

19.0 Host Blacklists API

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

19.0 Host Blacklists API

19.1 Query Host Blacklists

Retrieve host blacklist records by query. Host blacklists provide a list of host names, (or host name expressions), that are used to filter invalid or disabled hosts. Queries would typically be used to drive administration browsers and similar tooling. Queried host blacklists, (one or more), are always returned wrapped in an XML `<hostBlacklists>` element or JSON object with an `hostBlacklists` array member. Getting a single host blacklist record is not wrapped.

19.1.1 Method: GET Host Blacklists

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

19.1.2 Method: GET a Single Host Blacklist by Host Name

[GET /api/hostblacklists/{hostName}](#)

19.1.3 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
hostName	Path	the host name to retrieve, (case insensitive)	yes

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

19.1.4 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

19.1.5 Query Fields

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

Field	Description	Alias
hostBlacklistId	primary key	id
hostName	primary/host host name	host

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

19.1.6 Example Queries

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

1. query for first page of all host blacklists
[GET /api/hostblacklists?count=25](#)

2. query for SERVER_2's host blacklist
GET /api/hostblacklists/SERVER_2
3. query for the second page of host blacklists that start with a prefix, (ordered by hostName)
GET /api/hostblacklists?query=hostName ilike 'server_%' ORDER BY hostName&first=25&count=25

19.1.7 HTTP Status Codes

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

19.1.8 Example Query Results

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

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

```
<hostBlacklists>
  <hostBlacklist hostBlacklistId="10" hostName="test-host-name-0"/>
  <hostBlacklist hostBlacklistId="11" hostName="test-host-name-1"/>
</hostBlacklists>
```

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

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

```
{
  "hostBlacklists" : [ {
    "hostBlacklistId" : 12,
    "hostName" : "test-host-name-0"
  }, {
    "hostBlacklistId" : 13,
    "hostName" : "test-host-name-1"
  } ]
}
```

19.1.9 Example Get Single Results

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

```
<hostBlacklist hostBlacklistId="10" hostName="test-host-name-0"/>
```

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

```
{
  "hostBlacklistId" : 12,
  "hostName" : "test-host-name-0"
}
```

19.2 Create or Update Host Blacklists

Add or update host blacklists by post. Any number of host blacklists can be posted to the server in a single request. Host blacklist ids will be generated by the server. Requests are considered updates if the host blacklist referenced by the host blacklist id exists. Updates that specify a hostName are assumed to change the host name.

19.2.1 Method: POST Host Blacklists

POST /api/hostblacklists

19.2.2 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

19.2.3 POST Data Example

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

```
<hostBlacklists>
  <hostBlacklist hostName="test-host-name-1"/>
  <hostBlacklist hostName="test-host-name-2"/>
</hostBlacklists>
```

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

```
<hostBlacklists>
  <hostBlacklist hostBlacklistId="11" hostName="test-host-name-1-changed"/>
</hostBlacklists>
```

Here is a JSON example post data creating a host blacklist, (the wrapping object with the `hostBlacklists` array member is always required, even for a single host blacklist object):

```
{
  "hostBlacklists" : [ {
    "hostName" : "test-host-name-0"
  } ]
}
```

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

```
{
  "hostBlacklists" : [ {
    "hostBlacklistId" : 13,
    "hostName" : "test-host-name-1-changed"
  } ]
}
```

19.2.4 HTTP Status Codes

Code	Description
200	Zero or more host blacklists were created
401	Authentication/authorization error occurred

500	An internal server error occurred while creating host blacklists
-----	--

19.2.5 Example Responses

Here are XML create responses:

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

```
<results successful="0" failed="1" entityType="HostBlacklist" operation="Insert" warning="0" count="1"
>
  <result><entity>test-host-name-3</entity><message>Failed to create HostBlacklist: 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 update response:

```
<results successful="1" failed="0" entityType="HostBlacklist" operation="Update" warning="0" count="1"
>
  <result><entity>1</entity><message>http:
//localhost:8080/foundation-webapp/api/hostblacklists/test-host-name-1-changed</message><status>success</
</results>
```

Here are JSON create responses:

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

```

{
  "successful" : 0,
  "failed" : 1,
  "entityType" : "HostBlacklist",
  "operation" : "Insert",
  "warning" : 0,
  "results" : [ {
    "entity" : "test-host-name-3",
    "status" : "failure",
    "message" : "Failed to create HostBlacklist: 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 update response:

```

{
  "successful" : 1,
  "failed" : 0,
  "entityType" : "HostBlacklist",
  "operation" : "Update",
  "warning" : 0,
  "results" : [ {
    "entity" : "13",
    "status" : "success",
    "message" : "http://localhost:8080/foundation-webapp/api/hostblacklists/test-host-name-1-changed"
  } ],
  "count" : 1
}

```

19.3 Delete Host Blacklists

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

19.3.1 Method: Delete Host Blacklists

DELETE /api/hostblacklists

19.3.2 Method: Delete Host Blacklists by Host Names

DELETE /api/hostblacklists/{hostNames}

19.3.3 HTTP Query and Path Parameters

Field	Type	Description	Required
hostNames	Path	list of host names to retrieve, (case insensitive, comma separated list)	True

19.3.4 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

19.3.5 POST Data Example

Here is an XML example post data deleting two host blacklists on the server, (the wrapping `<hostblacklists>` collection element is always required, even for a single host blacklist element):

```
<hostBlacklists>
  <hostBlacklist hostBlacklistId="11" hostName="test-host-name-1-changed"/>
  <hostBlacklist hostBlacklistId="12" hostName="test-host-name-2"/>
</hostBlacklists>
```

Here is a JSON example post data deleting two host blacklists, (the wrapping object with the `hostBlacklists` array member is always required, even for a single host blacklist object):

```
{
  "hostBlacklists" : [ {
    "hostBlacklistId" : 13,
    "hostName" : "test-host-name-1-changed"
  }, {
    "hostBlacklistId" : 14,
    "hostName" : "test-host-name-2"
  } ]
}
```

19.3.6 HTTP Status Codes

Code	Description
200	Zero or more host blacklists were deleted successfully
401	Authentication/authorization error occurred
500	An internal server error occurred while creating host blacklists

19.3.7 Example Responses

Here is an XML delete response:

```
<results successful="2" failed="0" entityType="HostBlacklist" operation="Delete" warning="0" count="2"
>
  <result><entity>12</entity><message>HostBlacklist
deleted</message><status>success</status></result>
  <result><entity>13</entity><message>HostBlacklist
deleted</message><status>success</status></result>
</results>
```

Here is a JSON delete response:

```
{
  "successful" : 2,
  "failed" : 0,
  "entityType" : "HostBlacklist",
  "operation" : "Delete",
  "warning" : 0,
  "results" : [ {
    "entity" : "13",
    "status" : "success",
    "message" : "HostBlacklist deleted"
  }, {
    "entity" : "14",
    "status" : "success",
    "message" : "HostBlacklist deleted"
  } ],
  "count" : 2
}
```