

22.0 Custom Group APIs

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

22.0 Custom Group APIs

22.1 Query Custom Groups

Retrieve custom groups by query. There are two kinds of queries supported:

1. Retrieving a single custom group. Retrieves exactly one custom group by name.
2. Retrieving one or more custom groups. Retrieves 1..n custom group objects wrapped by an XML `<customGroups>` collection or a JSON object with a `customGroups` array member.

22.1.1 Method: GET a Single Custom Group by Unique Name

[GET /api/customgroups/{customGroupName}](#)

22.1.2 HTTP Query and Path Parameters

Field	Type	Description	Required
customGroupName	Path	the unique name of the custom group	yes

22.1.3 Method: GET Custom Groups

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

22.1.4 HTTP Query and Path Parameters

Field	Type	Description	Required
query	Query	An encoded query string (where clause)	no **
first	Query	Paging for query. First record to start from	no
count	Query	Paging for query. Number of records to include when paging	no
appType	Query	Application type name filter for custom groups	no
agentId	Query	Agent id filter for custom groups	no

**Note: If a query parameter is not provided, all custom groups will be retrieved.

22.1.5 HTTP Headers

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

22.1.6 Query Fields

Because custom groups are implemented as categories, supported query fields are identical. However, entityType shortcuts and fully qualified fields are hardcoded for custom groups and should not be used in a query.

Field	Description	Alias
id	Custom group integer id	categoryId

name	The custom group primary unique name	
description	The description of this custom group	
root	Boolean flag indicating a root custom group of a hierarchy	
categoryEntities	A prefix for the host or service groups associated with this custom group. Valid sub-fields are: categoryEntities.categoryEntityID categoryEntities.objectID categoryEntities.entityType.name categoryEntities.entityType.entityTypeid categoryEntities.entityType.description categoryEntities.entityType.LogicalEntity categoryEntities.entityType.applicationTypeSupported	
ancestors	A prefix for the ancestor parent custom groups. Valid sub-fields include all query fields recursively	
parents	A prefix for the parent custom groups. Valid sub-fields include all query fields recursively	
children	A prefix for the child custom groups. Valid sub-fields include all query fields recursively	

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

22.1.7 Example Queries

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

1. query for all custom groups
[GET /api/customgroups](#)
2. query for all custom groups order by name descending
[GET /api/customgroups?query=order by name desc](#)
3. query for a single custom group named 'TEST-CUSTOM-GROUP-0'
[GET /api/customgroups/TEST-CUSTOM-GROUP-0](#)
4. a like query to find all custom groups starting with 'TEST-'
[GET /api/customgroups?query=name like 'TEST-%'](#)
5. query for one or more custom group names using IN query syntax
[GET /api/customgroups?query=name in \('CALL-CENTER','DATA-CENTER'\)](#)
6. query for all custom groups with application type 'VEMA'
[GET /api/customgroups/?appType=VEMA](#)
7. query for custom groups with a specific host group entity
[GET /api/customgroups/?query=categoryEntities.objectID = 512 and categoryEntities.entityType.name = 'HOST_GROUP'](#)
8. query for custom groups with a specific service group entity and agent id '007'
[GET /api/customgroups/?query=categoryEntities.objectID = 1024 and categoryEntities.entityType.name = 'SERVICE_GROUP'&agentId=007](#)

22.1.8 HTTP Status Codes

200 - One or more query results returned
404 - No query results found
500 - An internal server error occurred

22.1.9 Example Responses

Results of requesting a single entity is always wrapped with a single <customGroup> XML element. Here is an XML example of the result of a query finding one custom group. All simple fields are displayed as attributes. Collections of host and service groups appear as subelements.

```
<customGroup id="82" name="TEST-CUSTOM-GROUP-0" description="TEST-CUSTOM-GROUP-0" appType="NAGIOS"
agentId="TEST-AGENT" root="true">
  <hostGroups>
    <hostGroup id="20" name="TEST-HOST-GROUP-0" alias="TEST-HOST-GROUP-0" appType="NAGIOS" />
  </hostGroups>
</customGroup>
```

