16.0 Performance Data APIs

WAS THIS PAGE HELPFUL? Leave Feedback

16.0 Performance Data APIs

16.1 Create Perf Data

Send a batch (1..n) of Performance Data records to the GroundWork Server

If one or more Perf Data messages fails, others may still succeed. This is not an all-or-none transactional operation. The results of each individual Perf Data message is returned back in the resultset described below with a status of success or failure.

16.1.1 Method: POST

POST /api/perfdata

16.1.2 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

16.1.3 Post Data Attributes and Elements

Field	Description	Required
appType	the application type of this service	True
label	A text string label describing this service name that is attached to the plotted line graph	False
serverName	the Groundwork Host name	True
serverTime	total number of seconds since the epoch (time in seconds since Jan 1, 1970)	True
serviceName	the name of the Groundwork service	True
value	The performance data value to be plotted	True
warning	The Warning level threshold for this service	False
critical	The Critical level threshold for this service	False

16.1.4 XML POST Data Example

```
<perfDataList>
   <perfData
        appType='OS'
        label='CPU Utilized'
        serverName='localhost'
        serverTime='1397512737'
        serviceName='vm.cpu_util'
        value='65'
        critical='95'
        warning='85'
    />
</perfDataList>
```

See Appendix A for examples of usage with Curl
More Post Data Examples: XML - JSON

16.1.5 HTTP Status Codes

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

16.1.6 Example Response

In this example, we use the POST data above. Two Perf Data records were successfully created.

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<results successful="2" failed="0" entityType="PerfData"
         operation="Update" warning="0" count="2">
  <result>
    <entity>OS - localhost:vm.cpu_util</entity>
    <message>OK</message>
    <status>success</status>
  </result>
  <result>
    <entity>OS - localhost:host.free_ram_mb</entity>
    <message>OK</message>
    <status>success</status>
  </result>
</results>
```

16.2 Get Time Series Performance Data

Gets proxied time series from underlying performance data store. Data returned includes host/service values and thresholds for a specific time range. This service is primarily intended to drive graphing in the Status viewer and thus is not a generalized graph query endpoint. It's intended to return raw data posted to the performance data endpoint, (see 16.1 above). Outside of down sampling to return data in the specified interval, aggregation or other transformations cannot be performed. Returned performance data captures query parameters in a wrapping XML `<perfDataTimeSeries>` element or JSON object with a `perfDataTimeSeriesValues` array member.

> ⚠ Direct retrieval from RRD files via this API is not supported and will result in no results. However, if the underlying performance data store is configured to be RRD, then retrieval of performance data is possible if performance data is also being stored in Foundation by way of configuring `/usr/local/groundwork/config/perfdata.properties`. See the `send_perf_data` property in the `<foundation>` section for more details. This is not generally recommended unless you are using BIRT Performance (EPR) reports.

16.2.1 Method: GET Performance Data

GET /api/perfdata?(query parameters)

16.2.2 HTTP Query and Path Parameters

Field	Type	Description	Required
serverName	Query	Host name primary key	yes
serviceName	Query	Service name primary key	yes
startTime	Query	Start of time series, (millis)	no
endTime	Query	End of time series, (millis), defaults to current time	no
interval	Query	Interval time series down sampled to, (millis)	no
appType	Query	Service application type	no

16.2.3 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

16.2.4 Examples

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

1. *time series data for a service*
   GET
   /api/perfdata?serverName=loadtest-vm-0&serviceName=loadtest-vm-metric-0&startTime=1443160800000&endTime=1443207330000&in

16.2.5 HTTP Status Codes

Code	Description
200	Time series data returned
401	Authentication/authorization error occurred
404	No time series data returned for service in time range
500	An internal server error occurred while returning time series data

16.2.6 Example Time Series Data

Here is an XML example of returned time series performance data. Value types returned can include the `value`, `thold-c`, (critical threshold), and `thold-w`, (warning threshold).

XML query results are always wrapped in an `<perfDataTimeSeries>` collection element, with one or more `<perfDataTimeSeriesValue>` subelements.

```
<perfDataTimeSeries appType="VEMA" serverName="loadtest-vm-0" serviceName="loadtest-vm-metric-0"
startTime="1443208066000" endTime="1443208366000" interval="30000">
    <perfDataTimeSeriesValue valueType="value" timestamp="1443208110000" value="36.0"/>
    <perfDataTimeSeriesValue valueType="value" timestamp="1443208170000" value="27.0"/>
    <perfDataTimeSeriesValue valueType="value" timestamp="1443208230000" value="20.0"/>
    <perfDataTimeSeriesValue valueType="value" timestamp="1443208290000" value="13.0"/>
    <perfDataTimeSeriesValue valueType="thold-c" timestamp="1443208110000" value="100.0"/>
    <perfDataTimeSeriesValue valueType="thold-c" timestamp="1443208170000" value="100.0"/>
    <perfDataTimeSeriesValue valueType="thold-c" timestamp="1443208230000" value="100.0"/>
    <perfDataTimeSeriesValue valueType="thold-c" timestamp="1443208290000" value="100.0"/>
    <perfDataTimeSeriesValue valueType="thold-w" timestamp="1443208110000" value="90.0"/>
    <perfDataTimeSeriesValue valueType="thold-w" timestamp="1443208170000" value="90.0"/>
    <perfDataTimeSeriesValue valueType="thold-w" timestamp="1443208230000" value="90.0"/>
    <perfDataTimeSeriesValue valueType="thold-w" timestamp="1443208290000" value="90.0"/>
</perfDataTimeSeries>
```

Here is a JSON example of returned time series performance data.

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

```
{
  "appType" : "VEMA",
  "serverName" : "loadtest-vm-0",
  "serviceName" : "loadtest-vm-metric-0",
  "startTime" : 1443208066000,
  "endTime" : 1443208366000,
  "interval" : 30000,
  "perfDataTimeSeriesValues" : [ {
    "valueType" : "value",
    "timestamp" : 1443208110000,
    "value" : 36.0
  }, {
    "valueType" : "value",
    "timestamp" : 1443208170000,
    "value" : 27.0
  }, {
    "valueType" : "value",
    "timestamp" : 1443208230000,
    "value" : 20.0
  }, {
    "valueType" : "value",
    "timestamp" : 1443208290000,
    "value" : 13.0
  }, {
    "valueType" : "thold-c",
    "timestamp" : 1443208110000,
    "value" : 100.0
  }, {
    "valueType" : "thold-c",
    "timestamp" : 1443208170000,
    "value" : 100.0
  }, {
    "valueType" : "thold-c",
    "timestamp" : 1443208230000,
    "value" : 100.0
  }, {
    "valueType" : "thold-c",
    "timestamp" : 1443208290000,
    "value" : 100.0
  }, {
    "valueType" : "thold-w",
    "timestamp" : 1443208110000,
    "value" : 90.0
  }, {
    "valueType" : "thold-w",
    "timestamp" : 1443208170000,
    "value" : 90.0
  }, {
    "valueType" : "thold-w",
    "timestamp" : 1443208230000,
    "value" : 90.0
  }, {
    "valueType" : "thold-w",
    "timestamp" : 1443208290000,
    "value" : 90.0
  } ]
}
```