

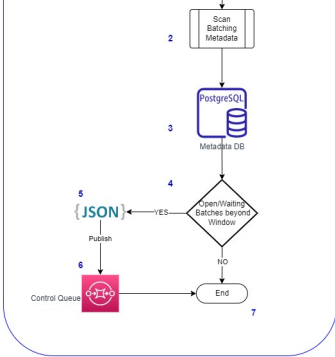
# Batch Monitoring Services

version

Vigenesh Raj

September 13, 2023





	1
	1
	1
	1
Installing Dependent Modules	1
acuity_de_batchingmonitor	2
acuity_de_batchingmonitor package	2
Subpackages	2
acuity_de_batchingmonitor.common package	2
Submodules	2
acuity_de_batchingmonitor.common.CONSTANTS module	2
acuity_de_batchingmonitor.common.connect_pg module	2
acuity_de_batchingmonitor.common.gen_sql module	2
acuity_de_batchingmonitor.common.json_validator module	2
acuity_de_batchingmonitor.common.log4j_logger module	2
acuity_de_batchingmonitor.common.publish_sqs module	3
Module contents	3
acuity_de_batchingmonitor.monitor package	3
Submodules	3
acuity_de_batchingmonitor.monitor.monitor_main module	3
Module contents	4
Module contents	4
Indices and tables	4
Index	5
Python Module Index	7



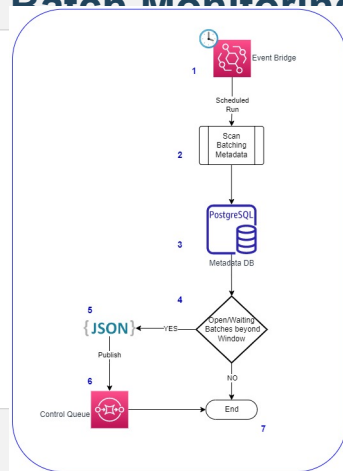
# Welcome to Batch Monitoring Services's documentation!

The Batch Monitor monitors the batch at the defined frequency and captures all the open and delayed branches. It will prepare the respective json message and publish it to the Control service queue.

## Documentation

The Batch Monitor monitors the batch at the defined frequency and captures all the open and delayed branches. It will prepare the respective json message and publish it to the Control service queue.

## Batch Monitoring Flow



“Batch Monitoring” function at the defined frequency.

ing transaction metadata tables.

nd Delayed batches beyond the window end date, from transaction tables.

elayed batches. If no End the process.

yed bactches, prepare respective JSON message.

service queue.

## Installing Dependent Modules

If you're using a recent version of Debian or Ubuntu Linux, you can install with the system package manager:

```
$ apt-get install python-module`
```

package name is published through PyPi, so if you can't install it with the system packager, you can install it with `easy_install` or `pip`. The package name is `packagename`, and the same package works on Python 2 and Python 3.

```
$ easy_install packagename
```

```
$ pip install packagename
```

To install necessary packages:

```
$ pip install uuid
```

```
$ pip install unittest
```

```
$ pip install pycpg2
```

```
$ pip install boto3
```

```
$ pip install botocore
```

```
$ pip install jsonschema
```

If all else fails, the license for package allows you to package the entire library with your application.

I use Python 2.7 and Python 3.2 to develop package, but it should work with other recent versions.

## acuity\_de\_batchingmonitor

### acuity\_de\_batchingmonitor package

#### Subpackages

#### acuity\_de\_batchingmonitor.common package

#### Submodules

#### acuity\_de\_batchingmonitor.common.CONSTANTS module

#### acuity\_de\_batchingmonitor.common.connect\_pg module

```
class acuity_de_batchingmonitor.common.connect_pg.connect_pg
  Bases: object

  commit_pg_txn ()

  get_conn_params ()
```

#### acuity\_de\_batchingmonitor.common.gen\_sql module

#### acuity\_de\_batchingmonitor.common.json\_validator module

acuity\_de\_batchingmonitor.common.json\_validator.**validate\_json**(schema, json\_dict)  
This method will validate an instance under a given schema.

**Parameters:**

- **schema** (*str*) – The schema to validate with.
- **json\_dict** (*dict*) – The instance to validate.

**Returns:** The Validation boolean.

**Return type:** boolean

#### acuity\_de\_batchingmonitor.common.log4j\_logger module

This module contains a class that wraps the log4j object instantiated by the active SparkContext, enabling Log4j logging for PySpark using.

```
class acuity_de_batchingmonitor.common.log4j_logger.Log4j (spark)
  Bases: object
```

**debug** (*message*)

It prints messages with the level (Level.DEBUG).

**Parameters:** **message** (*obj*) – The message object.

**Returns:** None.

**error** (*message*)

It prints messages with the level (Level.ERROR).

**Parameters:** **message** (*obj*) – The message object.

**Returns:** None.

**info** (message)

It prints messages with the level (Level.INFO).

**Parameters:** **message** (*obj*) – The message object.

**Returns:** None.

**warn** (message)

It prints messages with the level (Level.WARN).

**Parameters:** **message** (*obj*) – The message object.

**Returns:** None.

**acuity\_de\_batchingmonitor.commonspublish\_sqs module**

**class** acuity\_de\_batchingmonitor.commonspublish\_sqs.**publish\_sqs**

Bases: **object**

**gen\_msg** (meta\_dict: dict, msg\_dict: dict, exceptionType: str, evtMsg: str)

This method will Create SQS client

**Parameters:**

- **queue\_name** (*str*) – The Queue Name.
- **pub\_msg** (*dict*) – The Public Message.

