmF1bHQyY29tLnRlc3QuaG9zdDEuMTAuMTAuMTAuMjEu:10.10.10.21/host1.test.com/default", "configure_for_dhcp": false, "host": "host1.test.com", "ipv4addr": "10.10.10.21" }}, "name": "host1.test.com", "view": "default" }} </pre>		
Remove IP address from a host record	PUT	wapi_url>/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS50ZXN0Lmhvc3Qx:host1.test.com/default	<pre> {"ipv4addrs-": [{"ipv4addr": "192.168.1.21"}]} </pre>
	<pre> curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS50ZXN0Lmhvc3Qx:host1.test.com/default?t_return_fields%2B=ipv4addrs&_return_as_object=1" -d '{"ipv4addrs-":[{"ipv4addr":"192.168.1.21"}]}' </pre>		
	<pre> {"result": {"_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS50ZXN0Lmhvc3Qx:host1.test.com/default", "ipv4addrs": [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWVWRkcmVzcyQuX2RlZmF1bHQyY29tLnRlc3QuaG9zdDEuMTcyLjI2LjEuMjEu:172.26.1.21/host1.test.com/default", "configure_for_dhcp": false, "host": "host1.test.com", "ipv4addr": "172.26.1.21" }, {"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWVWRkcmVzcyQuX2RlZmF1bHQyY29tLnRlc3QuaG9zdDEuMTAuMTAuMTAuMjEu:10.10.10.21/host1.test.com/default", "configure_for_dhcp": false, "host": "host1.test.com", "ipv4addr": "10.10.10.21" }]}, "name": "host1.test.com", "view": "default" } } </pre>		

Add a host with the next available IP address from a network

Use the function *next_available_ip* by specifying it in the option *_function*. You can use it in the longhand form or the shorthand form(*func*). The following example also covers the different forms *func* supports in shorthand.

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Add host with next available IP address from a network	POST	<wapi_url>/record:host	<pre>{ "name": "wapi.info.com", "ipv4addrs": [{ "ipv4addr": { "_object_function": "next_available_ip", "_parameters": { "exclude": ["10.10.10.1", "10.10.10.2"] }, "_result_field": "ips", "_object": "network", "_object_parameters": { "network": "10.10.10.0/24" } } }] }</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST 'https://127.0.0.1/wapi/v2.7/record:host?_return_fields%2B=name,ipv4addrs&_return_as_object=1' -d '{ "name": "wapi.info.com", "ipv4addrs": [{ "ipv4addr": { "_object_function": "next_available_ip", "_parameters": { "exclude": ["10.10.10.1", "10.10.10.2"], "_result_field": "ips", "_object": "network", "_object_parameters": { "network": "10.10.10.0/24" } } } }] }'</pre>		
	<pre>{ "result": { "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLndhcGk:wapi.info.com/default", "ipv4addrs": [{ "_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRRkcmVzcyQuX2RlZmF1bHQyY29tLmluZm8ud2FwaS4xMC4xMC4xMC4xLg:10.10.10.3/wapi.info.com/default", "configure_for_dhcp": false, "host": "wapi.info.com", "ipv4addr": "10.10.10.3" }], "name": "wapi.info.com", "view": "default" } }</pre>		
	POST	<wapi_url>/record:host	<pre>{ "name": "wapi.info.com", "ipv4addrs": [{ "ipv4addr": { "func": "nextavailableip:10.10.10.0/24" } }] }</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST 'https://127.0.0.1/wapi/v2.7/record:host?_return_fields%2B=name,ipv4addrs&_return_as_object=1' -d '{ "name": "wapi.info.com", "ipv4addrs": [{ "ipv4addr": { "func": "nextavailableip:10.10.10.0/24" } }] }'</pre>		
	<pre>{ "result": { "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLndhcGk:wapi.info.com/default", "ipv4addrs": [{ "_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRRkcmVzcyQuX2RlZmF1bHQyY29tLmluZm8ud2FwaS4xMC4xMC4xMC4xLg:10.10.10.10/wapi.info.com/default", "configure_for_dhcp": false, "host": "wapi.info.com", "ipv4addr": "10.10.10.10" }], "name": "wapi.info.com", "view": "default" } }</pre>		
	POST	<wapi_url>/record:host	<pre>{ "name": "wapi.info.com", "ipv4addrs": [{ "ipv4addr": { </pre>

		"func:nextavailableip:network/ZG5zLm5ldHdvcmSkMTkyLjE2OC4xLjAvMjQvMA:192.168.1.0/24/default"]}]}
<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/record:host?_return_fields%2B=name,ipv4addrs&_return_as_object=1 " -d '{"name": "wapi.info.com", "ipv4addrs": [{"ipv4addr": "func:nextavailableip:network/ZG5zLm5ldHdvcmSkMTkyLjE2OC4xLjAvMjQvMA:192.168.1.0/24/default "}]}'</pre>		
<pre>{"result": {"_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLndhcGk:wapi.info.com/default", "ipv4addrs": [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRRkcmVzcyQuX2RlZmF1bHQyY29tLmluZm8ud2FwaS4xOTluMTY4LjEu MS4:192.168.1.5/wapi.info.com/default", "configure_for_dhcp": false, "host": "wapi.info.com", "ipv4addr": "192.168.1.5" }], "name": "wapi.info.com", "view": "default" } }</pre>		
POST	<wapi_url>/record:host	<pre>{"name": "wapi.info.com", "ipv4addrs": [{"ipv4addr": "func:nextavailableip:172.21.10.0/24,default"}]}</pre>
<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/record:host?_return_fields%2B=name,ipv4addrs&_return_as_object=1 " -d '{"name": "wapi.info.com", "ipv4addrs": [{"ipv4addr": "func:nextavailableip:172.21.10.0/24,default"}]}'</pre>		
<pre>{"result": {"_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLndhcGk:wapi.info.com/default", "ipv4addrs": [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRRkcmVzcyQuX2RlZmF1bHQyY29tLmluZm8ud2FwaS4xNzluMjEuMTAu MS4:172.21.10.1/wapi.info.com/default", "configure_for_dhcp": false, "host": "wapi.info.com", "ipv4addr": "172.21.10.1" }], "name": "wapi.info.com", "view": "default" } }</pre>		
POST	<wapi_url>/record:host	<pre>{"name": "wapi.info.com", "ipv4addrs": [{"ipv4addr": "func:nextavailableip: 192.168.1.10-192.168.1.15"}]}</pre>
<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/record:host?_return_fields%2B=name,ipv4addrs&_return_as_object=1 " -d '{"name": "wapi.info.com", "ipv4addrs": [{"ipv4addr": "func:nextavailableip: 192.168.1.10-192.168.1.15"}]}'</pre>		
<pre>{"result": {"_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLndhcGk:wapi.info.com/default", "ipv4addrs": [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRRkcmVzcyQuX2RlZmF1bHQyY29tLmluZm8ud2FwaS4xOTluMTY4LjEu MTAu:192.168.1.10/wapi.info.com/default", "configure_for_dhcp": false, "host": "wapi.info.com", "ipv4addr": "192.168.1.10" }], "name": "wapi.info.com", "view": "default" } }</pre>		

	<pre> "configure_for_dhcp": false, "host": "wapi.info.com", "ipv4addr": "192.168.1.10" }}, "name": "wapi.info.com", "view": "default" }} </pre>		
Add host with next available IP from a network using a complex search (using Extensible Attributes) and inherit the EAs	POST	<wapi_url>/record:host	<pre> {"name": "wapi.info.com", "ipv4addrs": [{"ipv4addr": {"_object_function": "next_available_ip", "_result_field": "ips", "_object": "network", "_object_parameters": {"Region": "EMEA"}, "_parameters": {"exclude": ["10.0.0.1", "10.0.0.2"]}}, "use_for_ea_inheritance": true}}] </pre>
	<pre> curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/record:host?_return_fields%2B=name,ipv4addrs&_return_as_object=1" -d '{"name": "wapi.info.com", "ipv4addrs": [{"ipv4addr": {"_object_function": "next_available_ip", "_result_field": "ips", "_object": "network", "_object_parameters": {"Region": "EMEA"}, "_parameters": {"exclude": ["10.0.0.1", "10.0.0.2"]}}, "use_for_ea_inheritance": true}}]' </pre>		
	<pre> {"result": {"_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLndhcGk:wapi.info.com/default", "ipv4addrs": [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRkcml9kZWZhdWx0LmNvbS5pbmZvLndhcGk:wapi.info.com/default", "configure_for_dhcp": false, "host": "wapi.info.com", "ipv4addr": "10.0.0.3" }], "name": "wapi.info.com", "view": "default" } </pre>		

Similarly, you can create an A record or AAAA record with the next available IP address from a network.

Add a host with a fixed address

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Add a host with a fixed address	POST	<wapi_url>/record:host	<pre> {"name": "host.info.com", "ipv4addrs": [{"ipv4addr": "172.26.1.20", "mac": "aa:bb:cc:11:22:21"}]} </pre>
	<pre> curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/record:host?_return_fields%2B=name,ipv4addrs&_return_as_object=1" -d '{"name": "host.info.com", "ipv4addrs": [{"ipv4addr": "172.26.1.20", "mac": "aa:bb:cc:11:22:21"}]}' </pre>		
	<pre> {"result": {"_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLmhvc3Q: host.info.com/default", "ipv4addrs": </pre>		

	<pre>{ "_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWVWRkcmVzcyQuX2RlZmF1bHQyY29tLmluZm8uaG9zdC4xNzluMjYyMS4yMC4:172.26.1.20/host.info.com/default", "configure_for_dhcp": true, "host": "host.info.com", "ipv4addr": "172.26.1.20", "mac": "aa:bb:cc:11:22:21" }, { "name": "host.info.com", "view": "default" } }</pre>
--	---

Add Extensible Attributes to an object

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Add an extensible attribute	PUT	<code><wapi_url>/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLndhcGk:wapi.info.com/default</code>	<pre>{ "extattrs": { "Site": { "value": "East" } } }</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLndhcGk:wapi.info.com/default?_return_fields%2B=extattrs&_return_as_object=1" -d '{"extattrs":{"Site": {"value": "East"}}}'</pre>		
	<pre>{ "result": { "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLndhcGk:wapi.info.com/default", "extattrs": { "Site": { "value": "East" } }, "ipv4addrs": [{ "_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWVWRkcmVzcyQuX2RlZmF1bHQyY29tLmluZm8ud2FwaS4xMC4wLjAuMy4:10.0.0.3/wapi.info.com/default", "configure_for_dhcp": false, "host": "wapi.info.com", "ipv4addr": "10.0.0.3" }], "name": "wapi.info.com", "view": "default" } }</pre>		

TTL

By default, the TTL is inherited from the Grid.

To override the default settings, while modifying TTL at the zone level, specify the TTL value in the `soa_default_ttl` field and also set the `use_grid_zone_timer` field to true. When this is set to `false`, the `soa_default_ttl` value indicates the default ttl. While modifying at the record level, remember to set the `use_ttl` flag to true. Otherwise, the value will be inherited from a higher-level setting.

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Fetch the default grid-wide TTL value	GET	<wapi_url>/grid:dns?_return_fields%2B=default_ttl	
	<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/grid:dns?_return_fields%2B=default_ttl&_return_as_object=1"</pre>		
	<pre>{"result": [{"_ref": "grid:dns/ZG5zLmNsdXN0ZXJfZG5zX3Byb3BlcnRpZXMkMA:Infoblox", "default_ttl": 28800 }]}</pre>		
Fetch TTL for a zone	GET	<wapi_url>/zone_auth?fqdn=info.com&_return_fields%2B=use_grid_zone_timer,soa_default_ttl	
	<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/zone_auth?fqdn=info.com&_return_fields%2B=use_grid_zone_timer,soa_default_ttl&_return_as_object=1"</pre>		
	<pre>{"result": [{"_ref": "zone_auth/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZv:info.com/default", "fqdn": "info.com", "soa_default_ttl": 28800, "use_grid_zone_timer": false, "view": "default" }]}</pre>		
Modify TTL for a zone	PUT	<wapi_url>/zone_auth/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZv:info.com/default	{"use_grid_zone_timer":true, "soa_default_ttl":5000}
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/zone_auth/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZv:info.com/default?_return_fields%2B=use_grid_zone_timer,soa_default_ttl&_return_as_object=1" -d '{"use_grid_zone_timer":true,"soa_default_ttl":5000}'</pre>		
	<pre>{"result": [{"_ref": "zone_auth/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZv:info.com/default", "fqdn": "info.com", "soa_default_ttl": 5000, "use_grid_zone_timer": true, "view": "default" }]}</pre>		
Modify TTL for a record	PUT	<wapi_url>/record:txt/ZG5zLmJpbmRfdHh0JC5fZGVmYXVsdC5jb20udGVzdC5tYWIsLiJUAaGzliAiYSIglm1haWwiiCjZlZjZl:mail.test.com/default	{"use_ttl": true, "ttl":3600}
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/record:txt/ZG5zLmJpbmRfdHh0JC5fZGVmYXVsdC5jb20udGVzdC5tYWIsLiJUAaGzliAiYSIglm1haWwiiCjZlZjZl:mail.test.com/default?_return_fields%2B=ttl&_return_as_object=1" -d '{"use_ttl": true,"ttl":3600}'</pre>		
	<pre>{"result": [{"_ref": "record:txt/ZG5zLmJpbmRfdHh0JC5fZGVmYXVsdC5jb20udGVzdC5tYWIsLiJUAaGzliAiYSIglm1haWwiiCjZlZjZl:m</pre>		

	<pre> ail.test.com/default", "name": "mail.test.com", "text": "This a mail server", "ttl": 3600, "view": "default" }} </pre>
--	--

Zone transfers

Zone transfers can be enabled and configured at the grid, member or zone level. While configuring at the zone level, in addition to the *allow_transfer* flag, set the *use_allow_transfer* flag.

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Enable and specify the zone transfer settings at the grid level	PUT	<pre> <wapi_url>/grid:dns/ZG5zLmNsdXN0ZXJfZG5zX3Byb3BlcnRpZXMkMA:Infoblox?_return_fields%2B=allow_transfer </pre>	<pre> {"allow_transfer": [{"_struct": "addressac", "address": "Any", "permission": "ALLOW"}]} </pre>
	<pre> curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/grid:dns/ZG5zLmNsdXN0ZXJfZG5zX3Byb3BlcnRpZXMkMA:Infoblox?_return_fields%2B=allow_transfer&_return_as_object=1" -d '{"allow_transfer": [{"_struct": "addressac", "address": "Any", "permission": "ALLOW"}]}' </pre>		
	<pre> {"result": {"_ref": "grid:dns/ZG5zLmNsdXN0ZXJfZG5zX3Byb3BlcnRpZXMkMA:Infoblox", "allow_transfer": [{"_struct": "addressac", "address": "Any", "permission": "ALLOW"}]} } </pre>		
Enable and specify the zone transfer settings at the member level	PUT	<pre> <wapi_url>/member:dns/ZG5zLm1lbWJlcl9kbmNfcHJvcGVydGllcyQx:member1.localdomain?_return_fields%2B=allow_transfer </pre>	<pre> {"allow_transfer": [{"_struct": "addressac", "address": "10.10.0.0/16", "permission": "ALLOW"}]} </pre>
	<pre> curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/member:dns/ZG5zLm1lbWJlcl9kbmNfcHJvcGVydGllcyQx:member1.localdomain?_return_fields%2B=allow_transfer&_return_as_object=1" -d '{"allow_transfer": [{"_struct": "addressac", "address": "10.10.0.0/16", "permission": "ALLOW"}]}' </pre>		
	<pre> {"result": {"_ref": "member:dns/ZG5zLm1lbWJlcl9kbmNfcHJvcGVydGllcyQx:member1.localdomain", "allow_transfer": [{"_struct": "addressac", "address": "10.10.0.0/16", "permission": "ALLOW"}]}, {"host_name": "member1.localdomain", "ipv4addr": "172.26.1.3"} } </pre>		

Enable and specify the zone transfer settings at the zone level	PUT	<wapi_url>/zone_auth/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5kZW1v:demo.com/default?_return_fields%2B=allow_transfer	{ "use_allow_transfer":true, "allow_transfer": [{"_struct": "addressac", "address": "192.168.0.1/24", "permission": "ALLOW"}]}
	curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/zone_auth/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5kZW1v:demo.com/default?_return_fields%2B=allow_transfer&_return_as_object=1" -d '{"use_allow_transfer":true,"allow_transfer": [{"_struct": "addressac", "address": "192.168.0.0/24", "permission": "ALLOW"}]}'		
	{ "result": { "_ref": "zone_auth/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5kZW1v:demo.com/default", "allow_transfer": [{ "_struct": "addressac", "address": "192.168.0.0/24", "permission": "ALLOW" }], "fqdn": "demo.com", "view": "default" }, }		

Response Policy Zones

DNS RPZs (Response Policy Zones), a technology developed by ISC (Internet System Consortium) for allowing reputable sources to dynamically communicate domain name reputation so you can implement policy controls for DNS lookups. You can configure RPZs and define RPZ rules to block DNS resolution for malicious or unauthorized domain names, or redirect clients to a walled garden by substituting responses. You can assign actions to RPZ rules.

