

30.0 Biz Feeder APIs

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

30.0 Biz Feeder APIs

30.1 Form Create or Update Host

Create or update a single host accepting POST form parameters. In addition to the host, host groups, host categories, events, and notifications are created, updated, and/or sent. Events and notifications are created and sent on status change. New hosts are always created in the `PENDING` state unless the `setStatusOnCreate` flag is set. The created or updated host is returned.

30.1.1 Method: POST Form Create or Update Host

[POST /api/biz/host](#)

30.1.2 POST Data Example

Example form create post data.

```
message=PENDING+-+new
&host=host5&status=PENDING&device=10.0.0.5&agentId=a4ca80e0-e855-43ee-86d8-f4e499f0a824&hostGroup=hosts&
```

Example form update post data.

```
message=UP&host=host5&status=UP&agentId=a4ca80e0-e855-43ee-86d8-f4e499f0a824&hostGroup=hosts&appType=NAGI
```

30.1.3 POST Parameters

Name	Description	Required
agentId	agent id	True
allowInserts	allow host inserts	False, defaults to TRUE
appType	application type name	True
checkIntervalMinutes	check interval in minutes	False, defaults to 5
device	device identification	True, for create
host	host name	True
hostCategory	host category name	False
hostGroup	host group name	False
mergeHosts	allow hosts to be merged if host name is case-insensitive equal	False, defaults to TRUE
message	host last plugin output message	False
setStatusOnCreate	sets specified status on create after creating in <code>PENDING</code> state	False, defaults to FALSE
status	host status	True

30.1.4 HTTP Headers

Header	Valid Values	Required
Content-Type	application/x-www-form-urlencoded	False

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

30.1.5 HTTP Status Codes

Code	Description
200	Authorized hosts and services returned
401	Authentication/authorization error occurred
500	An internal server error occurred while updating host

30.1.6 Example Results

Here is an XML example. The single deep host with services, (not wrapped), is returned.

```
<host id="7" hostName="host5" description="host5" monitorStatus="UP" appType="NAGIOS"
deviceIdentification="10.0.0.5" deviceDisplayName="10.0.0.5" lastCheckTime=
"2016-01-04T23:42:30.468-0700" nextCheckTime="2016-01-04T23:47:30.468-0700" bubbleUpStatus="UP"
serviceAvailability="0" acknowledged="false" agentId="a4ca80e0-e855-43ee-86d8-f4e499f0a824"
serviceCount="1" stateType="HARD" checkType="ACTIVE" lastStateChange="2016-01-04T23:42:30.000-0700"
lastPlugInOutput="-1^^-1^^-1^^^01/04/2016 11:42:30 PM^^UP^^01/04/2016 11:47:30 PM">
  <properties>
    <property name="LastPlugInOutput" value="UP"/>
    <property name="LastStateChange" value="2016-01-04T23:42:30.000-0700"/>
  </properties>
  <device id="6" displayName="10.0.0.5" identification="10.0.0.5"/>
  <hostStatus hostStatusId="7" lastCheckTime="2016-01-04T23:42:30.468-0700" nextCheckTime=
"2016-01-04T23:47:30.468-0700">
    <monitorStatus monitorStatusId="1" name="UP" description="UP"/><checkType checkTypeId="1"
name="ACTIVE" description="Active Check"/>
    <stateType>
      <stateTypeId>2</stateTypeId>
      <name>HARD</name>
      <description>State Hard</description>
    </stateType>
  </hostStatus>
  <services>
    <service id="86" appType="NAGIOS" description="local_cpu_httpd" monitorStatus="PENDING"
lastCheckTime="2016-01-04T23:42:30.000-0700" nextCheckTime="2016-01-04T23:47:30.000-0700" hostName=
"host5" stateType="HARD" checkType="ACTIVE" lastHardState="PENDING" agentId=
"a4ca80e0-e855-43ee-86d8-f4e499f0a824" monitorServer="localhost" deviceIdentification="10.0.0.5"
lastPlugInOutput="-1^^-1^^-1^^^01/01/1970 12:00:00 AM^^PENDING - new^^NA">
      <properties>
        <property name="LastPlugInOutput" value="PENDING - new"/>
      </properties>
    </service>
  </services>
  <hostGroups>
    <hostGroup id="6" name="hosts" appType="NAGIOS"/>
  </hostGroups>
  <applicationType id="100" name="NAGIOS" description="System monitored by Nagios"
stateTransitionCriteria="Device;Host;ServiceDescription"/>
</host>
```

