decisionengine

Release 1.1.1

Fermi Research Alliance, LLC.

Feb 26, 2021

CONTENTS

The decisionengine_modules package contains provider specific implementations of base classes provided by the decision engine framework.

CHAPTER

ONE

RELEASE NOTES

1.1 Release 1.5.0

In this release:

- New AWS bill calculator
- Bug fixes

1.1.1 Issues fixed in this release

- 271 : Change configuration so the bill-calculator-hep-mapsacosta library doesn't write stuff to stdout (5410c46)
- 283 : Change DEAccountConstants.py to allow more variables through (7a65032)
- 293 : Unit test tests/test_job_q.py is failing (f5006fb)
- 263 : Python3 encodes hex strings differently, resulting in malformed htcondor classads (c16f0)
- 289 : Travis CI fails after decisionenigine transitioned to pybind11 (2bec1da)
- 281 : New exception in NERSC newt get_usage (398dc42)

1.1.2 Full list of commits since version 1.4.2

5410c46 : new bill calculator package (#303)

7a65032 : Resolving issue 283 (#301)

1a2c3e4 : Fix syntax errors and pylint bits (#300)

976c4a4 : Minor linting of actions (#297)

c43f311 : Drop redundant action, pep8speaks is already doing this (#296)

10691e4 : Add support to use custom docker layer as base of the docker image (#295)

f5006fb : fix for unit test test_job_q issue 293 (#294)

0b588ef : Update workflow per changes at github (#292)

7841d19 : Updated Jenkins configuration documentation (#291)

c08ffce : edited release notes

c16f0ac : Another fix to malformed strings (#288)

2bec1da : Fix #289 where TravisCI is missing changes from https://github.com/HEPCloud/decisionengine/commit/ea7ade57ca825afb0afbfa5ebd49820b16786d8b (#290)

8647bb2 : Update to current python test matrix (#287)

234a78e : update release notes

398dc42 : decisionengine_modules/NERSC/util: Added more 5xx codes to retry (#285)

d2df861 : force docker pull when building the docker container to make sure to use an updated base layer (#280)

9a78d22 : added 1.4.0 release notes

5d3b7ec : Updating Dockerfile for unittest (#279)

e16531e : Provide default for util function. (#278)

02f1738 : set RPM version number to 1.5.0rc and drop py3 suffix (#277)

664b9fe : Add timeout support for unit test on Jenkins for DE modules repo (#274)

e392a3c : Added initial unit test for glide_frontend_element.py (#269)

2f0069f : Updated DE modules Jenkins documentation

1.2 Release 1.4.2

In this release:

- Bug fixes to 1.4.1 release

1.2.1 Issues fixed in this release

- 282 : Decision Engine classad encoding problems make multiple exceptions in the factory (c16f0ac)

1.2.2 Full list of commits since version 1.4.1

c16f0ac : Another fix to malformed strings (#288)

1.3 Release 1.4.1

In this release:

- Bug fixes to 1.4.0 release

1.3.1 Issues fixed in this release

- 281 : New exception in NERSC newt get_usage (4d1301e)

1.3.2 Full list of commits since version 1.4.0

4d1301e : decisionengine_modules/NERSC/util: Added more 5xx codes to retry (#285)

1.4 Release 1.4.0

In this release:

- Works with python3 glideinWMS package
- Uses new AWS bill calculator
- Improved Jenkins CI pipeline

1.4.1 Issues fixes in this release

- 59 : Replace AWS Billing and GCE Billing modules with revamped bill Calculator code (71f25ea)
- 246 : Modify NERSCAllocation main() so that it can be run standalone (11ce772)
- 263 : Python3 encodes hex strings differently, resulting in malformed htcondor classads (13a24e4)
- 255 : Add option to config of condor/sources/job_q.py and slots.py (6f65af4)