For example, abc.com can have an action of pass thru or substitute (domain) with the domain xyz.com. You can also configure a Grid member to act as a lead secondary that receives RPZ updates from external reputation sources and redistributes the updates to other Grid members.

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Fetch RPZs	GET	<wapi_url>/zone_rp	
	curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/zone_rp?_return_fields%2B=rpz_policy&_return_as_object=1"		
	{ "result": [{"_ref": "zone_rp/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5nb29nbC53d3c:www.google.com/default", "fqdn": "www.google.com", "rpz_policy": "NXDOMAIN", "view": "default" },]}		
Create a RPZ	POST	<wapi_url>/zone_rp	{ "fqdn": "infoblow.com", "rpz_policy": "SUBSTITUTE", "rpz_severity": "WARNING", "substitute_name": "infoblox.com"} }

	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/zone_rp?_return_fields%2B=fqdn,rpz_policy,rpz_severity,substitute_name&_return_as_object=1" -d '{"fqdn": "infoblow.com","rpz_policy": "SUBSTITUTE","rpz_severity": "WARNING","substitute_name": "infoblox.com"}'</pre>		
	<pre>{ "result": { "_ref": "zone_rp/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5pbmZvYmxvdw:infoblow.com/default", "fqdn": "infoblow.com", "rpz_policy": "SUBSTITUTE", "rpz_severity": "WARNING", "substitute_name": "infoblox.com" } }</pre>		
Add a substitution rule for DNS A records	POST	<wapi_url>/record:rpz:a	<pre>{ "name": "server.infoblow.com", "ipv4addr": "1.1.1.156", "rp_zone": "infoblow.com" }</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/record:rpz:a?_return_fields%2B=name,rp_zone&_return_as_object=1" -d '{"name": "server.infoblow.com","ipv4addr": "1.1.1.156","rp_zone": "infoblow.com"}'</pre>		
	<pre>{ "result": { "_ref": "record:rpz:a/ZG5zLmJpbmRfYSQuX2RlZmF1bHQyY29tLmluZm9ibG93LHNlcnZlcwLjEuMS4xNTY:server.infoblow.com/default", "ipv4addr": "1.1.1.156", "name": "server.infoblow.com", "rp_zone": "infoblow.com", "view": "default" } }</pre>		
Add a substitution rule for DNS PTR records	POST	<wapi_url>/record:rpz:ptr	<pre>{ "name": "135.10.10.10.in-addr.arpa.infoblow.com", "ipv4addr": "10.10.10.135", "ptrdname": "infoblow.com", "rp_zone": "infoblow.com" }</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/record:rpz:ptr?_return_fields%2B=name,rp_zone&_return_as_object=1" -d '{"name": "135.10.10.10.in-addr.arpa.infoblow.com", "ipv4addr": "10.10.10.135", "ptrdname": "infoblow.com", "rp_zone": "infoblow.com"}'</pre>		
	<pre>{ "result": { "_ref": "record:rpz:ptr/ZG5zLmJpbmRfYHRyJC5fZGVmYXVsdC5jb20uaW5mb2Jsbn3cuYXJwYS5pb1hZGRyLjEwLjEwLjEwLjEzNS5pbmZvYmxvdj5jb20:135.10.10.10.in-addr.arpa.infoblow.com/default", "name": "135.10.10.10.in-addr.arpa.infoblow.com", "ptrdname": "infoblow.com", "rp_zone": "infoblow.com", "view": "default" } }</pre>		
Add a substitution rule for IP trigger policy	POST	<wapi_url>/record:rpz:a:ipaddress	<pre>{ "name": "1.1.1.6.infoblow.com", "ipv4addr": "2.2.2.6", "rp_zone": "infoblow.com" }</pre>

	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/record:rpz:a:ipaddress?_return_fields%2B=name,rp_zone&_return_as_object=1" -d '{"name":"1.1.1.6.infoblow.com","ipv4addr":"2.2.2.6","rp_zone":"infoblow.com"}'</pre> <pre>{ "result": { "_ref": "record:rpz:a:ipaddress/ZG5zLmJpbmRfYSQuX2RlZmF1bHQyY29tLmluZm9ibG93LHJwei1pcC4xLjEuMS42LjMyLDluMi4yLjY:1.1.1.6.infoblow.com/default", "ipv4addr": "2.2.2.6", "name": "1.1.1.6.infoblow.com", "rp_zone": "infoblow.com", "view": "default" } }</pre>		
Add a Block Domain Name (No Such Domain) Rule	POST	<wapi_url>/record:rpz:cname	<pre>{ "name": "test.infoblow.com", "canonical": "", "rp_zone": "infoblow.com" }</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/record:rpz:cname?_return_fields%2B=name,rp_zone&_return_as_object=1" -d '{"name":"test.infoblow.com","canonical":"","rp_zone":"infoblow.com"}'</pre> <pre>{ "result": { "_ref": "record:rpz:cname/ZG5zLmJpbmRfY25hbWUkLI9kZWZhdWx0LmNvbS5pbmZvYmxvdy50ZXN0:test.infoblow.com/default", "canonical": "", "name": "test.infoblow.com", "rp_zone": "infoblow.com", "view": "default" } }</pre>		
Add a Block Domain Name (No Data) Rule	POST	<wapi_url>/record:rpz:cname	<pre>{ "name": "demo.infoblow.com", "canonical": "", "rp_zone": "infoblow.com" }</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/record:rpz:cname?_return_fields%2B=name,rp_zone&_return_as_object=1" -d '{"name":"demo.infoblow.com","canonical":"","rp_zone":"infoblow.com"}'</pre> <pre>{ "result": { "_ref": "record:rpz:cname/ZG5zLmJpbmRfY25hbWUkLI9kZWZhdWx0LmNvbS5pbmZvYmxvdy5kZW1v:demo.infoblow.com/default", "canonical": "", "name": "demo.infoblow.com", "rp_zone": "infoblow.com", "view": "default" } }</pre>		
Add a Passthru Domain Name Rule	POST	<wapi_url>/record:rpz:cname	<pre>{ "name": "pass.infoblow.com", "canonical": "pass.infoblow.com", "rp_zone": "infoblow.com" }</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/record:rpz:cname?_return_fields%2B=name,rp_zone&_return_as_object=1" -d '{"name":"pass.infoblow.com","canonical":"pass.infoblow.com","rp_zone":"infoblow.com"}'</pre>		

Add a Block IP Address (No Data) Rule	POST	<wapi_url>/record:rpz:cname:ipaddress	{ "name": "5.5.5.25.infoblow.com", "canonical": "*", "rp_zone": "infoblow.com"} }
	curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/record:rpz:cname:ipaddress?_return_fields%2B=name,rp_zone&_return_as_object=1" -d '{ "name": "5.5.5.25.infoblow.com", "canonical": "*", "rp_zone": "infoblow.com"}'		
	{ "result": {"_ref": "record:rpz:cname:ipaddress/ZG5zLmJpbmRfY25hbWUkLi9kZWZhdWx0LmNvbS5pbmZvYmxvdy5ycHotaXAuNS41LjUuMjUuMzI:5.5.5.25.infoblow.com/default", "canonical": "*", "name": "5.5.5.25.infoblow.com", "rp_zone": "infoblow.com", "view": "default" }} }		
Add a Passthru IP Address Rule	POST	<wapi_url>/record:rpz:cname:ipaddress	{ "name": "5.5.5.100.infoblow.com", "canonical": "5.5.5.100", "rp_zone": "infoblow.com"} }
	curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/record:rpz:cname:ipaddress?_return_fields%2B=name,rp_zone&_return_as_object=1" -d '{ "name": "5.5.5.100.infoblow.com", "canonical": "5.5.5.100", "rp_zone": "infoblow.com"}'		
	{ "result": {"_ref": "record:rpz:cname:ipaddress/ZG5zLmJpbmRfY25hbWUkLi9kZWZhdWx0LmNvbS5pbmZvYmxvdy5ycHotaXAuNS41LjUuMTAwLjMy:5.5.5.100.infoblow.com/default", "canonical": "5.5.5.100", "name": "5.5.5.100.infoblow.com", "rp_zone": "infoblow.com", "view": "default" }} }		
Add a Substitute Domain Name (IP Address) Rule	POST	<wapi_url>/record:rpz:cname:ipaddressdn	{ "name": "5.5.5.200.infoblow.com", "canonical": "test", "rp_zone": "infoblow.com"} }
	curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/record:rpz:cname:ipaddressdn?_return_fields%2B=name,rp_zone&_return_as_object=1" -d '{ "name": "5.5.5.200.infoblow.com", "canonical": "test", "rp_zone": "infoblow.com"}'		
	{ "result": {"_ref": "record:rpz:cname:ipaddressdn/ZG5zLmJpbmRfY25hbWUkLi9kZWZhdWx0LmNvbS5pbmZvYmxvdy5ycHotaXAuNS41LjUuMjUuMzI:5.5.5.200.infoblow.com/default", "canonical": "test", "name": "5.5.5.200.infoblow.com", "rp_zone": "infoblow.com", "view": "default" }} }		
Fetch all objects belonging to a RPZ	GET	<wapi_url>/allrpzrecords?zone=infoblow.com	
	curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/allrpzrecords?zone=infoblow.com&_return_fields%2B=rpz_rule&_return_as_object=1"		

	<pre> {"result": [{"_ref": "allrpzrecords/ZG5zLnnpvbmVfc2VhcmNoX2luZGV4JGRucy5iaW5kX2EkLi9kZWZhdWx0LmNvbS5pbmZvYmxvdyxycH otaXAuMS4xLjEuNi4zMiwyljluMi42:1.1.1.6", "comment": "", "name": "1.1.1.6", "rpz_rule": "SubstitutelIPv4AddressRecord", "type": "record:rpz:a:ipaddress", "view": "default", "zone": "infoblow.com" }, {"_ref": "allrpzrecords/ZG5zLnnpvbmVfc2VhcmNoX2luZGV4JGRucy5iaW5kX2NuYW1lJC5fZGVmYXVsdC5jb20uaW5mb2Jsb3 cucnB6LWlwLjUuNS41LjEwLjMy:5.5.5.10", "comment": "", "name": "5.5.5.10", "rpz_rule": "BlockNxdomainIpaddr", "type": "record:rpz:cname:ipaddress", "view": "default", "zone": "infoblow.com" }, {"_ref": "allrpzrecords/ZG5zLnnpvbmVfc2VhcmNoX2luZGV4JGRucy5iaW5kX2NuYW1lJC5fZGVmYXVsdC5jb20uaW5mb2Jsb3 cucnB6LWlwLjUuNS41LjEwLjMy:5.5.5.25", "comment": "", "name": "5.5.5.25", "rpz_rule": "BlockNoDataIpaddr", "type": "record:rpz:cname:ipaddress", "view": "default", "zone": "infoblow.com" }, {"_ref": "allrpzrecords/ZG5zLnnpvbmVfc2VhcmNoX2luZGV4JGRucy5iaW5kX2NuYW1lJC5fZGVmYXVsdC5jb20uaW5mb2Jsb3 cucnB6LWlwLjUuNS41LjEwLjMy:5.5.5.100", "comment": "", "name": "5.5.5.100", "rpz_rule": "PassthruIpaddr", "type": "record:rpz:cname:ipaddress", "view": "default", "zone": "infoblow.com" }, {"_ref": "allrpzrecords/ZG5zLnnpvbmVfc2VhcmNoX2luZGV4JGRucy5iaW5kX2NuYW1lJC5fZGVmYXVsdC5jb20uaW5mb2Jsb3 cucnB6LWlwLjUuNS41LjEwLjMy:5.5.5.200", "comment": "", "name": "5.5.5.200", "rpz_rule": "SubstitutelIPAddressCname", "type": "record:rpz:cname:ipaddress", "view": "default", "zone": "infoblow.com" }, {"_ref": "allrpzrecords/ZG5zLnnpvbmVfc2VhcmNoX2luZGV4JGRucy5iaW5kX2NuYW1lJC5fZGVmYXVsdC5jb20uaW5mb2Jsb3 cucnB6LWlwLjUuNS41LjEwLjMy:10.10.10.200", "comment": "", "name": "10.10.10.200", </pre>
--	--

	<pre> "rpz_rule": "BlockNxdomainClientIpaddr", "type": "record:rpz:cname:clientipaddress", "view": "default", "zone": "infoblow.com" }, {"_ref": "allrpzrecords/ZG5zLnnpvbmVfc2VhcmNoX2luZGV4JGRucy5iaW5kX2NuYW11JC5fZGVmYXVsdC5jb20uaW5mb2Jsb3 cucnB6LWNsaWVudC1pcC4xMC4xMC4xMC4yMTAuMzI:10.10.10.210", "comment": "", "name": "10.10.10.210", "rpz_rule": "BlockNoDataClientIpaddr", "type": "record:rpz:cname:clientipaddress", "view": "default", "zone": "infoblow.com" }, {"_ref": "allrpzrecords/ZG5zLnnpvbmVfc2VhcmNoX2luZGV4JGRucy5iaW5kX2NuYW11JC5fZGVmYXVsdC5jb20uaW5mb2Jsb3 cucnB6LWNsaWVudC1pcC4xMC4xMC4xMC4yMjAuMzI:10.10.10.220", "comment": "", "name": "10.10.10.220", "rpz_rule": "PassthruClientIpaddr", "type": "record:rpz:cname:clientipaddress", "view": "default", "zone": "infoblow.com" }, {"_ref": "allrpzrecords/ZG5zLnnpvbmVfc2VhcmNoX2luZGV4JGRucy5iaW5kX2NuYW11JC5fZGVmYXVsdC5jb20uaW5mb2Jsb3 cucnB6LWNsaWVudC1pcC4xMC4xMC4xMC4yMzAuMzI:10.10.10.230", "comment": "", "name": "10.10.10.230", "rpz_rule": "SubstituteClientIpAddressCname", "type": "UNSUPPORTED", "view": "default", "zone": "infoblow.com" }, {"_ref": "allrpzrecords/ZG5zLnnpvbmVfc2VhcmNoX2luZGV4JGRucy5iaW5kX3B0ciQuX2RlZmF1bHQuY29tLmluZm9ibG93LmF ycGEuaW4tYWRkci4xMC4xMC4xMC4zMzUuaW5mb2Jsb3cuY29t:135.10.10.10.in-addr.arpa", "comment": "", "name": "135.10.10.10.in-addr.arpa", "rpz_rule": "SubstitutePTRRecord", "type": "record:rpz:ptr", "view": "default", "zone": "infoblow.com" }, {"_ref": "allrpzrecords/ZG5zLnnpvbmVfc2VhcmNoX2luZGV4JGRucy5iaW5kX2NuYW11JC5fZGVmYXVsdC5jb20uaW5mb2Jsb3 cuZGVtbw:demo", "comment": "", "name": "demo", "rpz_rule": "BlockNoDataDomain", "type": "record:rpz:cname", "view": "default", "zone": "infoblow.com" }, {"_ref": </pre>
--	--

	<pre> "allrpzrecords/ZG5zLnpvbmVfc2VhcmNoX2luZGV4JGRucy5iaW5kX2NuYW1lJC5fZGVmYXVsdC5jb20uaW5mb2Jsb3 cucGFzcw:pass", "comment": "", "name": "pass", "rpz_rule": "SubstituteCName", "type": "record:rpz:cname", "view": "default", "zone": "infoblow.com" }, {"_ref": "allrpzrecords/ZG5zLnpvbmVfc2VhcmNoX2luZGV4JGRucy5iaW5kX2EkLI9kZWZhdWx0LmNvbS5pbmZvYmxvdyxzZX J2ZXIsMS4xLjEuMTU2:server", "comment": "", "name": "server", "rpz_rule": "SubstituteARecord", "type": "record:rpz:a", "view": "default", "zone": "infoblow.com" }, {"_ref": "allrpzrecords/ZG5zLnpvbmVfc2VhcmNoX2luZGV4JGRucy5iaW5kX2NuYW1lJC5fZGVmYXVsdC5jb20uaW5mb2Jsb3 cuc3Vi:sub", "comment": "", "name": "sub", "rpz_rule": "SubstituteCName", "type": "record:rpz:cname", "view": "default", "zone": "infoblow.com" }, {"_ref": "allrpzrecords/ZG5zLnpvbmVfc2VhcmNoX2luZGV4JGRucy5iaW5kX2NuYW1lJC5fZGVmYXVsdC5jb20uaW5mb2Jsb3 cudGVzdA:test", "comment": "", "name": "test", "rpz_rule": "BlockNxdomainDomain", "type": "record:rpz:cname", "view": "default", "zone": "infoblow.com" }} } </pre>
--	---

DNS Traffic Control

Resource type	Operation	REST Method	API Call	Sample Body (for POST and PUT)
		Sample cURL Command		
		Sample Output		
DTC Servers	Fetch information	GET	<wapi_url>/dtc:server	
		curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/dtc:server?_return_as_object=1"		
		<pre> {"result": [{"_ref": "dtc:server/ZG5zLmlkbnNfc2VydjE:server1", </pre>		