Result of queries are always wrapped in a <customGroups> XML collection element, with one or more <customGroup> subelements.

```
<customGroups>
  <customGroup id="82" name="TEST-CUSTOM-GROUP-0" description="TEST-CUSTOM-GROUP-0" appType="NAGIOS"
agentId="TEST-AGENT" root="true">
  <hostGroups>
    <hostGroup id="20" name="TEST-HOST-GROUP-0" alias="TEST-HOST-GROUP-0" appType="NAGIOS"/>
  </hostGroups>
</customGroup>
<customGroup id="83" name="TEST-CUSTOM-GROUP-1" agentId="TEST-AGENT" root="true">
  <serviceGroups>
    <serviceGroup id="77" name="TEST-SERVICE-GROUP-0"/>
  </serviceGroups>
</customGroup>
<customGroup id="84" name="TEST-CUSTOM-GROUP-2" appType="NAGIOS" root="true"/>
<customGroup id="85" name="TEST-CUSTOM-GROUP-3" root="true"/>
</customGroups>
```

Here is a JSON example of the result of a query finding one custom group.

```
{
  "id" : 78,
  "name" : "TEST-CUSTOM-GROUP-0",
  "description" : "TEST-CUSTOM-GROUP-0",
  "appType" : "NAGIOS",
  "agentId" : "TEST-AGENT",
  "root" : true,
  "hostGroups" : [ {
    "id" : 20,
    "name" : "TEST-HOST-GROUP-0",
    "alias" : "TEST-HOST-GROUP-0",
    "appType" : "NAGIOS"
  } ]
}
```

The array results of queries are always wrapped in an object with a field named `customGroups`.

```

{
  "customGroups" : [ {
    "id" : 78,
    "name" : "TEST-CUSTOM-GROUP-0",
    "description" : "TEST-CUSTOM-GROUP-0",
    "appType" : "NAGIOS",
    "agentId" : "TEST-AGENT",
    "root" : true,
    "hostGroups" : [ {
      "id" : 20,
      "name" : "TEST-HOST-GROUP-0",
      "alias" : "TEST-HOST-GROUP-0",
      "appType" : "NAGIOS"
    } ]
  }, {
    "id" : 79,
    "name" : "TEST-CUSTOM-GROUP-1",
    "agentId" : "TEST-AGENT",
    "root" : true,
    "serviceGroups" : [ {
      "id" : 77,
      "name" : "TEST-SERVICE-GROUP-0"
    } ]
  }, {
    "id" : 80,
    "name" : "TEST-CUSTOM-GROUP-2",
    "appType" : "NAGIOS",
    "root" : true
  }, {
    "id" : 81,
    "name" : "TEST-CUSTOM-GROUP-3",
    "root" : true
  } ]
}

```

22.2 Create or Update Custom Groups

Persist a batch (1..n) of custom groups in foundation database. Custom groups will be created if they do not exist. If a custom group exists, it will be updated. You can also specify one or more host or service groups to be associated with the custom groups. Specified host or service groups will replace existing groups.

If one or more custom group creation operations fails, others may still succeed. This is not an all-or-none transactional operation. The results of each individual creation/update operation is returned back in the resultset described below with a status of success or failure.

22.2.1 Method: POST

[POST /api/customgroups](#)

22.2.2 HTTP Headers

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

22.2.3 Post Data Attributes and Elements

Field	Description	Required	Type
name	The unique custom group name	Yes	Attribute/Field
description	The description of this custom group	No	Attribute/Field

appType	Application type name of this custom group	No	Attribute/Field
agentId	Agent id of this custom group	No	Attribute/Field
hostGroupNames	Host group names to be associated with this custom group	No	Elements/Array Field
serviceGroupNames	Service group names to be associated with this custom group	No	Elements/Array Field