Here is a JSON example. The single deep host object with services is returned.

```
{
  "properties" : {
    "LastPlugInOutput" : "UP",
```

```
"LastStateChange" : "2016-01-04T23:42:31.000-0700"
},
"id" : 9,
"hostName" : "host5",
"description" : "host5",
"monitorStatus" : "UP",
"appType" : "NAGIOS",
"deviceIdentification" : "10.0.0.5",
"deviceDisplayName" : "10.0.0.5",
"lastCheckTime" : "2016-01-04T23:42:31.906-0700",
"nextCheckTime" : "2016-01-04T23:47:31.906-0700",
"bubbleUpStatus" : "UP",
"serviceAvailability" : "0",
"acknowledged" : false,
"agentId" : "a4ca80e0-e855-43ee-86d8-f4e499f0a824",
"device" : {
  "id" : 8,
  "displayName" : "10.0.0.5",
  "identification" : "10.0.0.5"
},
"hostStatus" : {
  "hostStatusId" : 9,
  "hostMonitorStatus" : {
    "monitorStatusId" : 1,
    "name" : "UP",
    "description" : "UP"
  },
  "lastCheckTime" : "2016-01-04T23:42:31.906-0700",
  "checkType" : {
    "checkTypeId" : 1,
    "name" : "ACTIVE",
    "description" : "Active Check"
  },
  "nextCheckTime" : "2016-01-04T23:47:31.906-0700",
  "stateType" : {
    "stateTypeId" : 2,
    "name" : "HARD",
    "description" : "State Hard"
  }
},
},
"services" : [ {
  "properties" : {
    "LastPluginOutput" : "PENDING - new"
  },
  "id" : 89,
  "appType" : "NAGIOS",
  "description" : "local_cpu_httpd",
  "monitorStatus" : "PENDING",
  "lastCheckTime" : "2016-01-04T23:42:31.000-0700",
  "nextCheckTime" : "2016-01-04T23:47:31.000-0700",
  "hostName" : "host5",
  "stateType" : "HARD",
  "checkType" : "ACTIVE",
  "lastHardState" : "PENDING",
  "agentId" : "a4ca80e0-e855-43ee-86d8-f4e499f0a824",
  "monitorServer" : "localhost",
  "deviceIdentification" : "10.0.0.5",
  "lastPlugInOutput" : "-1^^-1^^-1^^01/01/1970 12:00:00 AM^^PENDING - new^^NA"
} ],
"serviceCount" : 1,
"hostGroups" : [ {
  "id" : 7,
  "name" : "hosts",
  "appType" : "NAGIOS"
} ],
"applicationType" : {
  "properties" : {
  },
  "id" : 100,
  "name" : "NAGIOS",
```

```
"description" : "System monitored by Nagios",  
  "stateTransitionCriteria" : "Device;Host;ServiceDescription"  
},  
"stateType" : "HARD",  
"checkType" : "ACTIVE",  
"lastStateChange" : "2016-01-04T23:42:31.000-0700",
```

```

"lastPlugInOutput" : "-1^^-1^^-1^^01/04/2016 11:42:31 PM^^UP^^01/04/2016 11:47:31 PM"
}

```

30.2 Form Create or Update Service

Create or update a single service accepting POST form parameters. Will create the specified host if required and device identification is specified. In addition to the host, host groups, host categories, events, and notifications are created, updated, and/or sent. Events and notifications are created and sent on status change. New hosts and services are always created in the `PENDING` state unless the `setStatusOnCreate` flag is set. Service hosts are set to the `UP` status otherwise. The created or updated service is returned.