		<pre> "host": "1.1.1.25", "name": "server1" }, {"_ref": "dtc:server/ZG5zLmlkbnNfc2VydmVyJGhvc3Q: host", "host": "10.10.10.20", "name": "host" }, {"_ref": "dtc:server/ZG5zLmlkbnNfc2VydmVyJFNfUIZFUI9URU1QTkFNRV8yMjEyNw: SERVER_TEMPNAME_22127", "host": "example.com", "name": "SERVER_TEMPNAME_22127" }} </pre>		
	Create	POST	<wapi_url>/dtc:server	<pre> {"name": "test-server", "host": "2.2.2.56"} </pre>
		<pre> curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/dtc:server?_return_fields%2B=name,host&_return_as_object=1" -d '{"name": "test-server","host": "2.2.2.56"}' </pre>		
		<pre> {"result": {"_ref": "dtc:server/ZG5zLmlkbnNfc2VydmVyJHRlc3Qtc2VydmVy:test-server", "host": "2.2.2.56", "name": "test-server" }} </pre>		
DTC Pools	Fetch information	GET	<wapi_url>/dtc:pool	
		<pre> curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/dtc:pool?_return_as_object=1" </pre>		
		<pre> {"result": [{"_ref": "dtc:pool/ZG5zLmlkbnNfcG9vbCRwb29sMQ: pool1", "name": "pool1" }, {"_ref": "dtc:pool/ZG5zLmlkbnNfcG9vbCRQT09MX1RFTVBOQU1FXzlyMTI3: POOL_TEMPNAME_22127", "name": "POOL_TEMPNAME_22127" }] } </pre>		
	Create	POST	<wapi_url>/dtc:pool	<pre> {"name": "test-pool", "availability": "ALL", "lb_preferred_method": "GLOBAL_AVAILABILITY", "monitors": ["dtc:monitor:icmp/ZG5zLmlkbnNfbW9uaXRvcj9pY21wJGljbXA: icmp"], "servers": [{"ratio": 4, "server": "dtc:server/ZG5zLmlkbnNfc2VydmVyJHRlc3Qtc2VydmVy:test-server"}]} </pre>
		<pre> curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/dtc:pool?_return_fields%2B=name,servers&_return_as_object=1" -d '{"name": "test-pool","availability": "ALL","lb_preferred_method": "GLOBAL_AVAILABILITY","monitors": ["dtc:monitor:icmp/ZG5zLmlkbnNfbW9uaXRvcj9pY21wJGljbXA: icmp"],"servers": [{"ratio": 4,"server": "dtc:server/ZG5zLmlkbnNfc2VydmVyJHRlc3Qtc2VydmVy:test-server"}]}' </pre>		

		<pre>{ "result": { "_ref": "dtc:pool/ZG5zLmlkbnNfcG9vbCR0ZXN0LXBvb2w:test-pool", "name": "test-pool", "servers": [{ "ratio": 4, "server": "dtc:server/ZG5zLmlkbnNfc2VydmVyJHRlc3Qtc2VydmVy:test-server" }] } }</pre>		
	Modify to add another server	PUT	<wapi_url>/dtc:pool/ZG5zLmlkbnNfcG9vbCR0ZXN0LXBvb2w:test-pool	<pre>{ "servers": [{ "ratio": 4, "server": "dtc:server/ZG5zLmlkbnNfc2VydmVyJHRlc3Qtc2VydmVy:test-server" }, { "ratio": 4, "server": "dtc:server/ZG5zLmlkbnNfc2VydmVyJHRlc3Qtc2VydmVyMg:test-server2" }] }</pre>
		<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/dtc:pool/ZG5zLmlkbnNfcG9vbCR0ZXN0LXBvb2w:test-pool?_return_fields%2B=name,servers&_return_as_object=1" -d '{ "servers": [{ "ratio": 4, "server": "dtc:server/ZG5zLmlkbnNfc2VydmVyJHRlc3Qtc2VydmVy:test-server" }, { "ratio": 4, "server": "dtc:server/ZG5zLmlkbnNfc2VydmVyJHRlc3Qtc2VydmVyMg:test-server2" }] }'</pre>		
		<pre>{ "result": { "_ref": "dtc:pool/ZG5zLmlkbnNfcG9vbCR0ZXN0LXBvb2w:test-pool", "name": "test-pool", "servers": [{ "ratio": 4, "server": "dtc:server/ZG5zLmlkbnNfc2VydmVyJHRlc3Qtc2VydmVy:test-server" }, { "ratio": 4, "server": "dtc:server/ZG5zLmlkbnNfc2VydmVyJHRlc3Qtc2VydmVyMg:test-server2" }] } }</pre>		
DTC Load Balanced Domain Name	Fetch information	GET	<wapi_url>/dtc:lbdn	
		<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/dtc:lbdn?_return_as_object=1"</pre>		
		<pre>{ "result": { "_ref": "dtc:lbdn/ZG5zLmlkbnNfbGJkbjRMQkROX1RFTVBOQU1FXzlyMTI3:LBDN_TEMPNAME_22127", "name": "LBDN_TEMPNAME_22127" } }</pre>		
	Create	POST	<wapi_url>/dtc:lbdn	<pre>{ "auth_zones": [{ "zone_auth/ZG5zLnpvbmUkLi9kZWZhdWx0LmNvbS5kdGM:dtc.com/default", "lb_method": "ROUND_ROBIN", "name": "test-lbdn", "patterns": ["server.dtc.com"], "pools": [{ "ratio": 4, "pool": "dtc:pool/ZG5zLmlkbnNfcG9vbCR0ZXN0LXBvb2w:test-pool" }] }] }</pre>
		<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/dtc:lbdn?_return_fields%2B=name,pools,auth_zones,patterns&_return_as_objec"</pre>		

		<pre>t=1" -d {'auth_zones':["zone_auth/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5kdGM:dtc.com/default"],"lb_method": "ROUND_ROBIN","name": "test-lbdn","patterns": ["server.dtc.com"],"pools": [{"ratio": 4,"pool":"dtc:pool/ZG5zLmlkbnNfcG9vbCR0ZXN0LXBvb2w:test-pool"}]}</pre>
		<pre>{"result": {"_ref": "dtc:lbdn/ZG5zLmlkbnNfbGJkbiR0ZXN0LWxiZG4:test-lbdn", "auth_zones": ["zone_auth/ZG5zLnpvbmUkLI9kZWZhdWx0LmNvbS5kdGM:dtc.com/default"], "name": "test-lbdn", "patterns": ["server.dtc.com"], "pools": [{"pool": "dtc:pool/ZG5zLmlkbnNfcG9vbCR0ZXN0LXBvb2w:test-pool", "ratio": 4 }]}}</pre>

IP address management related scenarios

Add a network or container

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Add a network	POST	<wapi_url>/network	<pre>{"network": "1.1.1.0/24"}</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/network?_return_fields%2B=network&_return_as_object=1" -d '{"network": "1.1.1.0/24"}'</pre>		
	<pre>{"result": {"_ref": "network/ZG5zLm5ldHdvcmskMS4xLjEuMC8yNC8w:1.1.1.0/24/default", "network": "1.1.1.0/24", "network_view": "default" }}</pre>		
Add a network and assign to a member	POST	<wapi_url>/network	<pre>{"network": "2.2.2.0/24", "members": [{"_struct": "dhcpmember", "ipv4addr": "172.26.1.3"}]}</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/network?_return_fields%2B=network,members&_return_as_object=1" -d '{"network": "2.2.2.0/24","members":[{"_struct": "dhcpmember","ipv4addr":"172.26.1.3"}]}'</pre>		
	<pre>{"result": {"_ref": "network/ZG5zLm5ldHdvcmskMi4yLjluMC8yNC8w:2.2.2.0/24/default", "members": [{"_struct": "dhcpmember", "ipv4addr": "172.26.1.3", "ipv6addr": null, "name": "member1.localdomain" }], }</pre>		

	<pre>"network": "2.2.2.0/24", "network_view": "default" }}</pre>		
Add a network and automatically create the corresponding reverse lookup zone	POST	<wapi_url>/network	<pre>{ "network": "10.10.10.0/24", "members": [{ "_struct": "dhcpmember", "ipv4addr": "172.26.1.2" }], "auto_create_reversezone": true }</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST 'https://127.0.0.1/wapi/v2.7/network?_return_fields%2B=network,members&_return_as_object=1' -d '{"network": "10.10.10.0/24","members":[{"_struct": "dhcpmember","ipv4addr":"172.26.1.2"},"auto_create_reversezone":true}'</pre>		
	<pre>{ "result": { "_ref": "network/ZG5zLm5ldHdvcmSkMS4xLjEuMCM8yNC8w:10.10.10.0/24/default", "members": [{ "_struct": "dhcpmember", "ipv4addr": "172.26.1.2", "ipv6addr": null, "name": "infoblox.localdomain" }], "network": "10.10.10.0/24", "network_view": "default" } }</pre>		
Add a network from next available network	POST	<wapi_url>/network	<pre>{ "network": "func:nextavailablenetwork:10.0.0.0/8,default,24", "network_view": "default", "extattrs": { "Country": { "value": "UK" }, "Region": { "value": "EMEA" } } }</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST 'https://127.0.0.1/wapi/v2.7/network?_return_fields%2B=network,members,extattrs&_return_as_object=1' -d '{"network": "func:nextavailablenetwork:10.0.0.0/8,default,24","network_view": "default","extattrs": {"Country":{"value":"UK"},"Region":{"value":"EMEA"}}'</pre>		
	<pre>{ "result": { "_ref": "network/ZG5zLm5ldHdvcmSkMTAuMCM4wLjAvMjQvMA:10.0.0.0/24/default", "extattrs": { "Country": { "value": "UK" }, "Region": { "value": "EMEA" } }, "members": [], "network": "10.0.0.0/24", "network_view": "default" } }</pre>		

The last example uses the function *next_available_network* by specifying it in the option *_function*, in the shorthand form.

Search for a network

Operation	REST Method	API Call
	Sample cURL Command	
	Sample Output	
Search for a network	GET	<wapi_url>/network?network=172.21.10.0/24
	curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/network?network=172.21.10.0/24&_return_as_object=1"	
	{"result": [{"_ref": "network/ZG5zLm5ldHdvcmSkMTcyLjlxLjEwLjAvMjQvMA:172.21.10.0/24/default", "network": "172.21.10.0/24", "network_view": "default"}]}	
Search for network(s) that begins with a certain value (192.168)	GET	<wapi_url>/network?network~192.168
	curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/network?network~192.168&_return_as_object=1"	
	{"result": [{"_ref": "network/ZG5zLm5ldHdvcmSkMTkyLjE2OC4xLjAvMjQvMA:192.168.1.0/24/default", "network": "192.168.1.0/24", "network_view": "default"}, {"_ref": "network/ZG5zLm5ldHdvcmSkMTkyLjE2OC4xMDEuMC8yNC8w:192.168.101.0/24/default", "network": "192.168.101.0/24", "network_view": "default"}]}	
Search for network(s) based on the netmask (/27)	GET	<wapi_url>/network?network~.0/27
	curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/network?network~.0/27&_return_as_object=1"	
	{"result": [{"_ref": "network/ZG5zLm5ldHdvcmSkMTAuMTAuMTAwLjAvMjcvMA:10.10.100.0/27/default", "network": "10.10.100.0/27", "network_view": "default"}, {"_ref": "network/ZG5zLm5ldHdvcmSkMS4xLjEuMC8yNy8x:1.1.1.0/27/demo", "network": "1.1.1.0/27", "network_view": "demo"}]}	
Search by EA (Networks in the UK)	GET	<wapi_url>/network?_return_fields%2B=extattrs*&Country:=UK
	curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/network?_return_fields%2B=extattrs*&Country%3A=UK&_return_as_object=1"	
	{"result": [{"_ref": "network/ZG5zLm5ldHdvcmSkMTAuMC4wLjAvMjQvMA:10.0.0.0/24/default", "extattrs": {"Country": {"value": "UK"}, "Region": {"value": "EMEA"}}, "network": "10.0.0.0/24", "network_view": "default"}]}	

	<pre>{ "_ref": "network/ZG5zLm5ldHdvcmSkMTAuMS4wLjAvMTYvMA:10.1.0.0/16/default", "extattrs": { "Country": { "value": "UK" } }, "network": "10.1.0.0/16", "network_view": "default" }</pre>
--	--

Get next available address from a subnet

Use the function *next_available_ip* by specifying it in the option *_function*. For this operation you would need the *_ref* of the subnet you want to query.

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Get the next available IP address in a subnet	POST	<wapi_url>/network/ZG5zLm5ldHdvcmSkMTcyLjlxLjEwLjAvMjQvMA:172.21.10.0/24/default?_function=next_available_ip	{"num":1}
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/network/ZG5zLm5ldHdvcmSkMTcyLjlxLjEwLjAvMjQvMA:172.21.10.0/24/default?_function=next_available_ip&_return_as_object=1" -d '{"num":1}'</pre>		
	<pre>{"result": {"ips": ["172.21.10.1"] } }</pre>		
Get the next 5 available IP addresses in a subnet	POST	<wapi_url>/network/ZG5zLm5ldHdvcmSkMTcyLjlxLjEwLjAvMjQvMA:172.21.10.0/24/default?_function=next_available_ip	{"num":5}
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/network/ZG5zLm5ldHdvcmSkMTcyLjlxLjEwLjAvMjQvMA:172.21.10.0/24/default?_function=next_available_ip&_return_as_object=1" -d '{"num":5}'</pre>		
	<pre>{"result": {"ips": ["172.21.10.1", "172.21.10.2", "172.21.10.3", "172.21.10.4", "172.21.10.5"] } }</pre>		

Get all the addresses based on a search criteria

Operation	REST Method	API Call
	Sample cURL Command	
	Sample Output	
Get addresses in a subnet	GET	<wapi_url>/ip4address?network=10.10.10.0/24

	<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/ipv4address?network=10.10.10.0/24&_return_as_object=1"</pre> <pre>{ "result": [{ "_ref": "ipv4address/Li5pcHY0X2FkZHJlc3MkMTAuMTAuMTAuMjUzLzA:10.10.10.1", "ip_address": "10.10.10.1", "is_conflict": false, "mac_address": "", "names": [], "network": "10.10.10.0/24", "network_view": "default", "objects": [], "status": "UNUSED", "types": [], "usage": [] }, { "_ref": "ipv4address/Li5pcHY0X2FkZHJlc3MkMTAuMTAuMTAuMjU0LzA:10.10.10.2", "ip_address": "10.10.10.2", "is_conflict": false, "mac_address": "", "names": [], "network": "10.10.10.0/24", "network_view": "default", "objects": [], "status": "UNUSED", "types": [], "usage": [] }, { "_ref": "ipv4address/Li5pcHY0X2FkZHJlc3MkMTAuMTAuMTAuMjU1LzA:10.10.10.255", "ip_address": "10.10.10.255", "is_conflict": false, "mac_address": "", "names": [], "network": "10.10.10.0/24", "network_view": "default", "objects": [], "status": "USED", "types": ["BROADCAST"], "usage": [] }] }</pre>		
Get addresses in a given range	<table border="1" data-bbox="474 1465 1500 1516"> <tr> <td data-bbox="474 1465 587 1516">GET</td><td data-bbox="587 1465 1500 1516">127.0.0.1/wapi/v2.7/ipv4address?ip_address=10.10.10.252&ip_address=10.10.10.254</td></tr> </table> <pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/ipv4address?ip_address%3E=10.10.10.252&ip_address%3C=10.10.10.254&_return_as_object=1"</pre> <pre>{ "result": [{ "_ref": "ipv4address/Li5pcHY0X2FkZHJlc3MkMTAuMTAuMTAuMjUyLzA:10.10.10.252", "ip_address": "10.10.10.252", "is_conflict": false, "mac_address": "", "names": [], "network": "10.10.10.0/24", "network_view": "default", "objects": [], </pre>	GET	127.0.0.1/wapi/v2.7/ipv4address?ip_address=10.10.10.252&ip_address=10.10.10.254
GET	127.0.0.1/wapi/v2.7/ipv4address?ip_address=10.10.10.252&ip_address=10.10.10.254		

	<pre> "status": "UNUSED", "types": [], "usage": [] }, {"_ref": "ipv4address/Li5pcHY0X2FkZHJlc3MkMTAuMTAuMTAuMjUzLzA:10.10.10.253", "ip_address": "10.10.10.253", "is_conflict": false, "mac_address": "", "names": [], "network": "10.10.10.0/24", "network_view": "default", "objects": [], "status": "UNUSED", "types": [], "usage": [] }, {"_ref": "ipv4address/Li5pcHY0X2FkZHJlc3MkMTAuMTAuMTAuMjU0LzA:10.10.10.254", "ip_address": "10.10.10.254", "is_conflict": false, "mac_address": "", "names": [], "network": "10.10.10.0/24", "network_view": "default", "objects": [], "status": "UNUSED", "types": [], "usage": [] } }}} </pre>
--	---

Zone associations on a network

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
List zone associations on a network	GET	<wapi_url>/network/ZG5zLm5ldHdvcmskMTAuMS4wLjAvMTYvMA:10.1.0.0/16/default	
	<pre> curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/network/ZG5zLm5ldHdvcmskMTAuMS4wLjAvMTYvMA:10.1.0.0/16/default?_return_fields%2B=zone_associations&_return_as_object=1" </pre>		
	<pre> {"result": {"_ref": "network/ZG5zLm5ldHdvcmskMTAuMS4wLjAvMTYvMA:10.1.0.0/16/default", "network": "10.1.0.0/16", "network_view": "default", "zone_associations": [{"fqdn": "info.com", "is_default": true, "view": "default"}]} } </pre>		

Add a zone association to a network	PUT	<wapi_url>/network/ZG5zLm5ldHdvcmSkMTcyLjlxLjEwLjAvMjQvMA:172.21.10.0/24/default	{ "zone_associations": [{"fqdn": "test.org", "is_default": true}] }
	curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/network/ZG5zLm5ldHdvcmSkMTcyLjlxLjEwLjAvMjQvMA:172.21.10.0/24/default?_return_fields%2B=zone_associations&_return_as_object=1" -d '{"zone_associations": [{"fqdn": "test.com", "is_default": true}]}'		
	{ "result": { "_ref": "network/ZG5zLm5ldHdvcmSkMTcyLjlxLjEwLjAvMjQvMA:172.21.10.0/24/default", "network": "172.21.10.0/24", "network_view": "default", "zone_associations": [{"fqdn": "test.com", "is_default": true, "view": "default"}] } }		

Add a fixed address

A fixed address is a specific IP address that a DHCP server always assigns when a lease request comes from a particular MAC address of the client.

