

18.0 Host Identity APIs

WAS THIS PAGE HELPFUL? [Leave Feedback](#)

18.0 Host Identities APIs

18.1 Query Host Identities

Retrieve host identity records by query. Host identities provide a mapping between a UUID key and a set of alternate host names and a host. Queries would typically be used to drive administration browsers and similar tooling. Queried host identities, (one or more), are always returned wrapped in an XML <hostIdentities> element or JSON object with an `hostIdentities` array member. Getting a single host identity record is not wrapped.

18.1.1 Method: GET Host Identities

[GET /api/hostidentities?query=\(query criteria see below\)](#)

18.1.2 Method: GET a Single Host Identity by Id or Host Name

[GET /api/hostidentities/{idOrHostName}](#)

18.1.3 HTTP Query and Path Parameters

Field	Type	Description	Required
query	Query	An encoded query string (where clause)	no
first	Query	Paging, first record to start from	no
count	Query	Paging, the number of records to include when paging	no
idOrHostName	Path	the id or host name to retrieve, (case insensitive)	yes

If neither the `idOrHostName` path parameter nor query query parameter are provided, all host identity records will be retrieved. The first and count paging parameters can be used to avoid excessive retrieval.

18.1.4 HTTP Headers

Header	Valid Values	Required
Accept	application/xml or application/json	False
GWOS-API-TOKEN	a valid token returned from login	True
GWOS-APP-NAME	your application name	True

18.1.5 Query Fields

The table below contains valid fields in the value of the 'query' query parameter.

Field	Description	Alias
hostIdentityId	primary UUID key	id
hostName	primary/host host name	
hostNames	alternate host names, (also contains hostName)	
host	host the host identity is associated with	

Note: query fields are case-insensitive, thus camelCase, or all lower case will both work fine.

18.1.6 Example Queries

These examples are not HTTP encoded for readability. In practice queries must be encoded.

1. query for first page of all host identities
GET /api/hostidentities?count=25
2. query for SERVER_2's host identity
GET /api/hostidentities/SERVER_2
3. query for the second page of host identities that start with a prefix, (ordered by hostName)
GET /api/hostidentities?query=hostNames.id ilike 'server_%' ORDER BY hostName&first=25&count=25

18.1.7 HTTP Status Codes

Code	Description
200	Query returned no host identities
200	Query returned one or more host identities
401	Authentication/authorization error occurred
500	An internal server error occurred while querying host identities

18.1.8 Example Query Results

Here is an XML example of the result of a query finding host identities.

XML query results are always wrapped in an <hostIdentities> collection element, with one or more <hostIdentity> subelements.

```
<hostIdentities>
  <hostIdentity hostIdentityId="11c2d250-6437-11e4-9584-002219ed08d3" hostName="test-host-name-0"
host="false"><hostNames><hostName>test-host-name-0</hostName></hostNames></hostIdentity>
  <hostIdentity hostIdentityId="12042cdc-6437-11e4-9585-002219ed08d3" hostName="test-host-name-1"
host="false"
><hostNames><hostName>test-host-name-1</hostName><hostName>test-host-name-1.0</hostName><hostName>test-hc
```

Here is a JSON example of the result of a query finding host identities.

JSON query results are always wrapped in an object with a hostIdentities array member, with one or more object members.

```
{
  "hostIdentities" : [ {
    "hostIdentityId" : "1267adf2-6437-11e4-82be-002219ed08d3",
    "hostName" : "test-host-name-0",
    "hostNames" : [ "test-host-name-0" ],
    "host" : false
  }, {
    "hostIdentityId" : "12c545e8-6437-11e4-82bf-002219ed08d3",
    "hostName" : "test-host-name-1",
    "hostNames" : [ "test-host-name-1", "test-host-name-1.0", "test-host-name-1.1" ],
    "host" : false
  } ]
}
```

18.1.9 Example Get Single Results

Here is an XML example of the get host identity result.

```
<hostIdentity hostIdentityId="11c2d250-6437-11e4-9584-002219ed08d3" hostName="test-host-name-0" host="
false"><hostNames><hostName>test-host-name-0</hostName></hostNames></hostIdentity>
```