30.2.1 Method: POST Form Create or Update Service

[POST /api/biz/service](#)

30.2.2 POST Data Example

Example form create post data.

```

message=PENDING+--+new
&host=host5&status=PENDING&serviceGroup=cpu&device=10.0.0.5&service=local_cpu_httpd&agentId=a4ca80e0-e855

```

Example form update post data.

```

message=OK+--+10&host=host5&status=OK&service=local_cpu_httpd&serviceValue=30&warningLevel=75&criticalLevel=

```

30.2.3 POST Parameters

Name	Description	Required
agentId	agent id	True
allowInserts	allow host inserts	False, defaults to TRUE
appType	application type name	True
checkIntervalMinutes	check interval in minutes	False, defaults to 5
criticalLevel	service value critical level	False, defaults to -1
device	device identification	True, for create
host	host name	True
hostCategory	host category name	False
hostGroup	host group name	False
mergeHosts	allow hosts to be merged if host name is case-insensitive equal	False, defaults to TRUE
message	service last plugin output message	False
metricType	service metric type	False
service	service description	True
serviceCategory	service category name	False
serviceGroup	service group name	False
serviceValue	service value	False
setStatusOnCreate	sets specified status on create after creating in <code>PENDING</code> state	False, defaults to FALSE
status	service status	True
warningLevel	service value warning level	False, defaults to -1

30.2.4 HTTP Headers

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

30.2.5 HTTP Status Codes

Code	Description
200	Authorized hosts and services returned
401	Authentication/authorization error occurred
500	An internal server error occurred while updating service

30.2.6 Example Results

Here is an XML example. The single service, (not wrapped), is returned.

```
<service id="86" appType="NAGIOS" description="local_cpu_httpd" monitorStatus="PENDING" lastCheckTime="2016-01-04T23:42:30.000-0700" nextCheckTime="2016-01-04T23:47:30.000-0700" hostName="host5" stateType="HARD" checkType="ACTIVE" lastHardState="PENDING" agentId="a4ca80e0-e855-43ee-86d8-f4e499f0a824" monitorServer="localhost" deviceIdentification="10.0.0.5" lastPlugInOutput="-1^^-1^^-1^^01/01/1970 12:00:00 AM^^PENDING - new^^NA">
  <properties>
    <property name="LastPluginOutput" value="PENDING - new"/>
  </properties>
</service>
```

Here is a JSON example. The single JSON host object is returned.

```
{
  "properties": {
    "LastPluginOutput": "OK - 10"
  },
  "id": 89,
  "appType": "NAGIOS",
  "description": "local_cpu_httpd",
  "monitorStatus": "OK",
  "lastCheckTime": "2016-01-04T23:42:31.000-0700",
  "nextCheckTime": "2016-01-04T23:47:31.000-0700",
  "lastStateChange": "2016-01-04T23:42:31.000-0700",
  "hostName": "host5",
  "stateType": "HARD",
  "checkType": "ACTIVE",
  "lastHardState": "OK",
  "agentId": "a4ca80e0-e855-43ee-86d8-f4e499f0a824",
  "monitorServer": "localhost",
  "deviceIdentification": "10.0.0.5",
  "lastPlugInOutput": "-1^^-1^^-1^^01/04/2016 11:42:31 PM^^OK - 10^^NA"
}
```

30.3 Create or Update Hosts

Primary feeder interface used by CloudHub and updated feeders. Creates or updates multiple hosts with or without nested services. In addition to the hosts and services, host groups, host categories, service groups, service categories, events, and notifications are created, updated, and/or sent. Events and notifications are created and sent on host and service status change. New hosts and services are always created in the

PENDING state unless the `setStatusOnCreate` flag is set. When nested services are specified, some service parameters can be inherited from the parent host: `message`, `appType`, and `agentId`. The standard API operation results are returned. General API conventions are followed wrapping POST data and results for XML and JSON content types.