1.4.2 Full list of commits since version 1.3.0

42f62d3 : remove py3 from package name

6e56c4d : bump version number

6433dba : Add support to test any branch in Jenkins for DE modules repo (#276)

60db1b3 : Accommodate breaking changes to framework's configuration system. (#273)

13a24e4 : Fix263 (#272)

a866e47 : Correct 'file_cache is unavailable when using oauth2client >= 4.0.0' (#267)

2c82064 : Fix requirements.txt for #266 (#268)

6f65af4 : A more generic solution for the issue 255 (#265)

df08cb4 : Speed up tests. (#264)

057e578 : Update Jenkins pipeline to properly test closing PR (#262)

a28f272 : Adding Jenkins pipeline documentation (#261)

71f25ea : T59 (#258)

90b6580 : Updating Jenkins pipeline configuration (#260)

b4bd759 : dealing with 2 columnsindividually (#257)

11ce772 : decisionengine_modules/NERSC: Can run NerscAllocationInfo.py without (#256)

4a70465 : doc: add release notes

1.5 Release 1.3.0

In this release:

- Introduced Jsonnet based configuration system
- Improved error handling

1.5.1 Full list of commits since version 1.2.0

e8c8e24 : Update to split a tuple key into two separate keys, this allows python config file to be JSON compliant config file (#254)

fbb5d28 : add jsonnet dependency in build/scripts/utils.sh for unit tests (#252)

95a0a32 : decisionengine_modules/glideinwms: Added retry to factory queries (#250)

77f894a : Aws (#249)

3be5be1 : Set theme jekyll-theme-slate

a73e7f8 : Doc (#247)

f8aa96d : Adding editor config file, see https://editorconfig.org/ (#248)

1ab1720 : Issue no. 237 (#245)

5edc8f9 : Issue 227 (#243)

e6047f2 : GCE channel goes OFFLINE on a timeout (#242)

63e3627 : Aws billing (#241)

c3626b9 : 1.1.3/fix (#239)

ae7c5a6 : Jenkins pipeline improvements (#236)

310befd : T198 (#235)

a65886d : Fix import as reported in : https://github.com/HEPCloud/decisionengin… (#232)

93711cc : Run coveralls even if tests fail (#229)

03d763a : Jenkins pipeline improvements (#230)

f48d30f : Fix/223 (#228)

c8aa262 : github ticket 199 (#222)

0323bda : Address : https://github.com/HEPCloud/decisionengine_modules/issues/224 (#226)

62e4df6 : Add support to run CI on Jenkins (#221)

5ab1541 : bump master version to 1.2.0 (for now) (#219)

bc19c65 : decisionengine_modules/NERSC: Added retry loop for NERSC API Calls (#220)

41a50de : Sync up pep8speaks and run_pylint.sh with decisionengine settings (#218)

db4634f : silence pylint error (#217)

1b95141 : Fix whitespace around operator error

746ea38 : ignore W503

8a8b5f4 : remove unused variable

a6668bf : fix PEP8 warnings

13773ee : address pep8 warnings

6bea4ca : silence pylint error

f589895 : Pass sort=True parameter to fix future warning (#215)

a1d0507 : fixing pep8 warning

a10bd17 : debugging one import error

ec501ad : make coveralls.io links work

deab1a7 : T201 (#204)

69f2645 : Add coveragerc

6d8a5f5 : decisionengine_modules/NERSC: Make Nersc API call backward-compatible with old config (#196)

a7e0af9 : Only run Github Actions for python3.6 (#24)

1.6 Release 1.2.0

In this release:

- Switched to python3
- NERSC newt interface adapted to changed NERSC RESTful API
- Improved unit tests
- Added Jenkins CI pipeline

1.7 Patches:

30d928b : clone version 1.2.0 of decisionengine

ae7c5a6 : Jenkins pipeline improvements (#236)

310befd : T198 (#235)

a65886d : Fix import as reported in : https://github.com/HEPCloud/decisionengin… (#232)

93711cc : Run coveralls even if tests fail (#229)

