LMIngestPy

Release 0.0.1

Logicmonitor

Jan 27, 2021

CONTENTS:

This Python Library for ingesting the metrics, logs into the LogicMonitor Platform

CHAPTER

ONE

PUSHMETRICS - METRICS INGESTION

1.1 Overview

LogicMonitor's Push Metrics feature allows you to send metrics directly to the LogicMonitor platform via a dedicated API, removing the need to route the data through a LogicMonitor Collector. Once ingested, these metrics are presented alongside all other metrics gathered via LogicMonitor, providing a single pane of glass for metric monitoring and alerting.

More details are available on support site

1.2 Version

- API version: 0.0.1
- Package version: 0.0.1

1.3 Requirements.

Python 2.7 and 3.4+

1.4 Installation

1.4.1 pip install

If the python package ishosted on Github, you can install directly from Github

```
pip install git+ssh://git@stash.logicmonitor.com:7999/dev/lmingestpy.git
```

(you may need to run `pip` with root permission:

`sudo pip install git+ssh://git@stash.logicmonitor.com:7999/dev/lmingestpy.git`)

Then import the package:

```
import lmingest
```

1.4.2 Setuptools

Install via Setuptools.

```
python setup.py install --user
```

or `sudo python setup.py install` to install the package for all users

Then import the package:

```
import lmingest
```

1.5 Getting Started

Please follow the *installation procedure <#Installation>* and then run the following:

```
from __future__ import print_function
import time
import random
import lmingest

from lmingest.api.lm_metrics import MetricsApi
from lmingest.models.lm_datapoint import LMDataPoint
from lmingest.models.lm_datasource import LMDataSource
from lmingest.models.lm_datasource_instance import LMDataSourceInstance
from lmingest.models.lm_resource import LMResource

# Configure API key authorization: LMv1
configuration = lmingest.Configuration(company = 'YOUR_COMPANY', authentication={ 'id
↪': 'YOUR_ACCESS_ID', 'key' : 'YOUR_ACCESS_KEY'})

# create an instance of the API class
metric_api = MetricsApi(lmingest.ApiClient(configuration), interval=20, batch = True)
resource = LMResource(ids={"system.hostname": "SampleDevice"}, create=True, name=
↪"SampleDevice", properties={'some.sdk': 'true'})
ds = LMDataSource(name="DSName")
instance = LMDataSourceInstance(name="instance")
dp = LMDataPoint(name="dataPoint")

while True:
  values = { time.time() : random.randint() }
  metric_api.SendMetrics(resource=resource,
                         datasource=ds,
                         instance=instance,
                         datapoint=dp,
                         values=values)
  time.sleep(10)
```

1.6 Documentation for API Endpoints

All URIs are relative to *https://.logicmonitor.com/rest*

1.6.1 MetricsAPI

Metrics API client: It formats and submit REST API calls to LogicMonitor.

class lmingest.api.lm_metrics.**MetricsApi**(*api_client*, *batch=True*, *interval=30*, *response_callback=None*)

This API client is for ingesting the metrics in LogicMonitor and updating the properties of the resource or instance.

Parameters

- **api_client** (ApiClient) – The RAW HTTP REST client.
- **batch** (*bool*) – Enable the batching support.
- **interval** (*int*) – Batching flush interval. If batching is enabled then after that second we will flush the data to REST endpoint.
- **response_callback** (LMResonseInterface) – Callback for response handling.

classmethod send_metrics(***kwargs*)

This send_metrics method is used to send the metrics to rest endpoint.

Parameters

- **resource** (*lmingest.models.lm_resource.LMResource*) – The Resource object.
- **datasource** (LMDataSource) – The datasource object.
- **instance** (LMDataSourceInstance) – The instance object.
- **datapoint** (LMDataPoint) – The datapoint object.
- **values** (*dict*) – The values dictionary.

Returns If in *MetricsApi* batching is enabled then None Otherwise the REST response will be return.

update_instance_property(*resource_ids*, *datasource*, *instancename*, *instance_properties*, *patch=True*)

This update_resource_property method is used to update the property of the resource.

Parameters

- **resource_ids** (*dict*) – The Resource ids.
- **datasource** (*str*) – The datasource name.
- **instancename** (*str*) – The instance name.
- **instance_properties** (*dict*) – The properties which you want to add/update.
- **patch** (*bool*) – PATCH or PUT request.

