Flashflood Documentation

Release 0.14.0

Seiji Matsuoka

Apr 26, 2019

Contents

1 Features 3

2 Contents: 5

3 License 11

Python Module Index 13

Flashflood is a directed acyclic graph (DAG) workflow-based HTTP API server builder. Input data is processed according to the workflow definition (data processing nodes and dataflow edges) and then the result JSON (and other formats) will be broadcasted via HTTP. This offers instant database access API to users to share their results and analysis regardless of their programming langages and environments.

CHAPTER 1

Features

- HTTP API builder
- Directed acyclic graph (DAG) workflow to process database records
- Asynchronus workflow: you can get records processed so far from ongoing tasks
- Multiprocess (parallel) record processing
- Input from python list, SQLite table, text file (CSV-like) and SDFile.
- Chemical structure input (by using Chorus)
- Output to JSON, Excel, SDFile and SQLite

CHAPTER 2

Contents:

2.1 Getting started

2.1.1 Installation

Warning: As Flashflood is still in an experimental status, Public API and request-response format specification will be updated frequently. We highly recommend to clone latest master branch, which may be the most stable version, from GitHub.

PyPI

```
pip3 install flashflood
```

Anaconda

Distributing conda package is work in progress.

```
conda upgrade -n root conda
conda install -n root conda-build

conda skeleton pypi flashflood
conda build flashflood
conda install --use-local flashflood
```

- genindex
- modindex
- search

2.2 Building workflow

```
from tornado.ioloop import IOLoop

from flashflood.core.node import FuncNode
from flashflood.core.workflow import Workflow
from flashflood.node.reader.iterinput import IterInput


def twice(x):
    return x * 2


class TestWorkflow(Workflow):
    def __init__(self):
        super().__init__()
        self.output = Container()
        self.append(IterInput(range(100)))
        self.append(FuncNode(twice))
        self.append(ContainerWriter(output))


if __name__ == '__main__':
    wf = TestWorkflow()
    task = Task(wf)
    IOLoop.current().run_sync(task.execute)
    print(wf.output.records)
```

- genindex
- modindex
- search

2.3 API references

2.3.1 flashflood.core.concurrent

2.3.2 flashflood.core.container

class Container
Bases: object

Data container

class Counter
Bases: object

Count container

class Sampler(*size=5*, *frequency=1*, *deep_copy=False*)
Bases: object

Data sampler

2.3.3 flashflood.core.edge

2.3.4 flashflood.core.jobqueue

2.3.5 flashflood.core.node

2.3.6 flashflood.core.task

2.3.7 flashflood.core.workflow

2.3.8 flashflood.functional

compose(**funcs*)
Function composition

2.3.9 flashflood.interface.sqlite

2.3.10 flashflood.interface.xlsx

2.3.11 flashflood.lod

List of dict (LOD) module

List of dict class

class ListOfDict
Bases: `list`

List of dict utility class

values(*key*)
Returns record values with the given key

filter(*key*, *value*)
Filters by the given key-value pair

find(*key*, *value*)
Returns the record found by the given key-value pair. if not found, return None

add(*rcd*, *key='key'*, *dupkey='replace'*)
Adds a new record.

Duplicate key (dupkey) operations

- replace - replace existing record with the new one (default)
- update - update exisiting record (dict.update)
- skip - add no record when the key is duplicated
- replace or update will be applied for only the first one found

Parameters

- **rcd** (*dict*) – New record to be added
- **key** (*str*) – Record key

- **dupkey** (*str*) – Type of duplicate key operation

reduce(*key='key'*, *dupkey='update'*)
Removes records with duplicated key

Not appropriate for large data - worst case: O(n^2)

Duplicate key (dupkey) operations

- update - update exisiting record (dict.update)
- skip - add no record when the key is duplicated

Parameters

- **key** (*str*) – Record key
- **dupkey** (*str*) – Type of duplicate key operation

unique(*key='key'*)
Aliase of reduce(dupkey="skip")

merge(*rcds*, *key='key'*, *dupkey='replace'*)
Adds list of records

Parameters

- **rcds** (*list*) – List of dict records to be added
- **key** (*str*) – Record key
- **dupkey** (*str*) – Type of duplicate key operation (see *ListOfDict.add*)

join(*rcds*, *key*, *full_join=False*)
Left join records

Parameters

- **rcds** (*list*) – List of dict records to be joined
- **key** (*str*) – Join key
- **full_join** (*bool*) – If true, do full join (join all records regardless of whether key exists)

delete(*key*, *value*)
Removes records which matches the key-value pair

Parameters

- **key** (*str*) – key
- **value** – value

pick(*key*, *value*)
Removes a record which matches the key-value pair from the list and return the record.

If no records are found, this returns None.

Parameters

- **key** (*str*) – key
- **value** – value

LOD = <class 'flashflood.lod.ListOfDict'>
Shorthand of ListOfDict

Non-destructive list of dict functions

valuelist(*lod*, *key*)
Returns list of values which are assigned to the key.

Parameters

- **lod** – *flashflood.lod.ListOfDict*
- **key** – key

Returns list of values

Return type list

filtered(*lod*, *key*, *value*)
Returns filtered records by the key-value pair.

Parameters

- **key** – key
- **value** – value

Returns filtered records

Return type *flashflood.lod.ListOfDict*

2.3.12 flashflood.node.aggregate.first

2.3.13 flashflood.node.aggregate.sum

2.3.14 flashflood.node.aggregate.update

2.3.15 flashflood.node.chem.descriptor

2.3.16 flashflood.node.chem.molecule

2.3.17 flashflood.node.control.filter

2.3.18 flashflood.node.field.concat

2.3.19 flashflood.node.field.extend

2.3.20 flashflood.node.field.number

2.3.21 flashflood.node.field.split

2.3.22 flashflood.node.field.update

2.3.23 flashflood.node.reader.sdfile

2.3.24 flashflood.node.reader.sqlite

2.3.25 flashflood.node.writer.container

2.3.26 flashflood.node.writer.sqlite

CHAPTER 3

License

MIT license

- genindex
- modindex
- search

Python Module Index

f

flashflood.core.container, 6
flashflood.functional, 7
flashflood.interface.sqlite, 7
flashflood.lod, 7

Index

A

add() (*ListOfDict method*), 7

C

compose() (*in module flashflood.functional*), 7
Container (*class in flashflood.core.container*), 6
Counter (*class in flashflood.core.container*), 6

D

delete() (*ListOfDict method*), 8

F

filter() (*ListOfDict method*), 7
filtered() (*in module flashflood.lod*), 9
find() (*ListOfDict method*), 7
flashflood.core.container (*module*), 6
flashflood.functional (*module*), 7
flashflood.interface.sqlite (*module*), 7
flashflood.lod (*module*), 7

J

join() (*ListOfDict method*), 8

L

ListOfDict (*class in flashflood.lod*), 7
LOD (*in module flashflood.lod*), 8

M

merge() (*ListOfDict method*), 8

P

pick() (*ListOfDict method*), 8

R

reduce() (*ListOfDict method*), 8

S

Sampler (*class in flashflood.core.container*), 6

U

unique() (*ListOfDict method*), 8

V

valuelist() (*in module flashflood.lod*), 9
values() (*ListOfDict method*), 7