For example, if you have a printer in your network, you can reserve a particular IP address to be assigned to it every time it is turned on.

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Add a fixed address	POST	<wapi_url>/fixedaddress	{ "ipv4addr": "192.168.1.100", "mac": "aa:bb:cc:11:22:21" }
	curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/fixedaddress?_return_fields%2B=ipv4addr,mac&_return_as_object=1" -d '{"ipv4addr": "192.168.1.100", "mac": "aa:bb:cc:11:22:21"}'		
	{ "result": { "_ref": "fixedaddress/ZG5zLmZpeGVkX2FkZlJlc3MkMTkyLjE2OC4xLjEwMC4wLi4:192.168.1.100/default", "ipv4addr": "192.168.1.100", "mac": "aa:bb:cc:11:22:21", "network_view": "default" } }		
Add a fixed address reservation	POST	<wapi_url>/fixedaddress	{ "ipv4addr": "172.26.1.200", "mac": "00:00:00:00:00:00" }
	curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/fixedaddress?_return_fields%2B=ipv4addr,mac&_return_as_object=1" -d '{"ipv4addr": "172.26.1.200", "mac": "00:00:00:00:00:00"}'		
	{ "result": { "_ref": "fixedaddress/ZG5zLmZpeGVkX2FkZlJlc3MkMTkyLjE2LjEuMjAwLjAuLg:172.26.1.200/default", "ipv4addr": "172.26.1.200", "mac": "00:00:00:00:00:00", } }		

	<pre>"network_view": "default" }}</pre>		
Add a fixed address with next available IP	POST	<wapi_url>/fixedaddress?_return_fields=ipv4addr	<pre>{ "ipv4addr": "func:nextavailableip:10.1.0.0/16", "mac": "ff:ff:ff:ff:ff:ff" }</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/fixedaddress?_return_fields%2B=ipv4addr,mac&_return_as_object=1" -d '{"ipv4addr": "func:nextavailableip:10.1.0.0/16", "mac": "ff:ff:ff:ff:ff:ff"}'</pre>		
	<pre>{ "result": { "_ref": "fixedaddress/ZG5zLmZpeGVkX2FkZHJlc3MkMTAuMS4wLjEuMC4u:10.1.0.1/default", "ipv4addr": "10.1.0.1", "mac": "ff:ff:ff:ff:ff:ff", "network_view": "default" } }</pre>		

Search for a fixed address by MAC

The following example uses the `_return_fields+` to add the mac address to the list of fields that are returned.

Operation	REST Method	API Call
	Sample cURL Command	
	Sample Output	
Search fixed address by MAC	GET	<wapi_url>/fixedaddress?_return_fields%2B=mac&mac=aa:bb:cc:11:22:21
	<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/fixedaddress?_return_fields%2B=mac&mac=aa:bb:cc:11:22:21&_return_as_object=1"</pre>	
	<pre>{ "result": [{ "_ref": "fixedaddress/ZG5zLmZpeGVkX2FkZHJlc3MkMTkyLjE2OC4xLjEwMC4wLjE4:192.168.1.100/default", "ipv4addr": "192.168.1.100", "mac": "aa:bb:cc:11:22:21", "network_view": "default" }] }</pre>	

Get information about an IP address

Operation	REST Method	API Call
	Sample cURL Command	
	Sample Output	
Query an IP address	GET	<wapi_url>/ipv4address?ip_address=10.10.10.10
	<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/ipv4address?ip_address=10.10.10.10&_return_as_object=1"</pre>	
	<pre>{ "result": [{ "_ref": "ipv4address/Li5pcHY0X2FkZHJlc3MkMTAuMTAuMTAuMTAvMA:10.10.10.10", </pre>	

	<pre> "ip_address": "10.10.10.10", "is_conflict": false, "mac_address": "", "names": [], "network": "10.10.10.0/24", "network_view": "default", "objects": [], "status": "UNUSED", "types": [], "usage": [] }}} </pre>
--	--

Search for any object with an IP address

Operation	REST Method	API Call
	Sample cURL Command	
	Sample Output	
Search objects with an IP	GET	<wapi_url>/ipv4address?status=USED&ip_address=10.10.10.20
	<pre> curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/ipv4address?status=USED&ip_address=10.10.10.20&_return_as_object=1" </pre>	
	<pre> {"result": [{"_ref": "ipv4address/Li5pcHY0X2FkZHJlc3MkMTAuMTAuMTAuMjAvMA:10.10.10.20", "ip_address": "10.10.10.20", "is_conflict": false, "mac_address": "", "names": ["host1.info.com"], "network": "10.10.10.0/24", "network_view": "default", "objects": ["record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLmhvc3Qx:host1.info.com/default"], "status": "USED", "types": ["HOST"], "usage": ["DNS"]] } </pre>	

Search for an IP address

Operation	REST Method	API Call
	Sample cURL Command	
	Sample Output	
Search an IP address	GET	<wapi_url>/search?address=172.26.1.100
	<pre> curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/search?address=172.26.1.100&_return_as_object=1" </pre>	
	<pre> {"result": [{"_ref": </pre>	

	<pre>"record:a/ZG5zLmJpbmRFYSQuX2RlZmF1bHQyY29tLmluZm8sdGVzdC5ob3N0LDE3Mi4yNi4xLjEwMA:host .test.info.com/default", "ipv4addr": "172.26.1.100", "name": "host.test.info.com", "view": "default" }, {"_ref": "network/ZG5zLm5ldHdvcmskMTcyLjI2LjEuMC8yNC8w:172.26.1.0/24/default", "network": "172.26.1.0/24", "network_view": "default" }} </pre>
--	---

Delete an IP address

Operation	REST Method	API Call
	Sample cURL Command	
	Sample Output	
Delete an IP	DELETE	<wapi_url>/ipv4address/Li5pcHY0X2FkZHJlc3MkMTAuMTAuMTAuMjAvMA:10.10.10.20
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X DELETE "https://127.0.0.1/wapi/v2.7/ipv4address/Li5pcHY0X2FkZHJlc3MkMTAuMTAuMTAuMjAvMA:10.10.10.20&_return_as_object=1"</pre>	
	<pre>{"result": "ipv4address/Li5pcHY0X2FkZHJlc3MkMTAuMTAuMTAuMjAvMA:10.10.10.20"}</pre>	

Network Views

DNS views provide the ability to serve one version of DNS data to one set of clients and another version to another set of clients. With DNS views, the appliance can provide a different answer to the same query, depending on the source of the query.

A network view is a single routing domain with its own networks and shared networks. A network view can contain both IPv4 and IPv6 networks. All networks must belong to a network view.

For example, network views can be used in cloud environments to isolate each of the tenant to their own respective view. This enables support of overlapping networks across different tenants, each in its own network view, thus enabling multi-tenancy in cloud environments.

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Get all DNS views	GET	<wapi_url>/view	
	<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/view?_return_as_object=1"</pre>		
	<pre>{"result": [{"_ref": "view/ZG5zLnZpZXckLi9kZWZhdWx0:default/true", "is_default": true, "name": "default" }, {"_ref": "view/ZG5zLnZpZXckLjE:default.test/false",</pre>		

	<pre> "is_default": false, "name": "default.test" }, {"_ref": "view/ZG5zLnZpZXckLjI:default.cloud/false", "is_default": false, "name": "default.cloud" }} </pre>		
Get authoritative zones for a DNS view	GET	wapi_url/zone_auth?view=default.test	
	<pre> curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/zone_auth?view=default.test&_return_as_object=1" </pre>		
	<pre> {"result": [{"_ref": "zone_auth/ZG5zLnZpZXckLjEub3JnLnRlc3Q:info.com/default.test", "fqdn": "info.com", "view": "default.test" }, {"_ref": "zone_auth/ZG5zLnZpZXckLjEub3JnLnRlc3Q:test.org/default.test", "fqdn": "test.org", "view": "default.test" }] } </pre>		
Get all records for a zone in a DNS view	GET	wapi_url/record:host?zone=info.com&view=default.test	
	<pre> curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/record:host?zone=info.com&view=default.test&_return_as_object=1" </pre>		
	<pre> {"result": [{"_ref": "record:host/ZG5zLmhvc3QkLjEub3JnLnRlc3Quc2VydjVz:server.info.com/default.test", "ipv4addrs": [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRkcmVzcyQuMS5jb20uaW5mb5zZXJ2ZXluMTAuMTAuMC4xMDAu10.10.0.100/server.info.com/default.test", "configure_for_dhcp": false, "host": "server.info.com", "ipv4addr": "10.10.0.100" }], "name": "server.info.com", "view": "default.test" }, {"_ref": "record:host/ZG5zLmhvc3QkLjEub3JnLnRlc3QubWFpbGVy:mailer.info.com/default.test", "ipv4addrs": [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRkcmVzcyQuMS5jb20uaW5mb5tYWlsZXluMTAuMS4xLjEwMC4:10.1.1.100/mailer.info.com/default.test", "configure_for_dhcp": false, "host": "mailer.info.com", "ipv4addr": "10.1.1.100", "mac": "aa:bb:cc:11:22:33" }], "name": "mailer.info.com", "view": "default.test" }] } </pre>		

Get all network views	GET	<wapi_url>/networkview	
	curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/networkview?_return_as_object=1"		
	<pre>{ "result": [{ "_ref": "networkview/ZG5zLm5ldHdvcmtfdmldyQw:default/true", "is_default": true, "name": "default" }, { "_ref": "networkview/ZG5zLm5ldHdvcmtfdmldyQx:test/false", "is_default": false, "name": "test" }, { "_ref": "networkview/ZG5zLm5ldHdvcmtfdmldyQy:cloud/false", "is_default": false, "name": "cloud" }] }</pre>		
Create a network view	POST	<wapi_url>/networkview	<code>{"name": "demo"}</code>
	curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/networkview?_return_fields%2B=name&_return_as_object=1" -d '{"name": "demo"}'		
	<pre>{ "result": { "_ref": "networkview/ZG5zLm5ldHdvcmtfdmldyQx:demo/false", "is_default": false, "name": "demo" } }</pre>		
Create an authoritative zone within a network view	POST	<wapi_url>/zone_auth	<code>{"fqdn": "infoblox.com", "view": "default.test"}</code>
	curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/zone_auth?_return_fields%2B=fqdn,network_view&_return_as_object=1" -d '{"fqdn": "infoblox.com", "view": "default.test"}'		
	<pre>{ "result": { "_ref": "zone_auth/ZG5zLnpvbWUkLjEuY29tLmluZm9ibG94:infoblox.com/default.test", "fqdn": "infoblox.com", "network_view": "test", "view": "default.test" } }</pre>		
Create a host record within a network view	POST	<wapi_url>/record:host	<code>{"name": "host.infoblox.com", "ipv4addrs": [{"ipv4addr": "10.10.10.20"}], "view": "default.test"}</code>
	curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/record:host?_return_fields%2B=name,network_view&_return_as_object=1" -d '{"name": "host.infoblox.com", "ipv4addrs": [{"ipv4addr": "10.10.10.20"}], "view": "default.test"}'		
	<pre>{ "result": { "_ref": "record:host/ZG5zLmhvc3QkLjEuY29tLmluZm9ibG94Lmhvc3Q:host.infoblox.com/default.test", } }</pre>		

	<pre> "ipv4addrs": [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRRkcmVzcyQuMS5jb20uaW5mb2Jsb3guaG9zdC4xMC 4xMC4xMC4yMC4:10.10.10.20/host.infoblox.com/default.test", "configure_for_dhcp": false, "host": "host.infoblox.com", "ipv4addr": "10.10.10.20" }], "name": "host.infoblox.com", "network_view": "test", "view": "default.test" }} </pre>		
Add a network within a network view	POST	<wapi_url>/network	<pre> {"network": "192.168.1.0/24", "network_view": "test"} </pre>
	<pre> curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/network?_return_fields%2B=network&_return_as_object=1" -d '{"network": "192.168.1.0/24","network_view": "test"}' </pre>		
	<pre> {"result": {"_ref": "network/ZG5zLm5ldHdvcm5kMTkyLjE2OC4xLjAvMjQvMQ:192.168.1.0/24/test", "network": "192.168.1.0/24", "network_view": "test" }} </pre>		

DHCP related scenarios

DHCP ranges

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Get DHCP ranges	GET	<wapi_url>/range	
	<pre> curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/range?_return_as_object=1" </pre>		
	<pre> {"result": [{"_ref": "range/ZG5zLmRoY3BfcmluZ2UkMTkyLjE2OC4xLjEwLzE5Mi4xNjguMS4xNS8vLzAv:192.168.1.10/192.168.1.15/default", "end_addr": "192.168.1.15", "network": "192.168.1.0/24", "network_view": "default", "start_addr": "192.168.1.10" }]} </pre>		
Add a DHCP range	POST	<wapi_url>/range	<pre> {"start_addr": "10.10.10.20", "end_addr": "10.10.10.100", "server_association_type": "MEMBER", "member":{"_struct": "dhcpmember", "ipv4addr": "172.26.1.3"}} </pre>

	<pre>"network": "10.10.10.0/24", "network_view": "test", "start_addr": "10.10.10.150" }}}</pre>
--	---

DHCP Leases

Operation	REST Method	API Call
	Sample cURL Command	
	Sample Output	
Get DHCP leases	GET	<wapi_url>/lease
	<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/lease?_return_as_object=1"</pre>	
	<pre>{ "result": [{ "_ref": "lease/ZG5zLmxlYXNIJDVMTAuMTAuMC4yMDAvMC8:10.10.10.200/test", "address": "10.10.10.200", "network_view": "test" }, { "_ref": "lease/ZG5zLmxlYXNIJDVMTAuMTAuMC4xOTUvMC8:192.168.1.100/default", "address": "192.168.1.100", "network_view": "default" }, { "_ref": "lease/ZG5zLmxlYXNIJDVMTAuMTAuMC4xOTEvMC8:10.10.10.191/default", "address": "10.10.10.191", "network_view": "default" }] }</pre>	
Get DHCP lease by IP Address	GET	<wapi_url>/lease?address=172.26.1.250&_return_fields=binding_state,hardware,client_hostname,fingerprint&_max_results=1
	<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/lease?address=172.26.1.250&_return_fields=binding_state,hardware,client_hostname,fingerprint&_max_results=1&_return_as_object=1"</pre>	
	<pre>{ "result": [{ "_ref": "lease/ZG5zLmxlYXNIJDQvMTcyLjE2LjIwMC4xMDEvMC8:172.26.1.250/default1", "binding_state": "ACTIVE", "fingerprint": "Cisco/Linksys SPA series IP Phone", "hardware": "16:55:a4:1b:98:c9" }] }</pre>	
Get a DHCP lease by hostname	GET	<wapi_url>/lease?client_hostname~=demo&_max_results=1&_return_fields=binding_state,hardware,client_hostname,fingerprint
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/lease?client_hostname~=demo&_max_results=1&_return_fields=binding_state,hardware,client_hostname,fingerprint&_return_as_object=1"</pre>	
	<pre>{ "result": [{ "_ref": "lease/ZG5zLmxlYXNIJC8xOTIuMTY4LjQuMy8wLzE3:192.168.4.3/Company%201", "binding_state": "STATIC", "client_hostname": "test", }] }</pre>	

	"hardware": "12:34:56:78:91:23" }}}
--	--

Grid management related scenarios

For these operations, you would need the `_ref` of the grid.