Returns REST response will be return.

update_resource_property(*resource_ids*, *resource_properties*, *patch=True*)

This update_resource_property method is used to update the property of the resource.

Parameters

- **resource_ids** (*dict*) – The Resource ids.
- **resource_properties** (*dict*) – The properties which you want to add/update.
- **patch** (*bool*) – PATCH or PUT request.

Returns REST response will be return.

1.7 Documentation For Models & Configuration

1.7.1 Configuration

class lmingest.configuration.**Configuration**(***kwargs*)

This model is used to defining the configuration.

Parameters

- **company** (*str*) – The account name.
- **authentication** (*dict*) – LogicMonitor supports verious type of the authentication. This variable will be used to specify the authentication key.

```
>>> conf = lmingest.Configuration(company="ACCOUNT_NAME",   authentication={'id':
↪'API_ACCESS_ID', 'key': 'API_ACCESS_KEY', 'type' : 'LMv1'})
```

property async_req

The async request.

Parameters **value** – enable async request string.

Type bool

property debug

Debug status

Parameters **value** – The debug status, True or False.

Type bool

property logger_file

The logger file.

If the logger_file is None, then add stream handler and remove file handler. Otherwise, add file handler and remove stream handler.

Parameters **value** – The logger_file path.

Type str

property logger_format

The logger format.

The logger_formatter will be updated when sets logger_format.

Parameters **value** – The format string.

Type str

to_debug_report()

Gets the essential information for debugging.

Returns The report for debugging.

1.7.2 LMResource

class lmingest.models.lm_resource.**LMResource**(*ids*, *name*, *description=None*, *properties=None*, *create=False*)

This model is used to define the resource.

Parameters

- **ids** (*dict*) – An array of existing resource properties that will be used to identify the resource. See Managing Resources that Ingest Push Metrics for information on the types of properties that can be used. If no resource is matched and the create parameter is set to TRUE, a new resource is created with these specified resource IDs set on it. If the system.displayname and/or system.hostname property is included as resource IDs, they will be used as host name and display name respectively in the resulting resource.
- **name** (*str*) – Resource unique name. Only considered when creating a new resource.
- **properties** (*dict*) – New properties for resource. Updates to existing resource properties are not considered. Depending on the property name, we will convert these properties into system, auto, or custom properties.
- **description** (*str*) – Resource description. Only considered when creating a new resource.
- **create** (*bool*) – Do you want to create the resource.

property create

Gets the create flag.

Returns create flag.

Return type bool

property description

Resource description. Only considered when creating a new resource.

Returns The description of this LMResource.

Return type str

property ids

An array of existing resource properties that will be used to identify the resource. See Managing Resources that Ingest Push Metrics for information on the types of properties that can be used. If no resource is matched and the create parameter is set to TRUE, a new resource is created with these specified resource IDs set on it. If the system.displayname and/or system.hostname property is included as resource IDs, they will be used as host name and display name respectively in the resulting resource.

Returns The ids of this LMResource.

Return type dict

property name

Resource unique name. Only considered when creating a new resource.

Returns The name of this LMResource.

Return type str

property properties

New properties for resource. Updates to existing resource properties are not considered. Depending on the property name, we will convert these properties into system, auto, or custom properties.

Returns The properties of this LMResource.

Return type dict

1.7.3 LMDataSource

class lmingest.models.lm_datasource.**LMDataSource**(*name*, *display_name=None*, *group=None*, *id=None*)

This model is used to defining the datasource object.

Parameters

- **name** (*str*) – DataSource unique name. Used to match an existing DataSource. If no existing DataSource matches the name provided here, a new DataSource is created with this name.
- **display_name** (*str*) – DataSource display name. Only considered when creating a new DataSource.
- **group** (*str*) – DataSource group name. Only considered when DataSource does not already belong to a group. Used to organize the DataSource within a DataSource group. If no existing DataSource group matches, a new group is created with this name and the DataSource is organized under the new group.
- **id** (*int*) – DataSource unique ID. Used only to match an existing DataSource. If no existing DataSource matches the provided ID, an error results.

property display_name

DataSource display name. Only considered when creating a new DataSource.

Returns The display_name of this LMDataSource.

Return type str

property group

DataSource group name. Only considered when DataSource does not already belong to a group. Used to organize the DataSource within a DataSource group. If no existing DataSource group matches, a new group is created with this name and the DataSource is organized under the new group.