22.2.4 Example Post Data

As with the results of queries, custom groups are always wrapped in a `<customGroups>` XML collection element, with one or more `<customGroup>` subelements. For JSON, custom groups are always wrapped in an object with a field named `customGroups`.

```
<customGroups>
  <customGroup name="TEST-CUSTOM-GROUP-0" description="TEST-CUSTOM-GROUP-0" appType="NAGIOS" agentId="TEST-AGENT">
    <hostGroupNames>
      <hostGroupName>TEST-HOST-GROUP-0</hostGroupName>
    </hostGroupNames>
  </customGroup>
  <customGroup name="TEST-CUSTOM-GROUP-1" agentId="TEST-AGENT">
    <serviceGroupNames>
      <serviceName>TEST-SERVICE-GROUP-0</serviceName>
    </serviceGroupNames>
  </customGroup>
  <customGroup name="TEST-CUSTOM-GROUP-2" appType="NAGIOS"/>
  <customGroup name="TEST-CUSTOM-GROUP-3"/>
</customGroups>
```

```
{
  "customGroups" : [ {
    "name" : "TEST-CUSTOM-GROUP-0",
    "description" : "TEST-CUSTOM-GROUP-0",
    "appType" : "NAGIOS",
    "agentId" : "TEST-AGENT",
    "hostGroupNames" : [ "TEST-HOST-GROUP-0" ]
  }, {
    "name" : "TEST-CUSTOM-GROUP-1",
    "agentId" : "TEST-AGENT",
    "serviceGroupNames" : [ "TEST-SERVICE-GROUP-0" ]
  }, {
    "name" : "TEST-CUSTOM-GROUP-2",
    "appType" : "NAGIOS"
  }, {
    "name" : "TEST-CUSTOM-GROUP-3"
  } ]
}
```

22.2.5 HTTP Status Codes

200 - Zero or more custom groups were created without any internal server errors
500 - An internal server error occurred

22.2.6 Example Responses

In this example, we use the POST data above: four custom groups were successfully created.

Note that the results collection also returns the location of the created or updated custom group which can be directly used by the GET operation.

```

<results successful="4" failed="0" entityType="CustomGroup" operation="Update" warning="0" count="4">
  <result>
    <entity>TEST-CUSTOM-GROUP-0</entity>
    <location>http://localhost:8080/foundation-webapp/api/customgroups/TEST-CUSTOM-GROUP-0</location>
  <status>success</status>
</result>
  <result>
    <entity>TEST-CUSTOM-GROUP-1</entity>
    <location>http://localhost:8080/foundation-webapp/api/customgroups/TEST-CUSTOM-GROUP-1</location>
  <status>success</status>
</result>
  <result>
    <entity>TEST-CUSTOM-GROUP-2</entity>
    <location>http://localhost:8080/foundation-webapp/api/customgroups/TEST-CUSTOM-GROUP-2</location>
  <status>success</status>
</result>
  <result>
    <entity>TEST-CUSTOM-GROUP-3</entity>
    <location>http://localhost:8080/foundation-webapp/api/customgroups/TEST-CUSTOM-GROUP-3</location>
  <status>success</status>
</result>
</results>

```

```

{
  "successful" : 4,
  "failed" : 0,
  "entityType" : "CustomGroup",
  "operation" : "Update",
  "warning" : 0,
  "results" : [ {
    "entity" : "TEST-CUSTOM-GROUP-0",
    "status" : "success",
    "location" : "http://localhost:8080/foundation-webapp/api/customgroups/TEST-CUSTOM-GROUP-0"
  }, {
    "entity" : "TEST-CUSTOM-GROUP-1",
    "status" : "success",
    "location" : "http://localhost:8080/foundation-webapp/api/customgroups/TEST-CUSTOM-GROUP-1"
  }, {
    "entity" : "TEST-CUSTOM-GROUP-2",
    "status" : "success",
    "location" : "http://localhost:8080/foundation-webapp/api/customgroups/TEST-CUSTOM-GROUP-2"
  }, {
    "entity" : "TEST-CUSTOM-GROUP-3",
    "status" : "success",
    "location" : "http://localhost:8080/foundation-webapp/api/customgroups/TEST-CUSTOM-GROUP-3"
  } ],
  "count" : 4
}

```

22.3 Add or Delete Custom Group Members

Host and service groups are associated with custom groups and can be manipulated via these PUT operations.

22.3.1 Method: PUT adding members with POST Data

[PUT /api/customgroups/addmembers](#)

22.3.2 Method: PUT deleting members with POST Data

[PUT /api/customgroups/deletemembers](#)

22.3.3 HTTP Headers

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

22.3.4 Post Data Attributes and Elements

Field	Description	Required	Type
name	The unique custom group name	Yes	Attribute/Field
hostGroupNames	Host group names to be added to or removed from the custom group	No	Elements/Array Field
serviceGroupNames	Service group names to be added to or removed from the custom group	No	Elements/Array Field