03d763a : Jenkins pipeline improvements (#230)

f48d30f : Fix/223 (#228)

c8aa262 : github ticket 199 (#222)

0323bda : Address : https://github.com/HEPCloud/decisionengine_modules/issues/224 (#226)

62e4df6 : Add support to run CI on Jenkins (#221)

5ab1541 : bump master version to 1.2.0 (for now) (#219)

bc19c65 : decisionengine_modules/NERSC: Added retry loop for NERSC API Calls (#220)

41a50de : Sync up pep8speaks and run_pylint.sh with decisionengine settings (#218) db4634f : silence pylint error (#217)

1b95141 : Fix whitespace around operator error

746ea38 : ignore W503

8a8b5f4 : remove unused variable

a6668bf : fix PEP8 warnings

13773ee : address pep8 warnings

6bea4ca : silence pylint error

f589895 : Pass sort=True parameter to fix future warning (#215)

a1d0507 : fixing pep8 warning

a10bd17 : debugging one import error

ec501ad : make coveralls.io links work

deab1a7 : T201 (#204)

69f2645 : Add coveragerc

6d8a5f5 : decisionengine_modules/NERSC: Make Nersc API call backward-compatible with old config (#196)

a7e0af9 : Only run Github Actions for python3.6 (#24)

CHAPTER

TWO

DEVELOPER DOCUMENTATION

First command cd is just to make sure that you end up in a directory that will contain two subdirectory decisionengine and decisionengine_modules. Of course this can be done in any directory, not necessarily home directory.

2.1 Decisionengine_modules

2.1.1 Prerequisites:

In Addition to above installed packages

```
yum install condor
pip3 install htcondor boto boto3 google_auth google-api-python-client gcs-oauth2-boto-
↪plugin
```

2.1.2 Test

```
cd

git clone https://github.com/HEPCloud/decisionengine_modules
python3 -m pytest decisionengine_modules
```

Current status:

```
[root@fermicloud371 tmp]# python3 -m pytest decisionengine_modules
==================================== test session starts␣
↪=====================================
platform linux -- Python 3.6.8, pytest-5.3.5, py-1.8.1, pluggy-0.13.1
rootdir: /root/junjk
collected 85 items

decisionengine_modules/AWS/tests/test_AWSInstancePerformance.py ..                  ␣
↪ [  2%]
decisionengine_modules/AWS/tests/test_AWSJobLimits.py ..                            ␣
↪ [  4%]
decisionengine_modules/AWS/tests/test_AWSOccupancyWithSourceProxy.py ..             ␣
↪ [  7%]
decisionengine_modules/AWS/tests/test_AWSSpotPriceWithSourceProxy.py ..             ␣
↪ [  9%]
decisionengine_modules/AWS/tests/test_AWS_figure_of_merit_publisher.py ..           ␣
↪ [ 11%]
```

(continues on next page)