Here is a JSON example of the get host identity result.

```

{
  "hostIdentityId" : "1267adf2-6437-11e4-82be-002219ed08d3",
  "hostName" : "test-host-name-0",
  "hostNames" : [ "test-host-name-0" ],
  "host" : false
}

```

18.2 Create or Update Host Identities

Add or update host identities by post. Any number of host identities can be posted to the server in a single asynchronous, (the default), or synchronous request. Host identity ids will be defaulted by the server, but they can be specified as part of the post body request. Requests are considered updates if the host identity referenced by id or primary host name exists. Updates that specify a hostName are assumed to rename the host identity and associated host. If the update specifies hostNames, these are added as new alternate host names. The delete/clear request documented below can be used to remove all hostNames.

18.2.1 Method: POST Host Identities

[POST /api/hostidentities](#)

18.2.2 HTTP Query Parameters

Field	Description	Required
async	A boolean flag to indicate whether to submit the batch of host identity inserts and/or updates asynchronously or not. Default is false. Valid values are true or false. A synchronous request will not return back to your client code until the completion of processing all objects provided in the post data. Whereas an asynchronous request will submit the work to a queue, and return immediately.	False

18.2.3 HTTP Headers

Header	Valid Values	Required
Content-Type	application/xml or application/json	True
Accept	application/xml or application/json	True
GWOS-API-TOKEN	a valid token returned from login	True
GWOS-APP-NAME	your application name	True

18.2.4 POST Data Example

Here is an XML example post data creating two host identities on the server, (the wrapping <hostidentities> collection element is always required, even for a single host identity element):

```

<hostIdentities>
  <hostIdentity hostName="test-host-name-1"
  ><hostNames><hostName>test-host-name-1.0</hostName><hostName>test-host-name-1.1</hostName><hostName>test-
  <hostIdentity hostIdentityId="5d33537c-293e-4564-b0cf-d0998389b31f" hostName="test-host-name-2"
  ><hostNames><hostName>46eb4a5b-a08e-470b-9be7-8108aa5a2cc7</hostName><hostName>test-host-name-2</hostName

```

Here is an XML example of post data to update a host identity:

```

<hostIdentities>
  <hostIdentity hostIdentityId="12042cdc-6437-11e4-9585-002219ed08d3" hostName=
  "test-host-name-1-changed"
  ><hostNames><hostName>test-host-name-1-changed-alias</hostName></hostNames></hostIdentity>
</hostIdentities>

```

Here is a JSON example post data creating a host identity, (the wrapping object with the hostIdentities array member is always required,

even for a single host identity object):

```
{
  "hostIdentities": [{
    "hostName": "test-host-name-0",
    "hostNames": ["test-host-name-0"]
  }]
}
```

Here is a JSON example of post data to update a host identity:

```
{
  "hostIdentities": [{
    "hostIdentityId": "12c545e8-6437-11e4-82bf-002219ed08d3",
    "hostName": "test-host-name-1-changed",
    "hostNames": ["test-host-name-1-changed-alias"]
  }]
}
```

18.2.5 HTTP Status Codes

Code	Description
200	Zero or more host identities were created synchronously or an asynchronous create started
401	Authentication/authorization error occurred
500	An internal server error occurred while creating host identities or starting an asynchronous create

18.2.6 Example Responses

Here are XML asynchronous and synchronous create responses:

```
<results successful="1" failed="0" entityType="HostIdentity Async" operation="Insert" warning="0"
count="1">
  <result><entity>1415114840770</entity><message>Job 1415114840770
submitted</message><status>success</status></result>
</results>
```

```
<results successful="2" failed="0" entityType="HostIdentity" operation="Insert" warning="0" count="2">
  <result><entity>test-host-name-1</entity><message>http:
//localhost:8080/foundation-webapp/api/hostidentities/test-host-name-1</message><status>success</status><
<result><entity>test-host-name-2</entity><message>http:
//localhost:8080/foundation-webapp/api/hostidentities/test-host-name-2</message><status>success</status><
</results>
```

```
<results successful="0" failed="1" entityType="HostIdentity" operation="Insert" warning="0" count="1">
  <result><entity>test-host-name-3</entity><message>Failed to create HostIdentity: Could not execute
JDBC batch update; nested exception is org.hibernate.exception.ConstraintViolationException: Could not
execute JDBC batch update</message><status>failure</status></result>
</results>
```