30.3.1 Method: POST Create or Update Hosts

[POST /api/biz/hosts](#)

30.3.2 POST Data Example

Here are XML examples of post data without and with nested services. General API conventions are followed wrapping multiple hosts with XML tags.

```
<hosts>
  <host host="host6" status="PENDING" message="PENDING - new" hostGroup="hosts" device="10.0.0.6"
  appType="NAGIOS" agentId="a4ca80e0-e855-43ee-86d8-f4e499f0a824" checkIntervalMinutes="5" allowInserts=
  "true" mergeHosts="true" setStatusOnCreate="false">
    <services/>
  </host>
</hosts>
```

```
<hosts>
  <host host="host6" status="UP" message="UP" hostGroup="hosts" device="10.0.0.6" appType="NAGIOS"
  agentId="a4ca80e0-e855-43ee-86d8-f4e499f0a824" checkIntervalMinutes="5" allowInserts="true"
  mergeHosts="true" setStatusOnCreate="true">
    <services>
      <service host="host6" service="local_cpu_httpd" status="OK" message="OK - 10"
      serviceGroup="cpu" hostGroup="hosts" device="10.0.0.6" appType="NAGIOS" agentId=
      "a4ca80e0-e855-43ee-86d8-f4e499f0a824" serviceValue="10" metricType="vm" warningLevel="50"
      criticalLevel="90" checkIntervalMinutes="5" allowInserts="true" mergeHosts="true" setStatusOnCreate="
      true"/>
      <service host="host6" service="local_cpu_java" status="OK" message="OK - 20" serviceGroup=
      "cpu" hostGroup="hosts" device="10.0.0.6" appType="NAGIOS" agentId=
      "a4ca80e0-e855-43ee-86d8-f4e499f0a824" serviceValue="20" metricType="vm" warningLevel="75"
      criticalLevel="90" checkIntervalMinutes="5" allowInserts="true" mergeHosts="true" setStatusOnCreate="
      true"/>
    </services>
  </host>
</hosts>
```

Here are JSON examples of post data without and with nested services. General API conventions are followed wrapping multiple hosts with a JSON object and collection.

```
{
  "hosts" : [ {
    "properties" : {
    },
    "host" : "host6",
    "status" : "PENDING",
    "message" : "PENDING - new",
    "hostGroup" : "hosts",
    "device" : "10.0.0.6",
    "appType" : "NAGIOS",
    "agentId" : "a4ca80e0-e855-43ee-86d8-f4e499f0a824",
    "checkIntervalMinutes" : 5,
    "allowInserts" : true,
    "mergeHosts" : true,
    "setStatusOnCreate" : false,
    "services" : [ ]
  } ]
}
```

```

{
  "hosts" : [ {
    "properties" : {
    },
    "host" : "host6",
    "status" : "UP",
    "message" : "UP",
    "hostGroup" : "hosts",
    "device" : "10.0.0.6",
    "appType" : "NAGIOS",
    "agentId" : "a4ca80e0-e855-43ee-86d8-f4e499f0a824",
    "checkIntervalMinutes" : 5,
    "allowInserts" : true,
    "mergeHosts" : true,
    "setStatusOnCreate" : false,
    "services" : [ {
      "properties" : {
      },
      "host" : "host6",
      "service" : "local_cpu_httpd",
      "status" : "OK",
      "message" : "OK - 10",
      "serviceGroup" : "cpu",
      "hostGroup" : "hosts",
      "device" : "10.0.0.6",
      "appType" : "NAGIOS",
      "agentId" : "a4ca80e0-e855-43ee-86d8-f4e499f0a824",
      "serviceValue" : "10",
      "metricType" : "vm",
      "warningLevel" : 50,
      "criticalLevel" : 90,
      "checkIntervalMinutes" : 5,
      "allowInserts" : true,
      "mergeHosts" : true,
      "setStatusOnCreate" : true
    }, {
      "properties" : {
      },
      "host" : "host6",
      "service" : "local_cpu_java",
      "status" : "OK",
      "message" : "OK - 20",
      "serviceGroup" : "cpu",
      "hostGroup" : "hosts",
      "device" : "10.0.0.6",
      "appType" : "NAGIOS",
      "agentId" : "a4ca80e0-e855-43ee-86d8-f4e499f0a824",
      "serviceValue" : "20",
      "metricType" : "vm",
      "warningLevel" : 75,
      "criticalLevel" : 90,
      "checkIntervalMinutes" : 5,
      "allowInserts" : true,
      "mergeHosts" : true,
      "setStatusOnCreate" : true
    } ]
  } ]
} ]
}