(continued from previous page)

```
decisionengine_modules/AWS/tests/test_AWS_price_performance_publisher.py .. ␣
↪ [ 14%]
decisionengine_modules/AWS/tests/test_FigureOfMerit.py ... ␣
↪ [ 17%]
decisionengine_modules/tests/test_AwsBurnRate.py .. ␣
↪ [ 20%]
decisionengine_modules/tests/test_GCEBillingInfo.py .. ␣
↪ [ 22%]
decisionengine_modules/tests/test_GCEFigureOfMerit_publisher.py .. ␣
↪ [ 24%]
decisionengine_modules/tests/test_GCEInstancePerformanceInfo.py .. ␣
↪ [ 27%]
decisionengine_modules/tests/test_GCEPricePerformance_publisher.py .. ␣
↪ [ 29%]
decisionengine_modules/tests/test_GCEResourceLimits.py .. ␣
↪ [ 31%]
decisionengine_modules/tests/test_GceBurnRate.py .. ␣
↪ [ 34%]
decisionengine_modules/tests/test_GceFigureOfMerit.py .. ␣
↪ [ 36%]
decisionengine_modules/tests/test_GceOccupancy.py .. ␣
↪ [ 38%]
decisionengine_modules/tests/test_NerscAllocationInfo.py .. ␣
↪ [ 41%]
decisionengine_modules/tests/test_NerscFigureOfMerit.py .. ␣
↪ [ 43%]
decisionengine_modules/tests/test_NerscFigureOfMerit_publisher.py .. ␣
↪ [ 45%]
decisionengine_modules/tests/test_NerscInstancePerformance.py .. ␣
↪ [ 48%]
decisionengine_modules/tests/test_NerscJobInfo.py .. ␣
↪ [ 50%]
decisionengine_modules/tests/test_factory_client.py .... ␣
↪ [ 55%]
decisionengine_modules/tests/test_factory_entries.py .... ␣
↪ [ 60%]
decisionengine_modules/tests/test_factory_global.py .... ␣
↪ [ 64%]
decisionengine_modules/tests/test_fomorderplugin.py .... ␣
↪ [ 69%]
decisionengine_modules/tests/test_grid_figure_of_merit.py . ␣
↪ [ 70%]
decisionengine_modules/tests/test_htcondor_query.py .... ␣
↪ [ 75%]
decisionengine_modules/tests/test_job_clustering.py ..... ␣
↪ [ 81%]
decisionengine_modules/tests/test_job_clustering_publisher.py .. ␣
↪ [ 83%]
decisionengine_modules/tests/test_job_q.py ... ␣
↪ [ 87%]
decisionengine_modules/tests/test_slots.py .. ␣
↪ [ 89%]
decisionengine_modules/tests/glideinwms/publishers/test_decisionenginemonitor.py ... ␣
↪ [ 92%]
decisionengine_modules/tests/glideinwms/publishers/test_fe_group_classads.py ... ␣
↪ [ 96%]
decisionengine_modules/tests/glideinwms/publishers/test_glideclientglobal.py ... ␣
↪ [100%]
```

(continues on next page)

(continued from previous page)

```
====================================== warnings summary␣
↪======================================
/usr/local/lib/python3.6/site-packages/boto/plugin.py:40
  /usr/local/lib/python3.6/site-packages/boto/plugin.py:40: DeprecationWarning: the␣
↪imp module is deprecated in favour of importlib; see the module's documentation for␣
↪alternative uses
    import imp

-- Docs: https://docs.pytest.org/en/latest/warnings.html
=============================== 85 passed, 1 warning in 9.73s␣
↪================================
```

CHAPTER

THREE

SOURCE CODE

3.1 Welcome to decisionengine_modules's documentation!

3.1.1 decisionengine_modules package

Subpackages

decisionengine_modules.AWS package

Subpackages

decisionengine_modules.AWS.publishers package

Submodules

decisionengine_modules.AWS.publishers.AWS_burn_rate module

decisionengine_modules.AWS.publishers.AWS_figure_of_merit module

decisionengine_modules.AWS.publishers.AWS_generic_publisher module

decisionengine_modules.AWS.publishers.AWS_price_performance module

Module contents

decisionengine_modules.AWS.sources package

Submodules

decisionengine_modules.AWS.sources.AWSBurnRateSourceProxy module

decisionengine_modules.AWS.sources.AWSInstancePerformance module

decisionengine_modules.AWS.sources.AWSJobLimits module

decisionengine_modules.AWS.sources.AWSOccupancy module

decisionengine_modules.AWS.sources.AWSOccupancyWithSourceProxy module

decisionengine_modules.AWS.sources.AWSSpotPrice module

decisionengine_modules.AWS.sources.AWSSpotPriceWithSourceProxy module

decisionengine_modules.AWS.sources.BillingInfo module

decisionengine_modules.AWS.sources.BillingInfoSourceProxy module

decisionengine_modules.AWS.sources.DEAccountContants module

This data is taken from billing and arranged as dictionary of AccountConstants instead of separate class for each billing class. This way is more covenient to use in programms.

class `decisionengine_modules.AWS.sources.DEAccountContants.`**`AccountConstants`**`(`*constants={}*`)`

Bases: `object`

`info``()`

`decisionengine_modules.AWS.sources.DEAccountContants.`**`load_constants`**`(`*constants_file*`)`

Load constants from file.