Operation	REST Method	API Call
	Sample cURL Command	
	Sample Output	
Grid reference	GET	<wapi_url>/grid
	curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/grid"	
	{ "result": [{"_ref": "grid/b25lLmNsdXN0ZXIkMA:Infoblox"}] } }}	

Grid DNS properties

The `grid:dns` object supports DNS service management and configuration such as time-to-live (TTL) settings, zone transfers, queries, root name servers, dynamic updates, sort lists, Transaction Signatures (TSIG) for DNS and others, all at the grid level.

The service configurations of a grid are inherited by all members, zones, and networks unless you specifically override them for selected members, zones, and networks. For this reason, it is recommended that you configure services at the grid level before configuring member, zone and network services.

The Grid Member DNS (`member:dns`) object can be used to configure DNS properties for a Grid member, including enabling or disabling DNS services and other DNS service related parameters. Remember to set the corresponding `use` flag to true. The value of the field will only take effect when its use flag is true. Otherwise, Grid service configurations are inherited by all members.

In the following examples, `max_cached_lifetime` indicates the maximum time (in seconds) a DNS response can be stored in the hardware acceleration cache.

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Get grid DNS properties	GET	<wapi_url>/grid:dns?_return_fields%2B=max_cached_lifetime	
	curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/grid:dns?_return_fields%2B=max_cached_lifetime&_return_as_object=1"		
	{ "result": [{"_ref": "grid:dns/ZG5zLmNsdXN0ZXJfZG5zX3Byb3BlcnRpZXMkMA:Infoblox", "max_cached_lifetime": 86400 }] } }}		

Modify grid DNS properties (max_cached_lifetime) at the Grid level	PUT	<wapi_url>/grid:dns/ZG5zLmNsdXN0ZXJfZG5zX3Byb3BlcnRpZXMkMA:Infoblox?_return_fields%2B=max_cached_lifetime	{"max_cached_lifetime":60000}
	curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/grid:dns/ZG5zLmNsdXN0ZXJfZG5zX3Byb3BlcnRpZXMkMA:Infoblox?_return_fields%2B=max_cached_lifetime&_return_as_object=1" -d '{"max_cached_lifetime":60000}'		
	{ "result": { "_ref": "grid:dns/ZG5zLmNsdXN0ZXJfZG5zX3Byb3BlcnRpZXMkMA:Infoblox", "max_cached_lifetime": 60000 } } }		
Get member DNS properties	GET	<wapi_url>/member:dns?_return_fields%2B=max_cached_lifetime	
	curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/member:dns?_return_fields%2B=max_cached_lifetime&_return_as_object=1"		
	{ "result": [{ "_ref": "member:dns/ZG5zLm1lbWJlcl9kbNfcHJvcGVydGllcyQw:infoblox.localdomain", "host_name": "infoblox.localdomain", "ipv4addr": "172.26.1.2", "max_cached_lifetime": 86400 }, { "_ref": "member:dns/ZG5zLm1lbWJlcl9kbNfcHJvcGVydGllcyQx:member1.localdomain", "host_name": "member1.localdomain", "ipv4addr": "172.26.1.3", "max_cached_lifetime": 86400 }] }		
Modify grid DNS properties(max_cached_lifetime) at the member level	PUT	<wapi_url>/member:dns/ZG5zLm1lbWJlcl9kbNfcHJvcGVydGllcyQx:member1.localdomain?_return_fields%2B=max_cached_lifetime	{"use_max_cached_lifetime": true, "max_cached_lifetime":80000}
	curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/member:dns/ZG5zLm1lbWJlcl9kbNfcHJvcGVydGllcyQx:member1.localdomain?_return_fields%2B=max_cached_lifetime&_return_as_object=1" -d '{"use_max_cached_lifetime": true,"max_cached_lifetime":80000}'		
	{ "result": [{ "_ref": "member:dns/ZG5zLm1lbWJlcl9kbNfcHJvcGVydGllcyQx:member1.localdomain", "host_name": "member1.localdomain", "ipv4addr": "172.26.1.3", "max_cached_lifetime": 80000 }] }		

Members

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Get members	GET	<wapi_url>/member	

	<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/member?_return_as_object=1"</pre>		
	<pre>{ "result": [{ "_ref": "member/b25lLnZpcnR1YWxfbm9kZSQw:infoblox.localdomain", "config_addr_type": "IPv4", "host_name": "infoblox.localdomain", "platform": "VNIOS", "service_type_configuration": "ALL_V4" }, { "_ref": "member/b25lLnZpcnR1YWxfbm9kZSQx:member1.localdomain", "config_addr_type": "IPv4", "host_name": "member1.localdomain", "platform": "VNIOS", "service_type_configuration": "ALL_V4" }]}</pre>		
Get node information for all members	GET	<wapi_url>/member?_return_fields%2B=node_info	
	<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/member?_return_fields%2B=node_info&_return_as_object=1"</pre>		
	<pre>{ "result": [{ "_ref": "member/b25lLnZpcnR1YWxfbm9kZSQw:infoblox.localdomain", "config_addr_type": "IPv4", "host_name": "infoblox.localdomain", "node_info": [{ "ha_status": "NOT_CONFIGURED", "hwid": "42017879A14D4EE9E59F6ABEC0E8F729", "hwmodel": "", "hwplatform": "VMWARE", "hwtype": "IB-V815", "lan2_physical_setting": { "auto_port_setting_enabled": true }, "lan_ha_port_setting": { "ha_port_setting": { "auto_port_setting_enabled": true, "speed": "10" }, "lan_port_setting": { "auto_port_setting_enabled": true } }, "mgmt_physical_setting": { "auto_port_setting_enabled": true }, "paid_nios": false, "physical_oid": "0", "service_status": [{ "description": "Running", "service": "NODE_STATUS", "status": "WORKING" }, { "description": "2% - Primary drive usage is OK.", "service": "DISK_USAGE", "status": "WORKING" }, { "description": "172.26.1.2", "service": "ENET_LAN", "status": "WORKING" }, { "description": "7% - Database capacity usage is OK.", "service": "DB_OBJECT", "status": "WORKING" }] }] }</pre>		

	<pre>{ "service": "HSM", "status": "UNKNOWN" }, { "service_type_configuration": "ALL_V4" }, ... }</pre>		
Enable DHCP Service on a member	PUT	<wapi_url>/member:dhcproperties/ZG5zLm1lbWJlcl9kaGNwX3Byb3BlcnRpZXMKMQ:member1.localdomain	{"enable_dhcp":true}
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X PUT "https://127.0.0.1/wapi/v2.7/member:dhcproperties/ZG5zLm1lbWJlcl9kaGNwX3Byb3BlcnRpZXMKMQ:member1.localdomain?_return_as_object=1" -d '{"enable_dhcp":true}'</pre>		
	<pre>{"result": "member:dhcproperties/ZG5zLm1lbWJlcl9kaGNwX3Byb3BlcnRpZXMKMQ:member1.localdomain"}</pre>		

Services

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
Get grid service status	POST	<wapi_url>/grid/b25lMnsdXN0ZXIkMA:Infoblox?_function=requestrestartservicestatus	{"service_option": "ALL"}
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/grid/b25lMnsdXN0ZXIkMA:Infoblox?_function=requestrestartservicestatus" -d '{"service_option": "ALL"}'</pre>		
Restart the services	POST	<wapi_url>/grid/b25lMnsdXN0ZXIkMA:Infoblox?_function=requestrestartservices	<pre>{ "member_order" : "SIMULTANEOUSLY", "service_option": "ALL" }</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/grid/b25lMnsdXN0ZXIkMA:Infoblox?_function= restartservices" -d '{"member_order" : "SIMULTANEOUSLY","service_option": "ALL"}'</pre>		
Restart the services when no restart action is required	POST	<wapi_url>/grid/b25lMnsdXN0ZXIkMA:Infoblox?_function=requestrestartservices	<pre>{ "member_order" : "SIMULTANEOUSLY", "restart_option": "FORCE_RESTART", "service_option": "ALL" }</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/grid/b25lMnsdXN0ZXIkMA:Infoblox?_function= restartservices" -d '{"member_order" : "SIMULTANEOUSLY","restart_option": "FORCE_RESTART","service_option": "ALL"}'</pre>		
Get the pending changes	GET	<wapi_url>/grid:service restart:request:changedobject	
	<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/grid:service restart:request:changedobject"</pre>		

In the above examples, in the *service_option* you can replace *ALL* with *DNS* or *DHCP*

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Get vDiscovery tasks	GET	<wapi_url>/vdiscoverytask	
	curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/vdiscoverytask?_return_as_object=1"		
	<pre>{ "result": [{ "_ref": "vdiscoverytask/ZG5zLmNkaXNjb3ZlcnlfdGFzayRBenVyZSB2RGZlY292ZXJ5:Azure%20vDiscovery", "name": "Azure vDiscovery", "state": "WARNING" }, { "_ref": "vdiscoverytask/ZG5zLmNkaXNjb3ZlcnlfdGFzayRBV1MgdKRpc2NvdmVyeQ:AWS%20vDiscovery", "name": "AWS vDiscovery", "state": "WARNING" }, { "_ref": "vdiscoverytask/ZG5zLmNkaXNjb3ZlcnlfdGFzayRWTXdhcmUgdKRpc2NvdmVyeQ:VMware%20vDiscovery", "name": "VMware vDiscovery", "state": "WARNING" }, { "_ref": "vdiscoverytask/ZG5zLmNkaXNjb3ZlcnlfdGFzayRPaGVuc3RhY2sgdkRpc2NvdmVyeQ:Openstack%20vDiscovery", "name": "Openstack vDiscovery", "state": "WARNING" }] }</pre>		
Create a vDiscovery Task to discover the AWS environment	POST	<wapi_url>/vdiscoverytask	<pre>{ "name": "ec2.eu-west-1", "driver_type": "AWS", "fqdn_or_ip": "ec2.eu-west-1.amazonaws.com", "username": "test", "password": "test", "member": "infoblox.localdomain", "port": 443, "protocol": "HTTPS", "auto Consolidate Cloud EA": true, "auto Consolidate Managed Tenant": true, "auto Consolidate Managed VM": true, "merge_data": true, "private_network_view": "default", "private_network_view_mapping_policy": "DIRECT", "public_network_view": "default", "public_network_view_mapping_policy": "DIRECT", "update_metadata": true, }</pre>

			<pre>"scheduled_run": {"_struct": "setting:schedule", "frequency": "HOURLY", "repeat": "RECUR", "minutes_past_hour": 1, "disable": true}}'</pre>
		<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/vdiscoverytask?_return_fields=name,driver_type&_return_as_object=1" -d '{"name": "ec2.eu-west-1", "driver_type": "AWS", "fqdn_or_ip": "ec2.eu-west-1.amazonaws.com", "username": "test", "password": "test", "member": "infoblox.localdomain", "port": 443, "protocol": "HTTPS", "auto Consolidate Cloud EA": true, "auto Consolidate Managed Tenant": true, "auto Consolidate Managed VM": true, "merge_data": true, "private_network_view": "default", "private_network_view_mapping_policy": "DIRECT", "public_network_view": "default", "public_network_view_mapping_policy": "DIRECT", "update_metadata": true, "scheduled_run": {"_struct": "setting:schedule", "frequency": "HOURLY", "repeat": "RECUR", "minutes_past_hour": 1, "disable": true}}'</pre>	
		<pre>{"result": {"_ref": "vdiscoverytask/ZG5zLmNkaXNjb3ZlcnldGFzayRIYzluZXUtd2VzdC0x:ec2.eu-west-1", "driver_type": "AWS", "name": "ec2.eu-west-1" }}</pre>	
Start a vDiscovery task	POST	<wapi_url>/vdiscoverytask/ZG5zLmNkaXNjb3ZlcnldGFzayRIYzluZXUtd2VzdC0x:ec2.eu-west-1?_function=vdiscover_control	<pre>{"action": "START"}</pre>
		<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/vdiscoverytask/ZG5zLmNkaXNjb3ZlcnldGFzayRIYzluZXUtd2VzdC0x:ec2.eu-west-1?_function=vdiscovery_control" -d '{"action": "START"}'</pre>	

Permissions

Operation	REST Method	API Call
	Sample cURL Command	
	Sample Output	
Get permissions	GET	<wapi_url>/permission
		<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/permission?_return_as_object=1"</pre>
		<pre>{"result": {"_ref": "permission/b25lLmhpZXJfcjVsZSQuY29tLmluZm9ibG94LmRucy56b25lJC4uLi4uY29tLmluZm9ibG94Lm9uZS5yb2xlJEROUyBBZG1pbi4:DNS%20Admin/WRITE", "permission": "WRITE", "resource_type": "VIEW", "role": "DNS Admin" }, {"_ref": "permission/b25lLmhpZXJfcjVsZSQuY29tLmluZm9ibG94LmRucy56b25lJC5zcmRfcjVzdC4uLmNvbS5pbmZvYmxveC5vbmUucm9sZSRETIMgQWRtaW4u:DNS%20Admin/WRITE", "permission": "WRITE", "resource_type": "SHARED_RECORD_GROUP",</pre>

	<pre> "role": "DNS Admin" }, {"_ref": "permission/b25lLmhpZXJfcVVsZSQuY29tLmluZm9ibG94LmRucy5uZXR3b3JrX3ZpZXdfcGFyZW50JC8uLi5jb20uaW5mb2Jsb3gub25lLnJvbGUkR3JpZCBZG1pbi5kbnMubXNfYWwRfc2l0ZXNfZG9tYWlu: Grid%20Admin/WRITE", "permission": "WRITE", "resource_type": "MS_ADSITES_DOMAIN", "role": "Grid Admin" }}} </pre>	
Get permissions for a role (DHCP Admin)	GET	wapi_url/permission?role=DHCP%20Admin
	<pre> curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/permission?role=DHCP%20Admin&_return_as_object=1" </pre>	
	<pre> {"result": [{"_ref": "permission/b25lLmhpZXJfcVVsZSQuY29tLmluZm9ibG94LmRucy5uZXR3b3JrX3ZpZXdfcGFyZW50JC8uLi5jb20uaW5mb2Jsb3gub25lLnJvbGUkREhDUCBBZG1pbi4:DHCP%20Admin/WRITE", "permission": "WRITE", "resource_type": "NETWORK_VIEW", "role": "DHCP Admin" }, {"_ref": "permission/b25lLmhpZXJfcVVsZSQuY29tLmluZm9ibG94LmRucy5kaGNwX21hY19maWw0ZXJfcGFyZW50JDAuLi5jb20uaW5mb2Jsb3gub25lLnJvbGUkREhDUCBBZG1pbi4:DHCP%20Admin/WRITE", "permission": "WRITE", "resource_type": "DHCP_MAC_FILTER", "role": "DHCP Admin" }, {"_ref": "permission/b25lLmhpZXJfcVVsZSQuY29tLmluZm9ibG94LmRucy5jbHVzdGVyX2RoY3BfcHJvcGVydGllcyQwLi4uY29tLmluZm9ibG94Lm9uZS5yb2xJERI1AgQWRtaW4u:DHCP%20Admin/WRITE", "permission": "WRITE", "resource_type": "GRID_DHCP_PROPERTIES", "role": "DHCP Admin" }]} </pre>	
Get permissions for a group	GET	wapi_url/permission?group=cloud-api-only
	<pre> curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/permission?group=cloud-api-only" </pre>	
	<pre> {"result": [{"_ref": "permission/b25lLmhpZXJfcVVsZSQuY29tLmluZm9ibG94LmRucy5jbHVzdGVyX2RoY3BfcHJvcGVydGllcyQwLi4uY29tLmluZm9ibG94Lm9uZS5hZG1pbl9ncm91cCQuY2xvdWQtYXBpLW9ubHku:cloud-api-only/WRITE", "group": "cloud-api-only", "permission": "WRITE", "resource_type": "GRID_DHCP_PROPERTIES" }, {"_ref": "permission/b25lLmhpZXJfcVVsZSQuY29tLmluZm9ibG94LmRucy50ZW5hbnRfcGFyZW50JGFsbC4uLmNvbS5pbmZvYmxveC5vbmUuYWwRtaW5fZ3JvdXAkLmNsb3VkLWFwaS1vbmx5Lg:cloud-api-only/WRITE", }]} </pre>	

	<pre> "group": "cloud-api-only", "permission": "WRITE", "resource_type": "TENANT" }, {"_ref": "permission/b25lLmhpZXJfcVVsZSQuY29tLmluZm9ibG94LmRucy5uZXR3b3JrX3ZpZXdfcGFyZW50J C8uLi5jb20uaW5mb2Jsb3gub25lLmFkbWluX2dyb3VwJC5jbG91ZC1hcGktb25seS4:cloud-api- only/WRITE", "group": "cloud-api-only", "permission": "WRITE", "resource_type": "NETWORK_VIEW" }, {"_ref": "permission/b25lLmhpZXJfcVVsZSQuY29tLmluZm9ibG94LmRucy5jbHVzdGVyX2Ruc19wcm9wZXJ0 aWVzJDAuLi5jb20uaW5mb2Jsb3gub25lLmFkbWluX2dyb3VwJC5jbG91ZC1hcGktb25seS4:cloud-api- only/WRITE", "group": "cloud-api-only", "permission": "WRITE", "resource_type": "GRID_DNS_PROPERTIES" }}} </pre>
--	---

Export a database backup

When you initiate a backup, you will get a URL that can be used to download the file directly, and a token for the download session. After downloading the backup file, remember to remove the stored file using the token.