Returns The group of this LMDataSource.

Return type str

property id

DataSource unique ID. Used only to match an existing DataSource. If no existing DataSource matches the provided ID, an error results.

Returns The id of this LMDataSource. # noqa: E501

Return type int

property name

DataSource unique name. Used to match an existing DataSource. If no existing DataSource matches the name provided here, a new DataSource is created with this name.

Returns The data_source of this LMDataSource.

Return type str

1.7.4 LMDataSourceInstance

class lmingest.models.lm_datasource_instance.**LMDataSourceInstance**(*name*, *description=None*, *display_name=None*, *properties=None*)

This model is used to defining the datasource object.

Parameters

- **name** (*str*) – Instance name. If no existing instance matches, a new instance is created with this name.
- **display_name** (*str*) – Instance display name. Only considered when creating a new instance.
- **properties** (*dict*) – New properties for instance. Updates to existing instance properties are not considered. Depending on the property name, we will convert these properties into system, auto, or custom properties.

property display_name

Instance display name. Only considered when creating a new instance.

Parameters **display_name** – The display_name of this LMDataSourceInstance.

Type str

property name

Instance name. If no existing instance matches, a new instance is created with this name.

Returns The name of this LMDataSourceInstance.

Return type str

property properties

New properties for instance. Updates to existing instance properties are not considered. Depending on the property name, we will convert these properties into system, auto, or custom properties.

Returns The properties of this LMDataSourceInstance.

Return type MapStringString

1.7.5 LMDataPoint

class lmingest.models.lm_datapoint.**LMDataPoint**(*name*, *aggregation_type=None*, *description=None*, *type=None*)

This model is used to defining the datapoint object.

Parameters

- **name** (*str*) – Datapoint name. If no existing datapoint matches for specified DataSource, a new datapoint is created with this name.
- **aggregation_type** (*str*) – The aggregation method, if any, that should be used if data is pushed in sub-minute intervals. Only considered when creating a new datapoint. See the About the Push Metrics REST API section of this guide for more information on datapoint value aggregation intervals.

- **description** (*str*) – Datapoint description. Only considered when creating a new datapoint.
- **type** (*str*) – Metric type as a number in string format. Only considered when creating a new datapoint.

property aggregation_type
The aggregation method, if any, that should be used if data is pushed in sub-minute intervals. Only considered when creating a new datapoint.

Returns The type of this LMDataPoint.

Return type str

property description
Datapoint description. Only considered when creating a new datapoint.

Returns The description of this LMDataPoint.

Return type str

property name
Datapoint name. If no existing datapoint matches for specified DataSource, a new datapoint is created with this name.

Returns The name of this LMDataPoint.

Return type str

property type
Metric type as a number in string format. Only considered when creating a new datapoint.

Returns The aggregation_type of this LMDataPoint.

Return type str

1.7.6 LMResonseInterface

class lmingest.api.lm_response_interface.**LMResonseInterface**
This is the callback interface for handling the response. End user can create his own class using this one to get the response status.

classmethod error_callback(*request*, *response*, *status*, *request_id*, *reason*)
This callback gets invoked for any error or exception from the end REST endpoint.

Parameters

- **request** (*dict*) – The json payload send to REST endpoint.
- **response** (*dict*) – Response received from the REST endpoint.
- **status** (*int*) – HTTP status code.
- **request_id** (*str*) – Unique request id generated by Rest endpoint.
- **reason** (*str*) – The reason for error.

classmethod success_callback(*request*, *response*, *status*, *request_id*)
This callback gets invoked for successful response from the end REST endpoint.

Parameters

- **request** (*dict*) – The json payload send to REST endpoint.
- **response** (*dict*) – Response received from the REST endpoint.

- **status** (*int*) – HTTP status code.
- **request_id** (*str*) – Unique request id generated by Rest endpoint.

1.8 TODO

- [X] Exception Handling, passing any error to end user when ever he makes a Send request for that resource. e.g. SendMetrics is invoked against the resources which are not present
- [X] Supporting the single request
- [X] Validation all the models. e.g. no specical chars allowed in the resource name, length restriction. . . etc
- [X] Property Updation API
- [] Send* call using the unique name
- [] Code commenting for code documentation
- [] Any other authentication support
- [] version/Compression support in send* call
- [] Test cases and sample program.

PYTHON MODULE INDEX

l

INDEX

P

S

T

U