22.3.5 Example POST Data

```
<customGroup name="TEST-CUSTOM-GROUP-0">
  <hostGroupNames>
    <hostGroupName>TEST-HOST-GROUP-1</hostGroupName>
  </hostGroupNames>
</customGroup>
```

```
{
  "name" : "TEST-CUSTOM-GROUP-0",
  "hostGroupNames" : [ "TEST-HOST-GROUP-1" ]
}
```

22.3.6 HTTP Status Codes

200 - Custom groups were updated without any internal server errors
500 - An internal server error occurred

22.3.7 Example Responses

Note that the results collection also returns the location of the updated custom group which can be directly used by the GET operation.

```
<results successful="1" failed="0" entityType="CustomGroup" operation="Update" warning="0" count="1">
  <result>
    <entity>TEST-CUSTOM-GROUP-0</entity>
    <location>http://localhost:8080/foundation-webapp/api/customgroups/TEST-CUSTOM-GROUP-0</location>
  <status>success</status>
</result>
</results>
```

```

{
  "successful" : 1,
  "failed" : 0,
  "entityType" : "CustomGroup",
  "operation" : "Update",
  "warning" : 0,
  "results" : [ {
    "entity" : "TEST-CUSTOM-GROUP-0",
    "status" : "success",
    "location" : "http://localhost:8080/foundation-webapp/api/customgroups/TEST-CUSTOM-GROUP-0"
  } ],
  "count" : 1
}

```

22.4 Delete Custom Groups

Deletes one or more custom groups from the Collage database. Deletes are specified with POST data, passing in custom group names.

Note that while deleting hierarchical custom groups is supported for backward compatibility, this is not recommended. Hierarchical custom group delete operations should be carried out using the hierarchy-specific APIs for categories.

22.4.1 Method: DELETE with POST Data

[DELETE /api/customgroups](#)

22.4.2 HTTP Headers

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

22.4.3 Post Data Attributes and Elements

Field	Description	Required	Type
name	The unique custom group name	Yes	Attribute/Field

22.4.4 Example POST Data

The post data should be formatted like a POST payload, but the only required field is the custom group name.

```

<customGroups>
  <customGroup name="TEST-CUSTOM-GROUP-1"/>
  <customGroup name="TEST-CUSTOM-GROUP-2"/>
</customGroups>

```

```

{
  "customGroups" : [ {
    "name" : "TEST-CUSTOM-GROUP-1"
  }, {
    "name" : "TEST-CUSTOM-GROUP-2"
  } ]
}

```

22.4.5 HTTP Status Codes

200 - Custom groups were deleted successfully
500 - An internal server error occurred

22.4.6 Example Responses

```
<results successful="2" failed="0" entityType="CustomGroup" operation="Delete" warning="0" count="2">  
  <result>  
    <entity>TEST-CUSTOM-GROUP-1</entity>  
    <message>Custom Group TEST-CUSTOM-GROUP-1 deleted.</message>  
    <status>success</status>  
  </result>  
  <result>  
    <entity>TEST-CUSTOM-GROUP-2</entity>  
    <message>Custom Group TEST-CUSTOM-GROUP-2 deleted.</message>  
    <status>success</status>  
  </result>  
</results>
```

```
{  
  "successful" : 2,  
  "failed" : 0,  
  "entityType" : "CustomGroup",  
  "operation" : "Delete",  
  "warning" : 0,  
  "results" : [ {  
    "entity" : "TEST-CUSTOM-GROUP-1",  
    "status" : "success",  
    "message" : "Custom Group TEST-CUSTOM-GROUP-1 deleted."  
  }, {  
    "entity" : "TEST-CUSTOM-GROUP-2",  
    "status" : "success",  
    "message" : "Custom Group TEST-CUSTOM-GROUP-2 deleted."  
  } ],  
  "count" : 2  
}
```

22.5 Custom Group Name Autocomplete

Get limited number of sorted existing Custom Group 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 Custom Group names are modified.

22.5.1 Method: GET Custom Group Name Autocomplete

[GET /api/customgroups/autocomplete/{prefix}](#)

22.5.2 Path Parameters

Field	Type	Description	Required
prefix	Path	Encoded Custom Group name prefix to match	yes

22.5.3 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

22.5.4 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

22.5.5 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>CG1</name>
  <name>CG2</name>
</names>
```

Here is a JSON example of the autocomplete names returned.

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

```
{
  "names" : [ "CG1", "CG2" ]
}
```