Parameters **`constants_file`** (`file`) – configuration file as python dict

Return type *`AccountConstants`*

decisionengine_modules.AWS.sources.FigureOfMeritSourceProxy module

decisionengine_modules.AWS.sources.FinancialParameters module

decisionengine_modules.AWS.sources.occupancy_config_sample module

decisionengine_modules.AWS.sources.spot_price_config_sample module

Module contents

decisionengine_modules.AWS.transforms package

Submodules

decisionengine_modules.AWS.transforms.AwsBurnRate module

decisionengine_modules.AWS.transforms.FigureOfMerit module

Module contents

Submodules

decisionengine_modules.AWS.AWSSpotPrice module

decisionengine_modules.AWS.CloudBudget module

decisionengine_modules.AWS.CloudRequests module

decisionengine_modules.AWS.HPCAllocation module

decisionengine_modules.AWS.HTCondorJobQ module

decisionengine_modules.AWS.LocalCapacity module

decisionengine_modules.AWS.ProvisionerResourceList module

decisionengine_modules.AWS.pandas_scratch_pad module

`decisionengine_modules.AWS.pandas_scratch_pad.`**`load_data_frame`**(*list_of_dicts*)

Module contents

decisionengine_modules.GCE package

Subpackages

decisionengine_modules.GCE.publishers package

Submodules

decisionengine_modules.GCE.publishers.GCEBurnRatePublisher module

decisionengine_modules.GCE.publishers.GCEFigureOfMerit_publisher module

decisionengine_modules.GCE.publishers.GCEPricePerformance_publisher module

Module contents

decisionengine_modules.GCE.sources package

Submodules

decisionengine_modules.GCE.sources.GCEBillingInfo module

decisionengine_modules.GCE.sources.GCEBillingInfoSourceProxy module

decisionengine_modules.GCE.sources.GCEInstancePerformance module

decisionengine_modules.GCE.sources.GCEResourceLimits module

decisionengine_modules.GCE.sources.GceBurnRateSourceProxy module

decisionengine_modules.GCE.sources.GceFigureOfMeritSourceProxy module

decisionengine_modules.GCE.sources.GceOccupancy module

Module contents

decisionengine_modules.GCE.transforms package

Submodules

decisionengine_modules.GCE.transforms.GceBurnRate module

decisionengine_modules.GCE.transforms.GceFigureOfMerit module

Module contents

Module contents

decisionengine_modules.NERSC package

Subpackages

decisionengine_modules.NERSC.publishers package

Submodules

decisionengine_modules.NERSC.publishers.NerscFigureOfMerit_publisher module

Module contents

decisionengine_modules.NERSC.sources package

Submodules

decisionengine_modules.NERSC.sources.NerscAllocationInfo module

decisionengine_modules.NERSC.sources.NerscAllocationInfoSourceProxy module

decisionengine_modules.NERSC.sources.NerscFigureOfMeritSourceProxy module

decisionengine_modules.NERSC.sources.NerscInstancePerformance module

decisionengine_modules.NERSC.sources.NerscJobInfo module

Module contents

decisionengine_modules.NERSC.transforms package

Submodules

decisionengine_modules.NERSC.transforms.CompareNerscFactoryJobs module

decisionengine_modules.NERSC.transforms.CompareNerscUserpoolSlots module

decisionengine_modules.NERSC.transforms.NerscFigureOfMerit module

Module contents

decisionengine_modules.NERSC.util package

Submodules

decisionengine_modules.NERSC.util.newt module

Newt API interface

class `decisionengine_modules.NERSC.util.newt.`**`Newt`**(*password_file*, *newt_base_url=None*, *num_retries=0*, *retry_backoff_factor=0*)