Here is an XML synchronous update response:

```
<results successful="1" failed="0" entityType="HostIdentity" operation="Update" warning="0" count="1">
  <result><entity>12042cdc-6437-11e4-9585-002219ed08d3</entity><message>http:
//localhost:8080/foundation-webapp/api/hostidentities/test-host-name-1-changed</message><status>success</
</results>
```

Here are JSON asynchronous and synchronous create responses:

```
{
  "successful" : 1,
  "failed" : 0,
  "entityType" : "HostIdentity Async",
  "operation" : "Insert",
  "warning" : 0,
  "results" : [ {
    "entity" : "1415114842259",
    "status" : "success",
    "message" : "Job 1415114842259 submitted"
  } ],
  "count" : 1
}
```

```
{
  "successful" : 2,
  "failed" : 0,
  "entityType" : "HostIdentity",
  "operation" : "Insert",
  "warning" : 0,
  "results" : [ {
    "entity" : "test-host-name-1",
    "status" : "success",
    "message" : "http://localhost:8080/foundation-webapp/api/hostidentities/test-host-name-1"
  }, {
    "entity" : "test-host-name-2",
    "status" : "success",
    "message" : "http://localhost:8080/foundation-webapp/api/hostidentities/test-host-name-2"
  } ],
  "count" : 2
}
```

```
{
  "successful" : 0,
  "failed" : 1,
  "entityType" : "HostIdentity",
  "operation" : "Insert",
  "warning" : 0,
  "results" : [ {
    "entity" : "test-host-name-3",
    "status" : "failure",
    "message" : "Failed to create HostIdentity: Could not execute JDBC batch update; nested exception
is org.hibernate.exception.ConstraintViolationException: Could not execute JDBC batch update"
  } ],
  "count" : 1
}
```

Here is a JSON synchronous update response:

```

{
  "successful" : 1,
  "failed" : 0,
  "entityType" : "HostIdentity",
  "operation" : "Update",
  "warning" : 0,
  "results" : [ {
    "entity" : "12c545e8-6437-11e4-82bf-002219ed08d3",
    "status" : "success",
    "message" : "http://localhost:8080/foundation-webapp/api/hostidentities/test-host-name-1-changed"
  } ],
  "count" : 1
}

```

18.3 Delete or Clear Host Identities

Delete host identities or clear host identities host names by delete. Any number of host identities can be deleted to the server in a single request. Host identity ids or primary host names must be specified as part of the delete url or body request.

18.3.1 Method: Delete Host Identities

DELETE /api/hostidentities

18.3.2 Method: Delete Host Identities by Id or Host Names

DELETE /api/hostidentities/{idOrHostNames}

18.3.3 HTTP Query and Path Parameters

Field	Type	Description	Required
clear	Query	A boolean flag to indicate whether the host identities host names should be cleared or if the host identities should be deleted. Default is false, (delete host identities). Valid values are true or false.	False
idOrHostNames	Path	list of ids or host names to retrieve, (case insensitive, comma separated list)	True

18.3.4 HTTP Headers

Header	Valid Values	Required
Content-Type	application/xml or application/json	True
Accept	application/xml or application/json	True
GWOS-API-TOKEN	a valid token returned from login	True
GWOS-APP-NAME	your application name	True

18.3.5 POST Data Example

Here is an XML example post data deleting or clearing two host identities on the server, (the wrapping <hostidentities> collection element is always required, even for a single host identity element):

```

<hostIdentities>
  <hostIdentity hostIdentityId="12042cdc-6437-11e4-9585-002219ed08d3" hostName=
"test-host-name-1-changed"
><hostNames><hostName>test-host-name-1</hostName><hostName>test-host-name-1-changed</hostName><hostName>t
<hostIdentity hostIdentityId="5d33537c-293e-4564-b0cf-d0998389b31f" hostName="test-host-name-2"
><hostNames><hostName>test-host-name-2</hostName><hostName>46eb4a5b-a08e-470b-9be7-8108aa5a2cc7</hostName