**Returns:** The response message.

**Return type:** json

**pub\_sqs** (pub\_msg: dict)

This method will Create SQS client

**Parameters:**

- **queue\_name** (*str*) – The Queue Name.
- **pub\_msg** (*dict*) – The Public Message.

**Returns:** The response message.

**Return type:** json

**Module contents****acuity\_de\_batchingmonitor.monitor package****Submodules****acuity\_de\_batchingmonitor.monitor.monitor\_main module**

acuity\_de\_batchingmonitor.monitor.monitor\_main.**b\_mon** ()

This method will be used to create public directory.

**Parameters:**

- **ctrl\_pub\_msg** (*str*) – Event Message.
- **config\_nm** (*str*) – Configuration Name.
- **extract\_dt** (*str*) – Extraxt Date.
- **trgt\_obj\_nm** (*str*) – Target Object Name.

**Returns:** The JSON Response.

**Return type:** dict

acuity\_de\_batchingmonitor.monitor.monitor\_main.**create\_pub\_dict** (ctrl\_pub\_msg: str, config\_nm: str, extract\_dt: str, trgt\_obj\_nm: str)

This method will be used to create public directory.

**Parameters:**

- **ctrl\_pub\_msg** (*str*) – Event Message.
- **config\_nm** (*str*) – Configuration Name.
- **extract\_dt** (*str*) – Extraxt Date.
- **trgt\_obj\_nm** (*str*) – Target Object Name.

**Returns:** The JSON Response.

**Return type:** dict

**Module contents**

**Module contents**

## Indices and tables

- **genindex**
- **modindex**
- **search**



# Index

## A

**acuity\_de\_batchingmonitor**

module

**acuity\_de\_batchingmonitor.common**s

module

**acuity\_de\_batchingmonitor.common**s.connect\_pg

module

**acuity\_de\_batchingmonitor.common**s.CONSTANTS

module

**acuity\_de\_batchingmonitor.common**s.gen\_sql

module

**acuity\_de\_batchingmonitor.common**s.json\_validator

module

**acuity\_de\_batchingmonitor.common**s.log4j\_logger

module

**acuity\_de\_batchingmonitor.common**s.publish\_sqs

module

**acuity\_de\_batchingmonitor.monitor**

module

**acuity\_de\_batchingmonitor.monitor**.monitor\_main

module

## B

b\_mon() (in module  
acuity\_de\_batchingmonitor.monitor.monitor\_main)

## C

commit\_pg\_txn() (acuity\_de\_batchingmonitor.common  
s.connect\_pg.connect\_pg method)

connect\_pg (class in  
acuity\_de\_batchingmonitor.common.s.connect\_pg)

create\_pub\_dict() (in module  
acuity\_de\_batchingmonitor.monitor.monitor\_main)

## D

debug() (acuity\_de\_batchingmonitor.common.s.log4j\_lo  
gger.Log4j method)

## E

error() (acuity\_de\_batchingmonitor.common.s.log4j\_lo  
gger.Log4j method)

## G

gen\_msg() (acuity\_de\_batchingmonitor.common.s.publi  
sh\_sqs.publish\_sqs method)

get\_conn\_params() (acuity\_de\_batchingmonitor.comm  
ons.connect\_pg.connect\_pg method)

## I

info() (acuity\_de\_batchingmonitor.common.s.log4j\_logg  
er.Log4j method)

## L

Log4j (class in  
acuity\_de\_batchingmonitor.common.s.log4j\_logger)

## M

**module**

acuity\_de\_batchingmonitor

acuity\_de\_batchingmonitor.common

acuity\_de\_batchingmonitor.common.s.connect\_pg

acuity\_de\_batchingmonitor.common.s.CONSTANTS

acuity\_de\_batchingmonitor.common.s.gen\_sql

acuity\_de\_batchingmonitor.common.s.json\_validator

acuity\_de\_batchingmonitor.common.s.log4j\_logger

acuity\_de\_batchingmonitor.common.s.publish\_sqs

acuity\_de\_batchingmonitor.monitor

acuity\_de\_batchingmonitor.monitor.monitor\_main

## P

pub\_sqs() (acuity\_de\_batchingmonitor.common.s.publis  
h\_sqs.publish\_sqs method)

publish\_sqs (class in  
acuity\_de\_batchingmonitor.common.s.publish\_sqs)

## V

validate\_json() (in module  
acuity\_de\_batchingmonitor.common.s.json\_validator)

## W

warn() (acuity\_de\_batchingmonitor.common.s.log4j\_lo  
gger.Log4j method)



# Python Module Index

## a

- [acuity\\_de\\_batchingmonitor](#)
- [acuity\\_de\\_batchingmonitor.common](#)s
- [acuity\\_de\\_batchingmonitor.common](#)s.connect\_pg
- [acuity\\_de\\_batchingmonitor.common](#)s.CONSTANTS
- [acuity\\_de\\_batchingmonitor.common](#)s.gen\_sql
- [acuity\\_de\\_batchingmonitor.common](#)s.json\_validator
- [acuity\\_de\\_batchingmonitor.common](#)s.log4j\_logger
- [acuity\\_de\\_batchingmonitor.common](#)s.publish\_sqs
- [acuity\\_de\\_batchingmonitor.monitor](#)
- [acuity\\_de\\_batchingmonitor.monitor.monitor\\_main](#)