```

30.3.3 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

30.3.4 HTTP Status Codes

Code	Description
200	Authorized hosts and services returned
401	Authentication/authorization error occurred
500	An internal server error occurred while updating hosts

30.3.5 Example Results

Here are XML examples. General API conventions are followed returning results. Note the successful operations for individual entities, (Insert or Update), are returned in the message elements.

```
<results successful="1" failed="0" entityType="Host" operation="Update" warning="0" count="1">
  <result>
    <entity>host6</entity>
    <location>http://localhost:8080/foundation-webapp/api/hosts/host6</location>
    <status>success</status>
    <message>Insert</message>
  </result>
</results>
```

```
<results successful="3" failed="0" entityType="Host" operation="Update" warning="0" count="3">
  <result>
    <entity>host6</entity>
    <location>http://localhost:8080/foundation-webapp/api/hosts/host6</location>
    <status>success</status>
    <message>Update</message>
  </result>
  <result>
    <entity>host6:local_cpu_httpd</entity>
    <location>http://localhost:8080/foundation-webapp/api/services/local_cpu_httpd%3FhostName=host6</location>
    <status>success</status>
    <message>Update</message>
  </result>
  <result>
    <entity>host6:local_cpu_java</entity>
    <location>http://localhost:8080/foundation-webapp/api/services/local_cpu_java%3FhostName=host6</location>
    <status>success</status>
    <message>Update</message>
  </result>
</results>
```

Here are JSON examples. General API conventions are followed returning results. Note the successful operations for individual entities, (Insert or Update), are returned in the message fields.

```

{
  "successful" : 1,
  "failed" : 0,
  "entityType" : "Host",
  "operation" : "Update",
  "warning" : 0,
  "results" : [ {
    "entity" : "host6",
    "status" : "success",
    "location" : "http://localhost:8080/foundation-webapp/api/hosts/host6",
    "message" : "Insert"
  } ],
  "count" : 1
}

```

```

{
  "successful" : 3,
  "failed" : 0,
  "entityType" : "Host",
  "operation" : "Update",
  "warning" : 0,
  "results" : [ {
    "entity" : "host6",
    "status" : "success",
    "location" : "http://localhost:8080/foundation-webapp/api/hosts/host6",
    "message" : "Update"
  }, {
    "entity" : "host6:local_cpu_httpd",
    "status" : "success",
    "location" : "http://localhost:8080/foundation-webapp/api/services/local_cpu_httpd%3FhostName=host6",
    "message" : "Update"
  }, {
    "entity" : "host6:local_cpu_java",
    "status" : "success",
    "location" : "http://localhost:8080/foundation-webapp/api/services/local_cpu_java%3FhostName=host6",
    "message" : "Update"
  } ],
  "count" : 3
}

```

30.4 Create or Update Services

Creates or updates multiple services; hosts are created if they do not already exist. In addition to the services and hosts, host groups, host categories, service groups, service categories, events, and notifications are created, updated, and/or sent. Events and notifications are created and sent on status change. New services and hosts are always created in the `PENDING` state unless the `setStatusOnCreate` flag is set. Service hosts are set to the `UP` status otherwise. The standard API operation results are returned. General API conventions are followed wrapping POST data and results for XML and JSON content types.

30.4.1 Method: POST Create or Update Services

[POST /api/biz/services](#)

30.4.2 POST Data Example

Here is an XML example post data. General API conventions are followed wrapping multiple services with XML tags.

```

<services>
  <service host="host6" service="local_cpu_httpd" status="PENDING" message="PENDING - new"
serviceGroup="cpu" hostGroup="hosts" device="10.0.0.6" appType="NAGIOS" agentId=
"a4ca80e0-e855-43ee-86d8-f4e499f0a824" warningLevel="-1" criticalLevel="-1" checkIntervalMinutes="5"
allowInserts="true" mergeHosts="true" setStatusOnCreate="false"/>
  <service host="host6" service="local_cpu_java" status="PENDING" message="PENDING - new"
serviceGroup="cpu" hostGroup="hosts" device="10.0.0.6" appType="NAGIOS" agentId=
"a4ca80e0-e855-43ee-86d8-f4e499f0a824" warningLevel="-1" criticalLevel="-1" checkIntervalMinutes="5"
allowInserts="true" mergeHosts="true" setStatusOnCreate="false"/>
</services>