```

Here is a JSON example post data deleting or clearing two host identities, (the wrapping object with the hostIdentities array member is always required, even for a single host identity object):

```

{
  "hostIdentities" : [ {
    "hostIdentityId" : "12c545e8-6437-11e4-82bf-002219ed08d3",
    "hostName" : "test-host-name-1-changed",
    "hostNames" : [ "test-host-name-1", "test-host-name-1-changed", "test-host-name-1-changed-alias" ]
  }, {
    "hostIdentityId" : "8fb9a880-f093-4892-a0f3-b1a403b49e5e",
    "hostName" : "test-host-name-2",
    "hostNames" : [ "test-host-name-2", "9e1b25a0-e036-4ce1-bf01-56c9cc8934ce" ]
  } ]
}

```

18.3.6 HTTP Status Codes

Code	Description
200	Zero or more host identities were deleted or cleared successfully
401	Authentication/authorization error occurred
500	An internal server error occurred while creating host identities or starting an asynchronous create

18.3.7 Example Responses

Here are XML clear and delete responses:

```

<results successful="1" failed="0" entityType="HostIdentity" operation="Clear" warning="0" count="1">
  <result><entity>test-host-name-1</entity><message>HostIdentity
cleared</message><status>success</status></result>
</results>

```

```

<results successful="2" failed="0" entityType="HostIdentity" operation="Delete" warning="0" count="2">
  <result><entity>12042cdc-6437-11e4-9585-002219ed08d3</entity><message>HostIdentity
deleted</message><status>success</status></result>
  <result><entity>5d33537c-293e-4564-b0cf-d0998389b31f</entity><message>HostIdentity
deleted</message><status>success</status></result>
</results>

```

Here are JSON clear and delete responses:

```

{
  "successful" : 1,
  "failed" : 0,
  "entityType" : "HostIdentity",
  "operation" : "Clear",
  "warning" : 0,
  "results" : [ {
    "entity" : "test-host-name-1",
    "status" : "success",
    "message" : "HostIdentity cleared"
  } ],
  "count" : 1
}

```

```

{
  "successful" : 2,
  "failed" : 0,
  "entityType" : "HostIdentity",
  "operation" : "Delete",
  "warning" : 0,
  "results" : [ {
    "entity" : "12c545e8-6437-11e4-82bf-002219ed08d3",
    "status" : "success",
    "message" : "HostIdentity deleted"
  }, {
    "entity" : "8fb9a880-f093-4892-a0f3-b1a403b49e5e",
    "status" : "success",
    "message" : "HostIdentity deleted"
  } ],
  "count" : 2
}

```

18.4 Host Identity Host Name Autocomplete

Get limited number of sorted existing Host Identity and Host host names that match a specified prefix. Matching and sorting is done in a case-insensitive fashion. The special wildcard prefix "*" matches all names. Autocomplete names are updated asynchronously when Host Identities or Hosts host names are modified.

18.4.1 Method: GET Host Identity Host Name Autocomplete

GET /api/hostidentities/autocomplete/{prefix}

18.4.2 Path Parameters

Field	Type	Description	Required
prefix	Path	Encoded Host Identity or Host host names prefix to match	yes

18.4.3 HTTP Headers

Header	Valid Values	Required
Accept	application/xml or application/json	True
GWOS-API-TOKEN	a valid token returned from login	True
GWOS-APP-NAME	your application name	True

18.4.4 HTTP Status Codes

Code	Description
200	Autocomplete names for prefix returned
404	No autocomplete names for prefix
401	Authentication/authorization error occurred
500	An internal server error occurred while processing autocomplete prefix

18.4.5 Example Responses

Here is an XML example of the autocomplete names returned.

XML results are always wrapped in an <names> collection element, with one or more <name> subelements.

```
<names>
  <name>localhost</name>
  <name>localhost2</name>
</names>
```

Here is a JSON example of the autocomplete returned.

JSON results are always wrapped in an object with a `names` array member, with one or more name strings.

```
{
  "names" : [ "localhost", "localhost2" ]
}
```