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Initiate a backup	POST	<wapi_url>/fileop?_function=getgriddata	{"type": "BACKUP"}
	<pre> curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/fileop?_function=getgriddata" -d '{"type": "BACKUP"}' </pre>		
	<pre> {"token": "eJydUE1PwzAMvfuPjMvaplVWjdtQmYSEnmlD4mi1STos2iQkGdr+PQ4SnDhxSOT4fcR+Sjl/w2DO\nwJdyNqZwUck F8ALuFNnB9aO7Fs6afNLNmwhltzo8mgF8DQqxv9CYyCKCJpXAL+BO+yWcZubqKdww\n0WRm4CXshKxkl+W6Xh WyyqZayhniaXcLi6lr5byn5eF+WoqkLpoiLnMLNfFoCQcaDZlrg/IA0vP2\n8Lp/PmzbuaqqUWqe+ikfrune2a7EyaqzU b/6VgaJNJWci1qJhXRt68O5syTf7fXwvxY4vGKqfJ\nnnnO3Zv+Hp30uF7+EyekcjchptduXLR4fd/ktlaajF6ucpGgYHMiM OmJyqNzku/CtWsM+b9l3Hsn6\nS8JPEyI5m7ENY33xBbQAIIA=\n", "url": "https://127.0.0.1/http_direct_file_io/req_id-DOWNLOAD-1001/database.bak" } </pre>		
Download the backup	GET	<url_from_the_previous_output> Example: https://x.x.x.x/http_direct_file_io/req_id-DOWNLOAD-1001/database.bak	
	<pre> curl -k -u admin:Infoblox -H 'content-type: application/force-download' "https://127.0.0.1/http_direct_file_io/req_id-DOWNLOAD-1001/database.bak" -o "database.bak" </pre>		
	#database.bak file is saved in the present working directory#		
	POST	<wapi_url>/fileop?_function=downloadcomplete	{"token": "eJydU..A="}

Remove the stored file using the token	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/fileop?_function=downloadcomplete" -d '{"token": "eyJyUE1PwzAMvfuPjMvaplVWjdtQmYSEnmlD4mi1STos2iQkGdr+PQ4SnDhxSOT4fcR+Sjl/w2DO\nwJdyNqZwUck F8ALuFNnB9aO7Fs6afNLNmwhltzo8mgF8DQqxv9CYyCKCJpXAL+BO+yWcZubqKdww\n0WRm4CXshKxkl+W6Xh WYqZayhniaXcLI6lr5byn5eF+WoqkLpoiLnMLNfFoCQcaDZlrg/IA0vP2\n8Lp/PmzbuagQUWqege+iKfrune2a7EyaqzU b/6VgaJNJWci1qJhXRt68O5syTf7fXwvxY4vGKqfJ\nnnnO3Zv+Hp30uF7+EyekcjchptduXLR4fd/ktlaajF6ucpGgYHMiM OmJyqNzku/CtWsM+b9l3Hsn6\nS8JPEyI5m7ENY33xBbQAIIA=\n"}'</pre>
--	---

Please note that while downloading the backup (in the second step above), the *Content-Type* in the header will be *application/force-download*.

Export a CSV file

When you initiate a CSV Export, you will get a URL that can be used to download the file directly, and a token for the download session. After downloading the backup file, remember to remove the stored file using the token.

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Initiate a CSV export	POST	<wapi_url>/fileop?_function=csv_export	{"_object": "record:host"}
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/fileop?_function=csv_export" -d '{"_object": "record:host"}</pre>		
	<pre>{"token": "eyJyUE1PwzAMvfuPjMvaplVWjdtQmYSEnmlD4mi1STos2iQkGdr+PQ4SnDhxSOT4fcR+Sjl/w2DO\nwJdyNqZwUck F8ALuFNnB9aO7Fs6afNLNmwhltzo8mgF8DQqxv9CYyCKCJpXAL+BO+yWcZubqKdww\n0WRm4CXshKxkl+W6Xh WYqZayhniaXcLI6lr5byn5eF+WoqkLpoiLnMLNfFoCQcaDZlrg/IA0vP2\n8Lp/PmzbuagQUWqege+iKfrune2a7EyaqzU b/6VgaJNJWci1qJhXRt68O5syTf7fXwvxY4vGKqfJ\nnnnO3Zv+Hp30uF7+EyekcjchptduXLR4fd/ktlaajF6ucpGgYHMiM OmJyqNzku/CtWsM+b9l3Hsn6\nS8JPEyI5m7ENY33xBbQAIIA=\n"}'</pre>		
Download the CSV file	GET	<url_from_the_previous_output> Example: https://x.x.x.x/http_direct_file_io/req_id-DOWNLOAD-1001/Hostrecords.csv	
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/force-download' "https://127.0.0.1/http_direct_file_io/req_id-DOWNLOAD-1001/Hostrecords.csv" -o "Hostrecords.csv"</pre>		
	#Hostrecords.csv file is saved in the present working directory#		
Remove the stored file using the token	POST	<wapi_url>/fileop?_function=downloadcomplete	{"token": "eyJyUE1PwzAMvfuPjMvaplVWjdtQmYSEnmlD4mi1STos2iQkGdr+PQ4SnDhxSOT4fcR+Sjl/w2DO\nwJdyNqZwUckF8ALuFNnB9aO7Fs6afNLNmwhltzo8mgF8DQqxv9CYyCKCJpXAL+BO+yWcZubqKdww\n0WRm4CXshKxkl+W6XhWYqZayhniaXcLI6lr5byn5eF+WoqkLpoiLnMLNfFoCQcaDZlrg/IA0vP2\n8Lp/PmzbuagQUWqege+iKfrune2a7EyaqzUb/6VgaJNJWci1qJhXRt68O5syTf7fXwvxY4vGKqfJ\nnnnO3Zv+Hp30uF7+EyekcjchptduXLR4fd/ktlaajF6ucpGgYHMiM\nOmJyqNzku/CtWsM+b9l3Hsn6\nS8JPEyI5m7ENY33xBbQAIIA=\n"}'}
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/fileop?_function=downloadcomplete" -d '{"token": "eyJyUE1PwzAMvfuPjMvaplVWjdtQmYSEnmlD4mi1STos2iQkGdr+PQ4SnDhxSOT4fcR+Sjl/w2DO\nwJdyNqZwUck F8ALuFNnB9aO7Fs6afNLNmwhltzo8mgF8DQqxv9CYyCKCJpXAL+BO+yWcZubqKdww\n0WRm4CXshKxkl+W6Xh WYqZayhniaXcLI6lr5byn5eF+WoqkLpoiLnMLNfFoCQcaDZlrg/IA0vP2\n8Lp/PmzbuagQUWqege+iKfrune2a7EyaqzU b/6VgaJNJWci1qJhXRt68O5syTf7fXwvxY4vGKqfJ\nnnnO3Zv+Hp30uF7+EyekcjchptduXLR4fd/ktlaajF6ucpGgYHMiM OmJyqNzku/CtWsM+b9l3Hsn6\nS8JPEyI5m7ENY33xBbQAIIA=\n"}'</pre>		

Please note that while downloading the CSV file (in the second step above), the *Content-Type* in the header will be *application/force-download*.

Export results of a WAPI call

You can cache the results by exporting the results of a WAPI call and save the data to disk.

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Export results of a WAPI call	POST	<code><wapi_url>/fileop?_function=read</code>	<pre>{ "_encoding": "JSON", "_filename": "allhosts.info.com.json", "_object": "record:host", "_return_fields": "name,extattrs", "zone": "info.com" }</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type: application/json' -X POST "https://127.0.0.1/wapi/v2.7/fileop?_function=read" -d '{"_encoding": "JSON", "_filename": "allhosts.info.com.json", "_object": "record:host", "_return_fields": "name,extattrs", "zone": "info.com"}'</pre>		

The file will be in a folder called 'wapi_output' in the HTTP file distribution.

Import a file (Certificate)

When you initiate a file import, you will get a URL to which you can upload the file to. Once you have uploaded the file, signal to the appliance to perform an action on the uploaded file. In this example, we will upload a certificate.

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Initiate file upload	POST	<code><wapi_url>/fileop?_function=uploadinit</code>	
	<pre>curl -k -u admin:Infoblox -H 'content-type:application/json' -X POST "https://127.0.0.1/wapi/v2.7/fileop?_function=uploadinit"</pre> <pre>{ "token": "eJydUMFOWzAMvftHymVt09Fu5TY0JiGhgTY4W2SDktYpIUbX9PMgQXbhwS2X7PL3IPSssXdPoE\n8ZLW+OBm GawDFnAjqY2H+05t0anEy6sPWy70OFBD8AVSMR+pjGQQQRFMgAv4UbxLRwzfWZyFww0\n6Qy4hp2oy3pVN1 XT5GXZtmIF/pjNboxoE/nvIbC/KwqxqvKqyUvEFWmEiuLXAg40aiRbOP2BpBZv\nL0/Pm+1CIKUoaGLrvhIR6ypLKlbr qPqXHoE2UVT0EWtRRlbo+nuplsw8f9eFeJHE7WRVpE5pWkV\nxe8f96lc/hlmq64rKaXt5nWDh4dd6mvw4cCiSQn Gdl7ZQHpUHoNFaSFu3HVRDftksO8YyfAc8FM7\nT9Ykr1Yn38Bt1KUqw==\n", "url": "https://172.26.1.2/http_direct_file_io/req_id-UPLOAD-1001/import_file" }</pre>		
Upload contents of a CA certificate	POST	<code><url_from_the_previous_output></code> Example: <code>https://x.x.x.x/http_direct_file_io/req_id-UPLOAD-1001/import_file</code>	<code>file=@ca.cert.pem</code>
	<pre>curl -k -u admin:Infoblox -H 'content-type:multipart-formdata' "https://127.0.0.1/http_direct_file_io/req_id-UPLOAD-1001/import_file" -F file=@ca.cert.pem</pre>		

Upload the CA Certificate	POST	<wapi_url>/fileop?_function=uploadcertificate	{ "certificate_usage": "EAP_CA", "member": "infoblox.localdomain", "token": "eJydk..==" }
	curl -k -u admin:Infoblox -H 'content-type:application/json' -X POST "https://127.0.0.1/wapi/v2.7/fileop?_function=uploadcertificate" -d '{"certificate_usage": "EAP_CA", "member": "infoblox.localdomain", "token": "eJydUMFOWzAMvftHymVt09Fu5TY0JiGhgTY4W22SDktYpIUbX9PMgQXbhwS2X7 PL3IPSssXdPoE\n8ZLW+OBmGawDFnAjqY2H+05t0anEy6sPWY70OFBD8AVSMR+pjGQQQRFMGAv4UbxLRwzf WZyFww0\n6Qy4hp2oy3pVN1XT5GXZtmIF/pjNboxoE\nvIbC/KwqxqvKqyUVeFWmEiuLXAg40aiRbOP2BpBZv\nL0/ Pm+1CIKUoaGLrvhIR6ypLKlbrqPqXHoE2UVT0EWtRRlIbho+nuplsw8f9eFeJHE7WRVpE5pWkV\nxe8f96lc/hlmq64r KaXt5nWDh4dd6mw4cCiSQnGdl7ZQHpUHoNFaFsu3HvRdfkO8YfAc8FM7\nT9Ykr1Yn38Bt1KUqw==\n"}'		

Please note that while uploading the file (in the second step above), the Content-Type in the header will be *multipart/form-data*

Import a CSV file

When you need to update or create multiple records, you can store the data as fields in a CSV file and import it at once. This will be faster in comparison with updating or creating each object with a separate API call. This example assumes that the values and fields in your CSV file are comma-separated. For more information on the CSV file, you can refer to the CSV Import Reference section under Tech Docs in the [Support Site](#).

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Initiate file upload	POST	<wapi_url>/fileop?_function=uploadinit	
	curl -k -u admin:Infoblox -H 'content-type:application/json' -X POST "https://127.0.0.1/wapi/v2.7/fileop?_function=uploadinit"		
	<pre>{ "token": "eJydUE1PwzAMvfuPbJe1Tbe2G7ehMQkJDbTB2WqTdfHqk5CkaPv3OENw4cYhkeP34TxLad0VvT4D\nX9KaEP0k o/XgBMwlmd52g71k1uh04tXpALs2tnjUPbgSJGI30RDJIIIiGcEtYa7cCk4zfXHkrxhp\n1DNwFexFVVRNVddV4l6tdyU Gwin2eQHhmsWvMfowl2ei6bMyjoTWZmnFiriv0XsadBINvf6A0kt\n3l6enre7hSiKmqfRWf/NYK8m2ZLias2uf+kM3CYr Dsk1KJiVB07dnnUeR/e/qUL8eKI20ioy59Qt\n2fz+8ZDK5S9htOomSWvabV+3eHzYp3cFIR6dqNMKRcNgT3pQAA\n FaUfX+ptqDYcUsGsdKnFTxE/\nA1mTsA1jXfYFNGOU6g==\n", "url": "https://172.26.1.2/http_direct_file_io/req_id-UPLOAD-1002/import_file" }</pre>		
Upload contents of the CSV file	POST	<url_from_the_previous_output> Example: https://x.x.x/http_direct_file_io/req_id-UPLOAD-1002/import_file	file=@HostRecords.csv
	curl -k -u admin:Infoblox -H 'content-type:multipart-formdata' "https://127.0.0.1/ http_direct_file_io/req_id-UPLOAD-1002/import_file" -F file=@HostRecords.csv		
Upload the CSV file	POST	<wapi_url>/fileop?_function=csv_import	{ "action": "START", "on_error": "CONTINUE", "update_method": "MERGE", "token": "eJydU..==" }
	curl -k -u admin:Infoblox -H 'content-type:application/json' -X POST "https://127.0.0.1/wapi/v2.7/fileop?_function=csv_import" -d '{"action": "START", "on_error": "CONTINUE", "update_method": "MERGE", "token": "eJydUE1PwzAMvfuPbJe1Tbe2G7e hMQkJDbTB2WqTdfHqk5CkaPv3OENw4cYhkeP34TxLad0VvT4D\nX9KaEP0k/XgBMwlmd52g71k1uh04tXpALs2t njUPbgSJGI30RDJIIIiGcEtYa7cCk4zfXHkrxhp\n1DNwFexFVVRNVddV4l6tdyUGwin2eQHhmsWvMfowl2ei6bMyjoT		

	WZmnFiriv0XsadBINvf6A0kt\n3l6enre7hSiKMqfRWf/NYK8m2ZLias2uf+kM3CYrDsK1KJiVB07dnnUeR/e/qUL8eKI20ioy59Qt\n2fz+8ZDK5S9htOomSWvabV+3eHzYp3cFIR6dqNMKRcNgT3pQAaNFaUfX+ptqDYcUsGsdKnFTxE/t\nA1mTsA1jXfYFNGOU6g==\n"}'
--	---

Please note that while uploading the file (in the second step above), the Content-Type in the header will be *multipart/form-data*

Sample paging request

When a search result yields a large number of results, you can use paging. To start a paging request, the initial search request must have `_paging` and `_return_as_object` set to 1, and `_max_results` set to the desired page size. The returned results object will contain the `next_page_id` field and the result field set to the first page of results. Note that the `next_page_id` field only contains URL-safe characters so it can be used as is and no quotation characters are required for subsequent requests. To get the subsequent results, you can re-send GET requests to the original object and set `_page_id` to the ID string returned in the previous page of results.

The following example shows how to retrieve all host records in the environment, 4 at a time. You can change the queried network object or the number of records according to your requirement.