```

Here is a JSON example post data. General API conventions are followed wrapping multiple hosts with a JSON object and collection.

```

{
  "services" : [ {
    "properties" : {
    },
    "host" : "host6",
    "service" : "local_cpu_httpd",
    "status" : "PENDING",
    "message" : "PENDING - new",
    "serviceGroup" : "cpu",
    "hostGroup" : "hosts",
    "device" : "10.0.0.6",
    "appType" : "NAGIOS",
    "agentId" : "a4ca80e0-e855-43ee-86d8-f4e499f0a824",
    "warningLevel" : -1,
    "criticalLevel" : -1,
    "checkIntervalMinutes" : 5,
    "allowInserts" : true,
    "mergeHosts" : true,
    "setStatusOnCreate" : false
  }, {
    "properties" : {
    },
    "host" : "host6",
    "service" : "local_cpu_java",
    "status" : "PENDING",
    "message" : "PENDING - new",
    "serviceGroup" : "cpu",
    "hostGroup" : "hosts",
    "device" : "10.0.0.6",
    "appType" : "NAGIOS",
    "agentId" : "a4ca80e0-e855-43ee-86d8-f4e499f0a824",
    "warningLevel" : -1,
    "criticalLevel" : -1,
    "checkIntervalMinutes" : 5,
    "allowInserts" : true,
    "mergeHosts" : true,
    "setStatusOnCreate" : false
  } ]
}

```

30.4.3 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

30.4.4 HTTP Status Codes

Code	Description
200	Authorized hosts and services returned
401	Authentication/authorization error occurred
500	An internal server error occurred while updating services

30.4.5 Example Results

Here is an XML example. General API conventions are followed wrapping results. Note the successful operations for individual entities, (`Insert` or `Update`), are returned in the message elements. The status of hosts involved in the operation are also returned.

```
<results successful="3" failed="0" entityType="ServiceStatus" operation="Update" warning="0" count="3"
>
  <result>
    <entity>host6</entity>
    <location>http://localhost:8080/foundation-webapp/api/hosts/host6</location>
    <status>success</status>
    <message>Insert</message>
  </result>
  <result>
    <entity>host6:local_cpu_httpd</entity>
    <location>http://localhost:8080/foundation-webapp/api/services/local_cpu_httpd%3FhostName=host6</location>
    <status>success</status>
    <message>Insert</message>
  </result>
  <result>
    <entity>host6:local_cpu_java</entity>
    <location>http://localhost:8080/foundation-webapp/api/services/local_cpu_java%3FhostName=host6</location>
    <status>success</status>
    <message>Insert</message>
  </result>
</results>
```

Here is a JSON example. General API conventions are followed returning results. Note the successful operations for individual entities, (`Insert` or `Update`), are returned in the message fields. The status of hosts involved in the operation are also returned.

```
{
  "successful" : 3,
  "failed" : 0,
  "entityType" : "ServiceStatus",
  "operation" : "Update",
  "warning" : 0,
  "results" : [ {
    "entity" : "host6",
    "status" : "success",
    "location" : "http://localhost:8080/foundation-webapp/api/hosts/host6",
    "message" : "Insert"
  }, {
    "entity" : "host6:local_cpu_httpd",
    "status" : "success",
    "location" : "http://localhost:8080/foundation-webapp/api/services/local_cpu_httpd%3FhostName=host6",
    "message" : "Insert"
  }, {
    "entity" : "host6:local_cpu_java",
    "status" : "success",
    "location" : "http://localhost:8080/foundation-webapp/api/services/local_cpu_java%3FhostName=host6",
    "message" : "Insert"
  } ],
  "count" : 3
}
```