Bases: `object`

`_add_retries_to_session`()

Adds retries to requests Session for requests to NEWT URLs :return: void

`_login`()

Establishes newt Session :return: void

`get_queue`(*system*, *query=None*)

Returns information about queues controlled by optional query name parameter :param system: string system name (like ‘cori’, ‘edison’ ..) :param query: string query (optional) :return: json containing queue information

`get_status`(*system=None*)

Returns system status for a given system (if provided) or all systems :param system: string name of the system (optional) :return: json containing system status

`get_usage`(*username*)

Returns allocation and usage for the given user using NEWT IRIS API :param username: string username :return: json containing allocation and usage information for a given user

Module contents

Module contents

decisionengine_modules.graphite package

Subpackages

decisionengine_modules.graphite.publishers package

Submodules

decisionengine_modules.graphite.publishers.generic_publisher module

Module contents

Module contents

decisionengine_modules.htcondor package

Subpackages

decisionengine_modules.htcondor.publishers package

Submodules

decisionengine_modules.htcondor.publishers.publisher module

Module contents

decisionengine_modules.htcondor.sources package

Submodules

decisionengine_modules.htcondor.sources.job_q module

decisionengine_modules.htcondor.sources.job_q_source_proxy module

decisionengine_modules.htcondor.sources.slots module

decisionengine_modules.htcondor.sources.slots_source_proxy module

decisionengine_modules.htcondor.sources.source module

Module contents

Submodules

decisionengine_modules.htcondor.htcondor_query module

class decisionengine_modules.htcondor.htcondor_query.**CondorQ**(*schedd_name=None*, *pool_name=None*)

Bases: *decisionengine_modules.htcondor.htcondor_query.CondorQuery*

Class to implement condor_q

_abc_impl = <_abc_data object>

fetch(*constraint=None*, *format_list=None*, *condor_config=None*)

Fetch job classads

class decisionengine_modules.htcondor.htcondor_query.**CondorQuery**(*resource_str*, *group_attr*, *pool_name=None*, *env=None*)

Bases: *decisionengine_modules.htcondor.htcondor_query.Query*

Fully implemented class for CondorQuery

_abc_impl = <_abc_data object>

class decisionengine_modules.htcondor.htcondor_query.**CondorStatus**(*subsystem_name=None*, *pool_name=None*, *group_attr=None*)

Bases: *decisionengine_modules.htcondor.htcondor_query.CondorQuery*

Class to implement condor_status

_abc_impl = <_abc_data object>

fetch(*constraint=None*, *format_list=None*, *condor_config=None*)

Fetch resource classads and return a list of evaluated classads

class decisionengine_modules.htcondor.htcondor_query.**Query**

Bases: object

Pure virtual class to have a minimum set of methods defined

_abc_impl = <_abc_data object>

abstract fetch(*constraint=None*, *format_list=None*, *condor_config=None*)

Fetch the classad attributes specified in the format_list matching the constraint

fetch_stored(*constraint_func=None*)

Parameters constraint_func – A boolean function, with only one argument (data el). If None, return all the data.