Operation	REST Method	API Call
		Sample cURL Command
		Sample Output
Fetch the first 4 host records	GET	<code><wapi_url>/record:host?_paging=1&_return_as_object=1&max_results=4</code>
		<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/record:host?_paging=1&_return_as_object=1&_max_results=4"</pre>
		<pre>{"next_page_id": "789c55904d4ec3301085f77391744354a7296db9413615a207b0fc1b8c128f653ba2bd3d330681d858f237efcd7bf6ce260 1b70eb3ab5b8ed207b7d82257b4a58334d0a48372eb569acba466c7f400d35e305579e6fb083b9b8e4da6ee3237cd334c2 313f6c8103d123b8131981ea498810e83b1d4bc998a19d2197686657ac17b8fd1f55a1507afe49e8842ba8091526f61a921 4a09a83f9ca990c41eae52d09c11504d7f5983f55b614277e7b96aa7265321261b084353470fca939634599b0841a304af4 beb836a7673cb5155655c5e0d482ce5034416a511fc931bf50343f29db97772cec1d5ad2ff7f652ce0ca9536fd87065e3a1c5 8ef43fe767355dad07f014b0f819a-e4718db1318cfc3099a781c541542ccd82f4738853cf5c1a2b0b371", "result": [{"_ref": "record:host/ZG5zLmhvc3QkLi9kZWZhdWx0LmNvbS5pbmZvLmhvc3Q:host.info.com/default", "ipv4addr": [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRkcmVzcyQuX2RlZmF1bHQyY29tLmluZm8uaG9zdC4xMC4xMC4xMC4yMC4: 10.10.10.20/host.info.com/default", "configure_for_dhcp": false, "host": "host.info.com", "ipv4addr": "10.10.10.20" }], "name": "host.info.com", "view": "default" }, {"_ref": "record:host/ZG5zLmhvc3QkLi9kZWZhdWx0LmNvbS5kZW1vLnNlcnZlcjE:server1.demo.com/default", "ipv4addr": [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRkcmVzcyQuX2RlZmF1bHQyY29tLmRlYW8uc2VydMvYMS4xLjEuMS4yNS4: 1.1.1.25/server1.demo.com/default", "configure_for_dhcp": false, "host": "server1.demo.com", "ipv4addr": "1.1.1.25" }]</pre>

	<pre> }}, "name": "server1.demo.com", "view": "default" }, { "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5kZW1vLm1vbml0b3I:monitor.demo.com/default", "ipv4addrs": [{ "_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRkcmVzcyQuX2RlZmF1bHQuY29tLmRibW8ubW9uaXRvci4xLjEuMS41MC4:1.1.1.50/monitor.demo.com/default", "configure_for_dhcp": false, "host": "monitor.demo.com", "ipv4addr": "1.1.1.50" },], "name": "monitor.demo.com", "view": "default" }, { "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLnRlc3Q:test.info.com/default", "ipv4addrs": [{ "_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRkcmVzcyQuX2RlZmF1bHQuY29tLmluZm8udGVzdC4yLjluMi4xMTAu2.2.2.110/test.info.com/default", "configure_for_dhcp": false, "host": "test.info.com", "ipv4addr": "2.2.2.110" },], "name": "test.info.com", "view": "default" }] } } } </pre>	
Query the next set of records	GET	 <wapi_url>/record:host?_paging=1&_return_as_object=1&_page_id=789c55904d4ec3301085f77391744354a7296db9413615a207b0fc1b8c128f653ba2bd3d330681d858f237efcd7bf6ce2601b70eb3ab5b8ed207b7d82257b4a58334d0a48372eb569acba466c7f400d35e305579e6fb083b9b8e4da6ee3237cd334c2313f6c8103d123b8131981ea498810e83b1d4bc998a19d2197686657ac17b8fd1f55a1507afe49e8842ba8091526f61a9214a09a83f9ca990c41eae52d09c11504d7f5983f55b614277e7b96aa7265321261b084353470fca939634599b0841a304af4beb836a7673cb5155655c5e0d482ce5034416a511fc931bf50343f29db97772cec1d5ad2ff7f652ce0ca9536fd87065e3a1c58ef43fe767355dad07f014b0f819a-e4718db1318cfc3099a781c541542ccd82f4738853cf5c1a2b0b371
	<pre> curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/record:host?_paging=1&_return_as_object=1&_page_id=789c55904d4ec3301085f77391744354a7296db9413615a207b0fc1b8c128f653ba2bd3d330681d858f237efcd7bf6ce2601b70eb3ab5b8ed207b7d82257b4a58334d0a48372eb569acba466c7f400d35e305579e6fb083b9b8e4da6ee3237cd334c2313f6c8103d123b8131981ea498810e83b1d4bc998a19d2197686657ac17b8fd1f55a1507afe49e8842ba8091526f61a9214a09a83f9ca990c41eae52d09c11504d7f5983f55b614277e7b96aa7265321261b084353470fca939634599b0841a304af4beb836a7673cb5155655c5e0d482ce5034416a511fc931bf50343f29db97772cec1d5ad2ff7f652ce0ca9536fd87065e3a1c58ef43fe767355dad07f014b0f819a-e4718db1318cfc3099a781c541542ccd82f4738853cf5c1a2b0b371" </pre>	
	<pre> { "next_page_id": "789c55904b6ec3300c44f7bc88b3322cd9f9f506de0445730041d6c755619b8224a3c9ed4b0a69812ef966c82179b051c0bd1e4ca9e36e5835b6c562bdadc4094a43490efcd4aba8a7a764c7b183bc154a799eb010e361eab4d3f54aa9e138c0393ff734938c38df1f4559ed1517da188e40c26fbf689b910b9721f67a9b07924203a3006e39346cfa0d8bce5927653309128e060d8382df8687173eda4b38377ea1f89924182516adac352c2a61450b43385780fb7f21145dd5e1c690d8fe95b27cb89a7df1373d1899712e7d7414b584305971798b1a08a984309b829f43ebbaa5f61ec59b7ba687e65c73992864e3c769ffe3e226555aad987547f20075aa083dcfe0072798173-a0384f2ecd4a419caa42aba69cb40f4827127cd8d59201fdd873f5e", "result": [{ "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLnMx:s1.info.com/default", "ipv4addrs": [{ "_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRkcmVzcyQuX2RlZmF1bHQuY29tLmRibW8ubW9uaXRvci4xLjEuMS41MC4:1.1.1.50/monitor.demo.com/default", "configure_for_dhcp": false, "host": "monitor.demo.com", "ipv4addr": "1.1.1.50" },], "name": "monitor.demo.com", "view": "default" }, { "_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLnRlc3Q:test.info.com/default", "ipv4addrs": [{ "_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRkcmVzcyQuX2RlZmF1bHQuY29tLmluZm8udGVzdC4yLjluMi4xMTAu2.2.2.110/test.info.com/default", "configure_for_dhcp": false, "host": "test.info.com", "ipv4addr": "2.2.2.110" },], "name": "test.info.com", "view": "default" }] }] } </pre>	

	<pre> [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWVRkcmVzcyQuX2RlZmF1bHQyY29tLmluZm8uczEuMTAuMTAuMTAuMjAu:10.1 0.10.20/s1.info.com/default", "configure_for_dhcp": false, "host": "s1.info.com", "ipv4addr": "10.10.10.20" }], "name": "s1.info.com", "view": "default" }, {"_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLmgy:h2.info.com/default", "ipv4addrs": [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWVRkcmVzcyQuX2RlZmF1bHQyY29tLmluZm8uaDIuNS41LjUuOTAu:5.5.5.90/h2.i nfo.com/default", "configure_for_dhcp": false, "host": "h2.info.com", "ipv4addr": "5.5.5.90" }], "name": "h2.info.com", "view": "default" }, {"_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLmUy:e2.info.com/default", "ipv4addrs": [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWVRkcmVzcyQuX2RlZmF1bHQyY29tLmluZm8uZTluOS45LjkuMTAwLg:9.9.9.100 /e2.info.com/default", "configure_for_dhcp": false, "host": "e2.info.com", "ipv4addr": "9.9.9.100" }], "name": "e2.info.com", "view": "default" }, {"_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLmRldnN0YWNR:devstack.info.com/default", "ipv4addrs": [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWVRkcmVzcyQuX2RlZmF1bHQyY29tLmluZm8uZGV2c3RhY2suMTcyLjI2LjEuMjA u:172.26.1.20/devstack.info.com/default", "configure_for_dhcp": false, "host": "devstack.info.com", "ipv4addr": "172.26.1.20" }], "name": "devstack.info.com", "view": "default" }]] </pre>	
Query the next set of records	GET	 <wapi_url>/record:host?_paging=1&_return_as_object=1&_page_id=789c55904b6ec3300c44f7bc88b3322cd9f9f506de0445730041d6c755619b8224a3c9ed4b0a69812ef966c82179b051c0bdc1e4ca9e36e5835b6c562bdadc4094a43490efcd4aba8a7a764c7b183bc154a799eb010e361eab4d3f54aa9e138c0393ff734938c38df1f4559ed1517da188e40c26fbf689b910b9721f67a9b07924203a3006e39346cfa0d8bce5927653309128e060d8382df8687173eda4b38377ea1f89924182516adac352c2a61450b43385780fb7f21145dd5e1c690d8fe95b27cb89a7df1373d1899712e7d7414b584305971798b1a08a984309b829f43ebbaa5f61ec59b7ba687e65c73992864e3c769ffe3e226555aad987547f20075aa083dcfe0072798173-a0384f2ecdda419caa42aba69cb40f4827127cd8d59201fdd873f5e

	<pre>curl -k -u admin:Infoblox -X GET "https://127.0.0.1/wapi/v2.7/record:host?_paging=1&_return_as_object=1&_page_id=789c55904b6ec3300c44f7bc88b33 22cd9f9f506de0445730041d6c755619b8224a3c9ed4b0a69812ef966c82179b051c0bdc1e4ca9e36e5835b6c562bdadc40 94a43490efcd4aba8a7a764c7b183bc154a799eb010e361eab4d3f54aa9e138c0393ff734938c38df1f4559ed1517da188e4 0c26fb689b910b9721f67a9b07924203a3006e39346cfa0d8bce5927653309128e060d8382df8687173eda4b38377ea1f89 924182516adac352c2a61450b43385780fb7f21145dd5e1c690d8fe95b27cb89a7df1373d1899712e7d7414b58430597179 8b1a08a984309b829f43ebbaa5f61ec59b7ba687e65c73992864e3c769ffe3e226555aad987547f20075aa083dcfe0072798 173-a0384f2ecdda419caa42aba69cb40f4827127cd8d59201fdd873f5e"</pre>
	<pre>{"result": [{"_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLnJvdXRlcmg:router.info.com/default", "ipv4addrs": [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWRkcmVzcyQuX2RlZmF1bHQuY29tLmluZm8ucm91dGVyLjEwMC4xMC4xMC4x Lg:100.10.10.1/router.info.com/default", "configure_for_dhcp": false, "host": "router.info.com", "ipv4addr": "100.10.10.1" }], "name": "router.info.com", "view": "default"]}]</pre>

Multiple Object Body Feature using */request*

The *request* object allows the control of WAPI through a single-entry point. You can either have Single Object Body Requests or Multiple Object Body Requests.

Single object body requests can be used for any supported WAPI operations, including scheduling and approval. Only HTTP methods (GET, PUT, DELETE and POST) are valid methods for the single object body

Multiple object body requests are composed of a list of single requests. While programming this enables to make multiple requests with one API call, thus simplifying an operation, and reducing the number of API calls to be made. In addition to the standard HTTP methods (GET, PUT, POST, and DELETE), single request inside a multiple object body request can be set to these values:

Method	Description
STATE:ASSIGN	Copy fields in the data object to the state object.
STATE:DISPLAY	Added the state object to the returned result list.

Note: The Multiple Object Body Feature is only available NIOS 7.1 onwards (WAPI v2.1 onwards)

Get records of multiple record types

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Get Host/A/PTR records	POST	<code><wapi_url>/request</code>	<pre>{ "method": "GET", "object": "record:host", "data": {}, "args": { "_return_fields": "name,dns_name,aliases,dns_aliases,ipv" } }</pre>

		<pre>4addrs,configure_for_dns"}}, {"method": "GET", "object": "record:a", "data": {}, "args": {}}, {"method": "GET", "object": "record:ptr", "data": {}, "args": {}}</pre>
		<pre>curl -k -u admin:Infoblox -H 'content-type:application/json' -X POST "https://127.0.0.1/wapi/v2.7/request" -d '[{"method": "GET", "object": "record:host", "data": {}, "args": {"_return_fields": "name,dns_name,aliases,dns_aliases,ipv4addr s,configure_for_dns"}}, {"method": "GET", "object": "record:a", "data": {}, "args": {}}, {"method": "GET", "object": "record:ptr", "data": {}, "args": {}}]'</pre>
		<pre>[[{"_ref": "record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS50ZXN0Lmhvc3Qx: host1.test.com/default", "configure_for_dns": true, "dns_name": "host1.test.com", "ipv4addrs": [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWw3cmVzcycyQuX2RlZmF1bHQuY29tLnRlc3QuaG9zdDEuMTcyLjI2LjEuMjEu: 1 72.26.1.21/host1.test.com/default", "configure_for_dhcp": false, "host": "host1.test.com", "ipv4addr": "172.26.1.21" }], {"_ref": "record:host/ZG5zLmhvc3QkLjEuY29tLmluZm8ubWFpbGVy: mailer.info.com/default.test", "configure_for_dns": true, "dns_name": "mailer.info.com", "ipv4addrs": [{"_ref": "record:host_ipv4addr/ZG5zLmhvc3RfYWw3cmVzcycyQuMS5jb20uaW5mb3Y5Y29tLmluMTAuMS4xLjEwMC4: 10.1.1.100 /mailer.info.com/default.test", "configure_for_dhcp": false, "host": "mailer.info.com", "ipv4addr": "10.1.1.100", "mac": "aa:bb:cc:11:22:33" }], "name": "mailer.info.com" }], [{"_ref": "record:a/ZG5zLmJpbmRfYSQuX2RlZmF1bHQuY29tLmluZm8sdGVzdC5ob3N0LDE3Mi4yNi4xLjEwMA: host.test.info.c om/default", "ipv4addr": "172.26.1.100", "name": "host.test.info.com", "view": "default" }], {"_ref": "record:a/ZG5zLmJpbmRfYSQuX2RlZmF1bHQuY29tLmluZm8sc2VydmlvLDEwLjEwLjEwLjE: server.info.com/default", "ipv4addr": "10.10.10.2", "name": "server.info.com", "view": "default" }], [{"_ref": "record:ptr/ZG5zLmJpbmRfcHRyJC5fZGVmYXVsdC5hcnBhLmluLWFkZHIuMTI3LjAuMC4xLmxvY2FsaG9zdA: 1.0.0.12 7.in-addr.arpa/default", "ptrdname": "localhost", "view": "default" }]</pre>

			"args": {"_function": "next_available_ip"}, "enable_substitution": true}]
		curl -k -u admin:Infoblox -H 'content-type:application/json' -X POST "https://127.0.0.1/wapi/v2.7/request" -d '[{"method": "GET", "object": "network", "data": {"network": "192.168.1.0/24"}, "assign_state": {"netw_ref": "_ref"}, "discard": true}, {"method": "POST", "object": "##STATE:netw_ref:##", "args": {"_function": "next_available_ip"}, "enable_substitution": true}]'	
		[{"ips": ["192.168.1.1"]}]	

Pre-provision a Grid member

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Pre-provision a grid member	POST	<wapi_url>/request	{ "method": "POST", "object": "member", "data": { "host_name": "autojoin01.localdomain", "vip_setting": { "address": "172.16.0.160", "gateway": "172.16.0.1", "subnet_mask": "255.255.255.0" }, "config_addr_type": "IPv4", "platform": "VNIOS", "assign_state": { "member_ref": "_ref" }, "enable_substitution": true, "discard": true, "method": "PUT", "object": "##STATE:member_ref:##", "enable_substitution": true, "data": { "pre_provisioning": { "hardware_info": { "hwmodel": "IB-VM-820", "hwtype": "IB-VNIOS" }, "licenses": ["dhcp", "dns", "enterprise", "vnios", "rpz"] } } } }, {"method": "STATE:DISPLAY"}]
	curl -k -u admin:Infoblox -H 'content-type:application/json' -X POST "https://127.0.0.1/wapi/v2.7/request" -d '[{"method": "POST", "object": "member", "data": {"host_name": "autojoin01.localdomain", "vip_setting": {"address": "172.16.0.160", "gateway": "172.16.0.1", "subnet_mask": "255.255.255.0"}, "config_addr_type": "IPv4", "platform": "VNIOS"}, "assign_state": {"member_ref": "_ref"}, "enable_substitution": true, "discard": true}, {"method": "PUT", "object": "##STATE:member_ref:##", "enable_substitution": true, "data": {"pre_provisioning": {"hardware_info": {"hwmodel": "IB-VM-820", "hwtype": "IB-VNIOS"}, "licenses": ["dhcp", "dns", "enterprise", "vnios", "rpz"]}}}, {"method": "STATE:DISPLAY"}]'		
	["member/b25lLnZpcnR1YWxfbm9kZSQy:autojoin01.infoblox", { "member_ref": "member/b25lLnZpcnR1YWxfbm9kZSQy:autojoin01.infoblox" }]		

Enable DHCP service on a member

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Enable the DHCP service on a member	POST	<wapi_url>/request	<pre>[{"method": "STATE:ASSIGN", "data":{"member_name":"member1.localdomain"}, {"method":"GET", "object": "member:dhcpproperties", "data": {"host_name": "##STATE:member_name:##", "assign_state": {"host_ref": " _ref"}, "enable_substitution": true, "discard": true}, {"method": "PUT", "object": "##STATE:host_ref:##", "enable_substitution": true, "data": {"enable_dhcp":true}}, {"method": "STATE:DISPLAY"}]</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type:application/json' -X POST "https://127.0.0.1/wapi/v2.7/request" - d '[{"method": "STATE:ASSIGN","data":{"member_name":"member1.localdomain"}},{"method":"GET","object": "member:dhcpproperties","data": {"host_name": "##STATE:member_name:##"},"assign_state": {"host_ref": " _ref"},"enable_substitution": true,"discard": true},{ "method": "PUT", "object": "##STATE:host_ref:##","enable_substitution": true,"data": {"enable_dhcp":true}},{"method": "STATE:DISPLAY"}]'</pre>		
	<pre>[{"member:dhcpproperties/ZG5zLm1lbWJlcl9kaGNwX3Byb3BlcnRpZXMKMQ:member1.localdomain", {"host_ref": "member:dhcpproperties/ZG5zLm1lbWJlcl9kaGNwX3Byb3BlcnRpZXMKMQ:member1.localdomain", "member_name": "member1.localdomain"}]</pre>		

Get permissions for an object

Operation	REST Method	API Call	Sample Body
	Sample cURL Command		
	Sample Output		
Get permissions for an object	POST	<wapi_url>/request	<pre>[{"method": "STATE:ASSIGN", "data": {"zone_name": "test.org"}, {"method": "GET", "object": "zone_auth", "data": {"fqdn": "##STATE:zone_name:##"}, "assign_state": {"zone_ref": " _ref"}, "enable_substitution": true,"discard": true}, {"method": "GET", "object": "permission", "enable_substitution": true }, {"method": "STATE:DISPLAY"}]</pre>
	<pre>curl -k -u admin:Infoblox -H 'content-type:application/json' -X POST "https://127.0.0.1/wapi/v2.7/request" -d '[{"method": "STATE:ASSIGN","data": {"zone_name": "test.org"}},{"method": "GET","object": "zone_auth","data": {"fqdn": "##STATE:zone_name:##"},"assign_state": {"zone_ref": " _ref"},"enable_substitution": true,"discard": true},{ "method": "GET", "object": "permission","enable_substitution": true },{"method": "STATE:DISPLAY"}]'</pre>		

	<pre> [{"_ref": "permission/b25lMhpZXJfcVVsZSQuY29tLmluZm9ibG94LmRucy56b25lJC4uLi4uY29tLmluZm9ibG94Lm9uZS5yb2xl JEROUyBBZG1pbi4:DNS%20Admin/WRITE", "permission": "WRITE", "resource_type": "VIEW", "role": "DNS Admin" }, ... {"_ref": "permission/b25lMhpZXJfcVVsZSQuY29tLmluZm9ibG94LmRucy5jbHVzdGVyX2Ruc19wcm9wZXJ0aWVzJDAuLi5jb 20uaW5mb2Jsb3gub25lMfkbWluX2dyb3VwJC5jbG91ZC1hcGktb25seS4:cloud-api-only/WRITE", "group": "cloud-api-only", "permission": "WRITE", "resource_type": "GRID_DNS_PROPERTIES" }], {"zone_name": "test.org", "zone_ref": "zone_auth/ZG5zLnprvbmUkLjEub3JnLnRlc3Q:test.org/default.test" }] </pre>
--	---

Sample Codes

This section covers sample API calls in different programming/scripting languages

Python

These examples use the *requests* module. For more details on the requests module, please refer <http://docs.python-requests.org>

Network Object	REST Operation	Sample Python Programme
HOST record	GET	<pre> import requests requests.packages.urllib3.disable_warnings() # Disable SSL warnings in requests # url = "https://127.0.0.1/wapi/v2.7/record:host?_return_as_object=1" response = requests.request("GET", url, auth=('admin', 'Infoblox'), verify=False) print(response.text) </pre>
	GET	<pre> #This example demonstrates the use of cookies(ibapauth)# import requests import copy requests.packages.urllib3.disable_warnings() # Disable SSL warnings in requests # url = "https://127.0.0.1/wapi/v2.7/record:host?_return_as_object=1" response = requests.request("GET", url, auth=('admin', 'Infoblox'), verify=False) authcookie = copy.copy(response.cookies['ibapauth']) print(authcookie) print(response.text) #You can re-use the authcookie in subsequent API calls# response_using_cookie = requests.request("GET", url, cookies=authcookie, verify=False) print(response_using_cookie.text) </pre>
	GET	<pre> #This example demonstrates how you can work with large results using paging# import requests requests.packages.urllib3.disable_warnings() # Disable SSL warnings in requests # url = "https://127.0.0.1/wapi/v2.7/record:host?_return_as_object=1" querystring = {'_max_results':"20","_paging":"1","_return_as_object":"1"} response = requests.request("GET", url, auth=('admin', 'Infoblox'), params=querystring,verify=False) print(response.text) next_page_id = response.json()[0]['next_page_id'] </pre>

		<pre> while next_page_id: query = {"_page_id":next_page_id} newresponse = requests.request("GET", url, auth=('admin', 'infoblox'), params=query,verify=False) print(newresponse.text) try: next_page_id = newresponse.json()['next_page_id'] except Exception: next_page_id = 0 </pre>
	POST	<pre> import requests requests.packages.urllib3.disable_warnings() # Disable SSL warnings in requests # url = "https://127.0.0.1/wapi/v2.7/record:host?_return_as_object=1" payload = '{"name":"host.info.com","ipv4addrs":[{"ipv4addr":"172.26.1.20"}]}' headers = {'content-type': "application/json"} response = requests.request("POST", url, auth=('admin', 'Infoblox'), data=payload, headers=headers, verify=False) print(response.text) </pre>
	POST (using the 'session' method that automatically handles the session cookies, etc when you perform multiple POST/GET)	<pre> import requests import getpass requests.packages.urllib3.disable_warnings() # Disable SSL warnings in requests # url = "https://127.0.0.1/wapi/v2.7" # Get username and password for auth # user = input('Username: ') passwd = getpass.getpass('Password: ') wapi_session = requests.session() wapi_session.auth = (user, 'passwd') payload = '{"name":"host.info.com","ipv4addrs":[{"ipv4addr":"172.26.1.20"}]}' headers = { 'content-type': "application/json" } # WAPI call POST /record:host # response = wapi_session.post(wapi_url+"/record:host?_return_as_object=1", data=payload, headers=headers, auth=(user, passwd), verify=False) # Error handling # if response.status_code == 201: print("Success.") else: print("Failed.") print(response.text) </pre>
	PUT	<pre> import requests requests.packages.urllib3.disable_warnings() # Disable SSL warnings in requests # url = "https://127.0.0.1/wapi/v2.7/record:host/ZG5zLmhvc3QkLl9kZWZhdWx0LmNvbS5pbmZvLmhvc3Qx:host.inf o.com/default?_return_as_object=1" payload = '{"name":"host1.info.com"}' headers = {'content-type': "application/json"} response = requests.request("PUT", url, auth=('admin', 'Infoblox'), data=payload, headers=headers, verify=False) print(response.text) </pre>
	DELETE	<pre> import requests requests.packages.urllib3.disable_warnings() # Disable SSL warnings in requests # url = "https://127.0.0.1/wapi/v2.7/record:host/ZG5zLmhvc3QkLl9kZWZhdWx0LmNvbS5pbmZvLmhvc3Qx:host.inf o.com/default?_return_as_object=1" headers = {'content-type': "application/json"} </pre>

		<pre>response = requests.request("DELETE", url, auth=('admin', 'Infoblox'), headers=headers, verify=False) print(response.text)</pre>
--	--	---

Powershell

The following examples use the following function to avoid SSL Certificate issue while using HTTPS.

Function Description	Function Contents
Function to ignore Self-Signed Certificates	<pre>function Ignore-SelfSignedCerts { try { Write-Host "Adding TrustAllCertsPolicy type." -ForegroundColor White Add-Type -TypeDefinition @" using System.Net; using System.Security.Cryptography.X509Certificates; public class TrustAllCertsPolicy : ICertificatePolicy { public bool CheckValidationResult(ServicePoint srvPoint, X509Certificate certificate, WebRequest request, int certificateProblem) { return true; } } "@ Write-Host "TrustAllCertsPolicy type added." -ForegroundColor White } catch { Write-Host \$_ -ForegroundColor "Yellow" } [System.Net.ServicePointManager]::CertificatePolicy = New-Object TrustAllCertsPolicy }</pre>

Network Object	REST Operation	Sample Powershell Code
HOST record	GET	<pre>Ignore-SelfSignedCerts \$url = "https://127.0.0.1/wapi/v2.7/record:host?_return_as_object=1" \$pwd = ConvertTo-SecureString "Infoblox" -AsPlainText -Force \$creds = New-Object Management.Automation.PSCredential ('admin', \$pwd) Invoke-RestMethod -Uri \$url -Method GET -Credential \$creds</pre>
	GET	<pre>#This example demonstrates the use of cookies(ibapauth)# Ignore-SelfSignedCerts \$url = "https://127.0.0.1/wapi/v2.7/record:host?_return_as_object=1" \$pwd = ConvertTo-SecureString "Infoblox" -AsPlainText -Force \$creds = New-Object Management.Automation.PSCredential ('admin', \$pwd) Invoke-RestMethod -Uri \$url -Method GET -Credential \$creds -SessionVariable authcookie #You can re-use the authcookie in subsequent API calls# Invoke-RestMethod -Uri \$url -Method GET -WebSession \$authcookie</pre>

	POST	<pre>Ignore-SelfSignedCerts \$url = "https://127.0.0.1/wapi/v2.7/record:host?_return_as_object=1" \$pwd = ConvertTo-SecureString "Infoblox" -AsPlainText -Force \$creds = New-Object Management.Automation.PSCredential ('admin', \$pwd) \$host_details = @{ name="host.info.com" ipv4addrs=@(@{ipv4addr="172.26.1.20"})} \$body = \$host_details ConvertTo-Json Invoke-RestMethod -Uri \$url -Method POST -Credential \$creds -ContentType 'application/json' -Body \$body</pre>
	PUT	<pre>Ignore-SelfSignedCerts \$url = "https://127.0.0.1/wapi/v2.7/record:host/ZG5zLmhvc3QkLi9kZWZhdWx0LmNvbS5pbmZvLmhvc3Qx:host.info.com/default?_return_as_object=1" \$pwd = ConvertTo-SecureString "Infoblox" -AsPlainText -Force \$creds = New-Object Management.Automation.PSCredential ('admin', \$pwd) \$host_details = @{ name="host1.info.com"} \$body = \$host_details ConvertTo-Json Invoke-RestMethod -Uri \$url -Method PUT -Credential \$creds -ContentType 'application/json' -Body \$body</pre>
	DELETE	<pre>Ignore-SelfSignedCerts \$url = "https://127.0.0.1/wapi/v2.7/record:host/ZG5zLmhvc3QkLi9kZWZhdWx0LmNvbS5pbmZvLmhvc3Qx:host.info.com/default?_return_as_object=1" \$pwd = ConvertTo-SecureString "Infoblox" -AsPlainText -Force \$creds = New-Object Management.Automation.PSCredential ('admin', \$pwd) Invoke-RestMethod -Uri \$url -Method DELETE -Credential \$creds</pre>

Java

Using Java Unirest

Network Object	REST Operation	Sample Java Unirest Code
HOST record	GET	<pre>HttpResponse<JsonNode> response = Unirest.get("https://127.0.0.1/wapi/v2.7/record:host?_return_as_object=1") .header("authorization", "Basic YWRtaW46aW5mb2Jsbn3g=") .header("accept", "application/json") .asJson();</pre>
	POST	<pre>HttpResponse<JsonNode> response = Unirest.post("https://127.0.0.1/wapi/v2.7/record:host?_return_as_object=1") .header("authorization", "Basic YWRtaW46aW5mb2Jsbn3g=") .header("accept", "application/json") .header("content-type", "application/json") .body("{\"name\":\"host.info.com\",\"ipv4addrs\": [{\"ipv4addr\":\"172.26.1.20\"}]}") .asJson();</pre>
	PUT	<pre>HttpResponse<JsonNode> response = Unirest.put("https://127.0.0.1/wapi/v2.7/record:host/ZG5zLmhvc3QkLi9kZWZhdWx0LmNvbS5pbmZvLmhvc3Qx:host.info.com/default?_return_as_object=1") .header("authorization", "Basic YWRtaW46aW5mb2Jsbn3g=") .header("accept", "application/json") .header("content-type", "application/json")</pre>

		<pre>.body("{\"name\":\"host1.info.com\"}") .json();</pre>
	DELETE	<pre>HttpResponse<JsonNode> response = Unirest.delete("https://127.0.0.1/wapi/v2.7/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLmhvc3Qx:host.info.com/default?_return_as_object=1") .header("authorization", "Basic YWRtaW46aW5mb2Jsbn3g=") .header("accept", "application/json") .header("content-type", "application/json") .body("{\"name\":\"host1.info.com\"}") .json();</pre>

Using Java OkHttpClient

Network Object	REST Operation	Sample Java OkHttpClient Code
HOST record	GET	<pre>OkHttpClient client = new OkHttpClient(); Request request = new Request.Builder() .url("https://127.0.0.1/wapi/v2.7/record:host?_return_as_object=1") .get() .addHeader("authorization", "Basic YWRtaW46aW5mb2Jsbn3g=") .build(); Response response = client.newCall(request).execute();</pre>
	POST	<pre>OkHttpClient client = new OkHttpClient(); MediaType mediaType = MediaType.parse("application/json"); RequestBody body = RequestBody.create(mediaType, "{\"name\":\"host.info.com\",\"ipv4addr\":{\"ipv4addr\":\"172.26.1.20\"}}"); Request request = new Request.Builder() .url("https://127.0.0.1/wapi/v2.7/record:host?_return_as_object=1") .post(body) .addHeader("authorization", "Basic YWRtaW46aW5mb2Jsbn3g=") .addHeader("content-type", "application/json") .build(); Response response = client.newCall(request).execute();</pre>
	PUT	<pre>OkHttpClient client = new OkHttpClient(); MediaType mediaType = MediaType.parse("application/json"); RequestBody body = RequestBody.create(mediaType, "{\"name\":\"host1.info.com\"}"); Request request = new Request.Builder() .url("https://127.0.0.1/wapi/v2.7/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLmhvc3Qx:host.info.com/default?_return_as_object=1") .put(body) .addHeader("authorization", "Basic YWRtaW46aW5mb2Jsbn3g=") .addHeader("content-type", "application/json") .build(); Response response = client.newCall(request).execute();</pre>
	DELETE	<pre>OkHttpClient client = new OkHttpClient(); Request request = new Request.Builder() .url("https://127.0.0.1/wapi/v2.7/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLmhvc3Qx:host.info.com/default?_return_as_object=1") .delete() .addHeader("authorization", "Basic YWRtaW46aW5mb2Jsbn3g=") .addHeader("content-type", "application/json") .build(); Response response = client.newCall(request).execute();</pre>

		<pre>.build(); Response response = client.newCall(request).execute();</pre>
--	--	---

C# (CSharp)

These examples use the *RestSharp* library.

Network Object	REST Operation	Sample C# Scripts
HOST record	GET	<pre>ServicePointManager.ServerCertificateValidationCallback +=(sender, certificate, chain, sslPolicyErrors) =>true; //Disable SSL Warnings// var client = new RestClient("127.0.0.1"); var request = new RestRequest("wapi/v2.7/record:host?_return_as_object=1",Method.GET); request.AddHeader("authorization", "Basic YWRtaW46aW5mb2Jsb3g="); IRestResponse response = client.Execute(request); var results = response.Data;</pre>
	POST	<pre>ServicePointManager.ServerCertificateValidationCallback +=(sender, certificate, chain, sslPolicyErrors) =>true; //Disable SSL Warnings// var client = new RestClient("127.0.0.1"); var request = new RestRequest("wapi/v2.7/record:host?_return_as_object=1",Method.POST); request.AddHeader("authorization", "Basic YWRtaW46aW5mb2Jsb3g="); request.AddHeader("content-type", "application/json"); request.AddParameter("application/json", "{\"name\":\"host.info.com\",\"ipv4addr\": [{\"ipv4addr\":\"172.26.1.20\"}]}", ParameterType.RequestBody); IRestResponse response = client.Execute(request); var results = response.Data;</pre>
	PUT	<pre>ServicePointManager.ServerCertificateValidationCallback +=(sender, certificate, chain, sslPolicyErrors) =>true; //Disable SSL Warnings// var client = new RestClient("127.0.0.1"); var request = new RestRequest("wapi/v2.7/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLmhvc3Qx:host.info.co m/default?_return_as_object=1",Method.PUT); request.AddHeader("authorization", "Basic YWRtaW46aW5mb2Jsb3g="); request.AddHeader("content-type", "application/json"); request.AddParameter("application/json", "{\"name\":\"host1.info.com\"}", ParameterType.RequestBody); IRestResponse response = client.Execute(request); var results = response.Data;</pre>
	DELETE	<pre>ServicePointManager.ServerCertificateValidationCallback +=(sender, certificate, chain, sslPolicyErrors) =>true; //Disable SSL Warnings// var client = new RestClient("127.0.0.1"); var request = new RestRequest("wapi/v2.7/record:host/ZG5zLmhvc3QkLI9kZWZhdWx0LmNvbS5pbmZvLmhvc3Qx:host.info.co m/default?_return_as_object=1",Method.DELETE); request.AddHeader("authorization", "Basic YWRtaW46aW5mb2Jsb3g="); request.AddHeader("content-type", "application/json"); IRestResponse response = client.Execute(request); var results = response.Data;</pre>

References

[Infoblox Community: Definitive List of REST Examples](#)

[Youtube Channel: WAPI Bytes](#)