@rtype: dict @return: Same as fetch(), but limited to constraint_func(el)==True

load(*constraint=None*, *format_list=None*, *condor_config=None*)

Fetch the data and store it in self.stored_data

exception decisionengine_modules.htcondor.htcondor_query.**QueryError**(*err_str*)

Bases: RuntimeError

Thrown when there are exceptions using htcondor python bindings or commands

decisionengine_modules.htcondor.htcondor_query.**apply_constraint**(*data*, *constraint_func*)

Return a subset of data that satisfies constraint_function If constraint_func is None, return back entire data `decisionengine_modules.htcondor.htcondor_query.`**`bindings_friendly_attrs`**(*format_list*)

Convert the format_list into attrs that can be used with python bindings Python bindings should take care of the typing

`decisionengine_modules.htcondor.htcondor_query.`**`bindings_friendly_constraint`**(*constraint*)

Convert the constraint to format that can be used with python bindings

`decisionengine_modules.htcondor.htcondor_query.`**`eval_classad_expr`**(*classads*, *format_list=None*)

Convert the classads into list of classads Guarantees any attribute in format_list to be added with default value None

`decisionengine_modules.htcondor.htcondor_query.`**`list2dict`**(*list_data*, *attr_name*)

Convert a list to a dictionary and group the results based on attributes specified by attr_name

`decisionengine_modules.htcondor.htcondor_query.`**`resource_str_to_py_adtype`**(*resource_str*)

Given the resource string return equivalent classad type

`decisionengine_modules.htcondor.htcondor_query.`**`split_collector_host`**(*collector_host*)

Given the collector_host string extract primary collector_host which is the first collector host in the list and sort the secondary collectors. Return a tuple of primary collector and CSV secondary collector.

Module contents

decisionengine_modules.util package

Submodules

decisionengine_modules.util.figure_of_merit module

Calculate figure of merit

`decisionengine_modules.util.figure_of_merit.`**`figure_of_merit`**(*performance*, *running*, *allowed*, *idle=None*, *max_idle=None*)

decisionengine_modules.util.retry_function module

`decisionengine_modules.util.retry_function.`**`retry_on_error`**(*nretries=1*, *retry_interval=2*, *backoff=True*)

`decisionengine_modules.util.retry_function.`**`retry_wrapper`**(*f*, *nretries=1*, *retry_interval=2*, *backoff=True*)

Retry on error with parameters of how many times (nretries) and interval in seconds (retry_interval). If the function to be decorated is an instance method and the values come from a configuration, then this decorator can retrieve these two parameters from instance attributes with the same name. Otherwise, use the default values or pass new values to the decorator.

decisionengine_modules.util.test_retry_function module

class `decisionengine_modules.util.test_retry_function.`**`Dummy`**(*name='Dummy'*)

Bases: `object`

`_func_failure`(*foo*, *input2=3*)

`_func_success`(*foo*, *input2=3*)

`func_failure`(*foo*, *input2=3*)

`func_success`(*foo*, *input2=3*)

`decisionengine_modules.util.test_retry_function.`**`test_all`**()

decisionengine_modules.util.testutils module

Utils to simplify testing

`decisionengine_modules.util.testutils.`**`compare_dfs`**(*df1*, *df2*)

for some reason df.equals does not work here but if I compare cell by cell it works :type df1: `pd.DataFrame` :arg df1: data frame instance :type df2: `pd.DataFrame` :arg df2: data frame instance :rtype: `bool` - True if equal

`decisionengine_modules.util.testutils.`**`input_from_file`**(*fname*)

`decisionengine_modules.util.testutils.`**`raw_input_from_file`**(*fname*)

Module contents

Submodules

decisionengine_modules.graphite_client module

class `decisionengine_modules.graphite_client.`**`Graphite`**(*host='fifemondata.fnal.gov'*, *pickle_port=2004*)

Bases: `object`

`send_dict`(*namespace*, *data*, *debug_print=True*, *send_data=True*)

send data contained in dictionary as {k: v} to graphite dataset $namespace.k with current timestamp

`decisionengine_modules.graphite_client.`**`sanitize_key`**(*key*)

decisionengine_modules.load_config module

`decisionengine_modules.load_config.`**`load`**(*python_file*, *retries=0*, *timeout=0*)

Load constants from file.

Parameters **`constants_file`** (`file`) – configuration file as python dict

Return type `object`

Module contents

3.2 Indices and tables

- genindex
- modindex
- search

CHAPTER FOUR

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX