Luiti

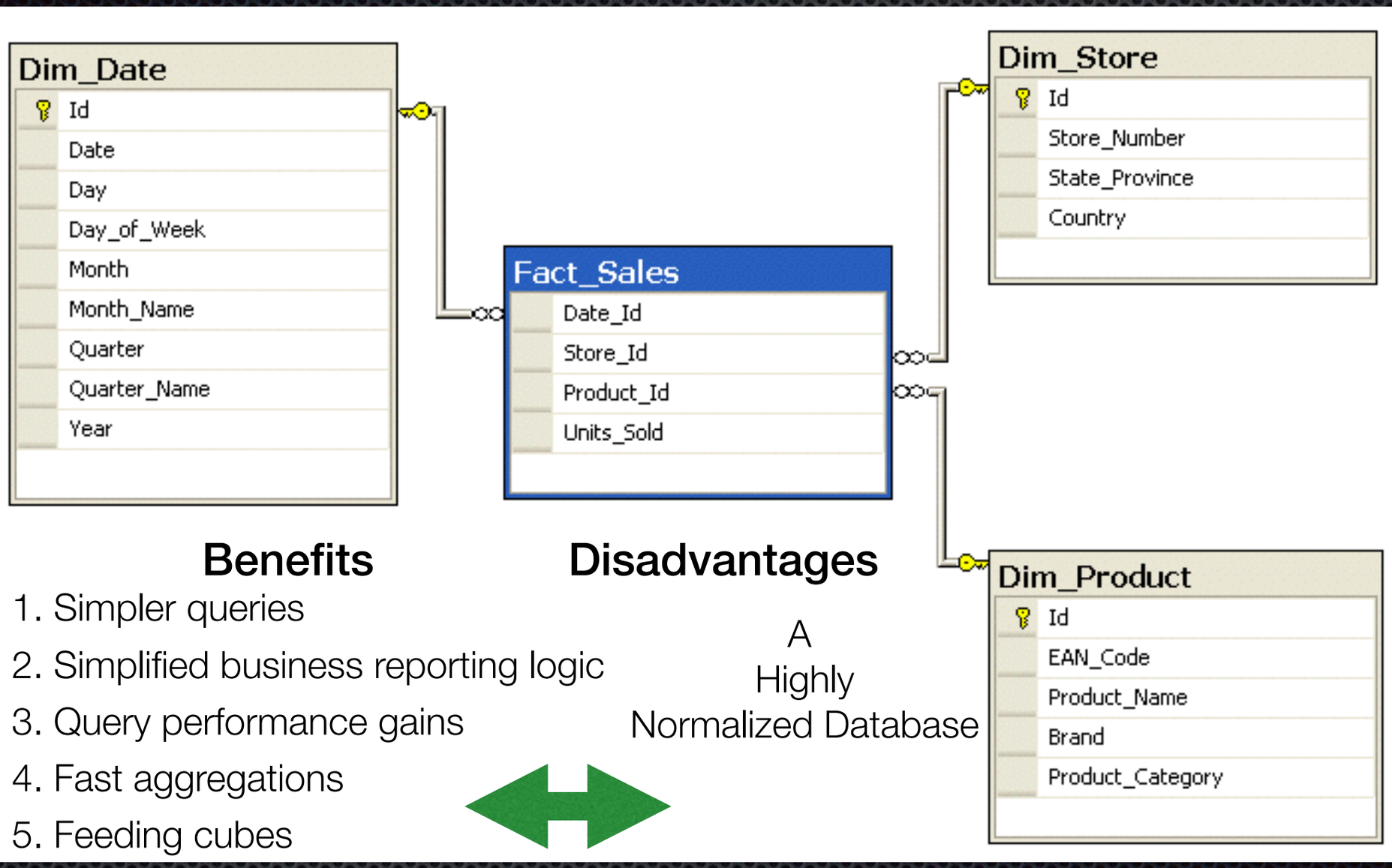
An Offline Task Management Framework

David Chen (@mvj3)
Data Engineer

http://github.com/17zuoye/luiti

July 18, 2015

Star Schema



Benefits

1. Simpler queries
2. Simplified business reporting logic
3. Query performance gains
4. Fast aggregations
5. Feeding cubes

Disadvantages

A
Highly
Normalized Database

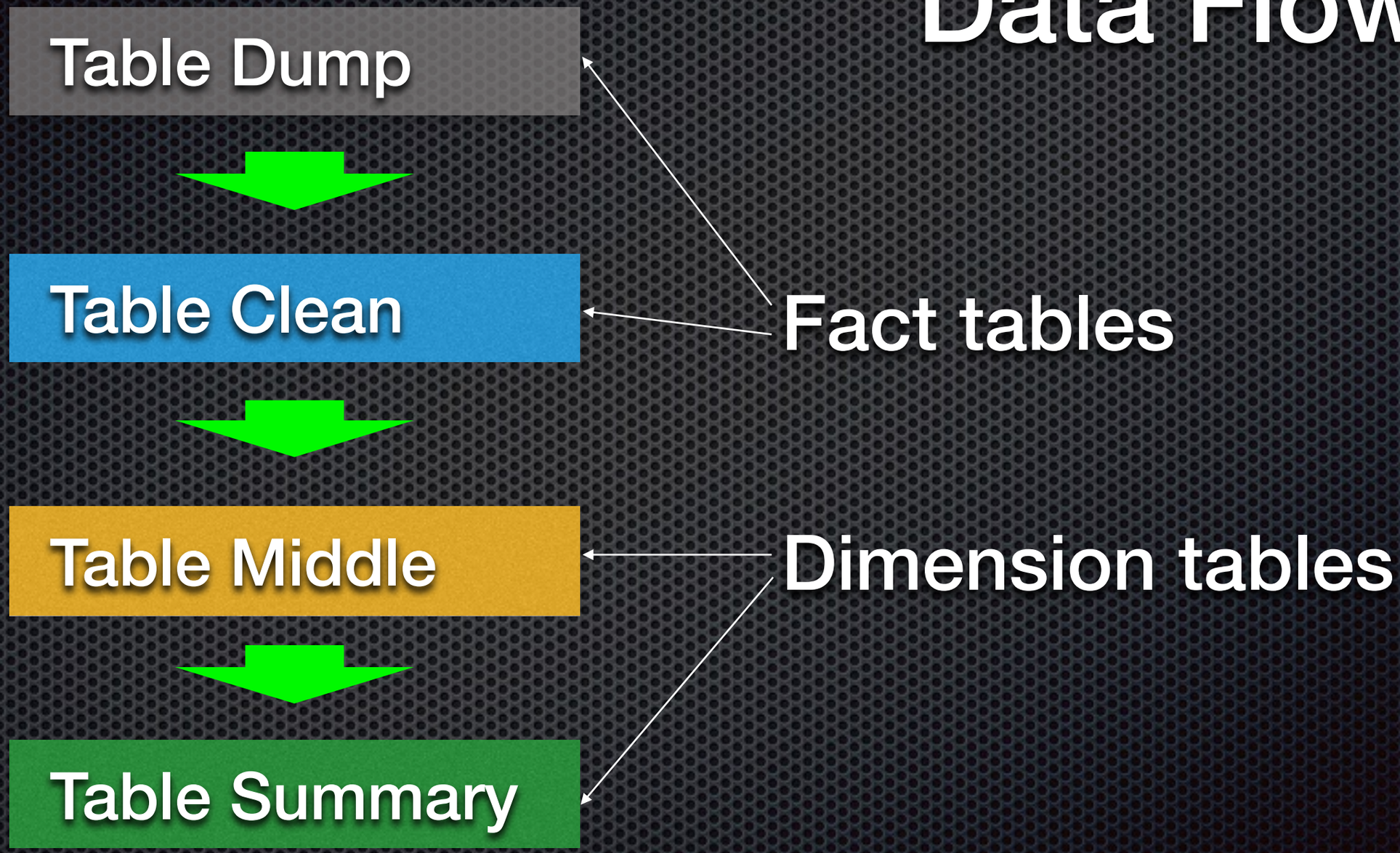
Ideal mode?!

https://en.wikipedia.org/wiki/Star_schema

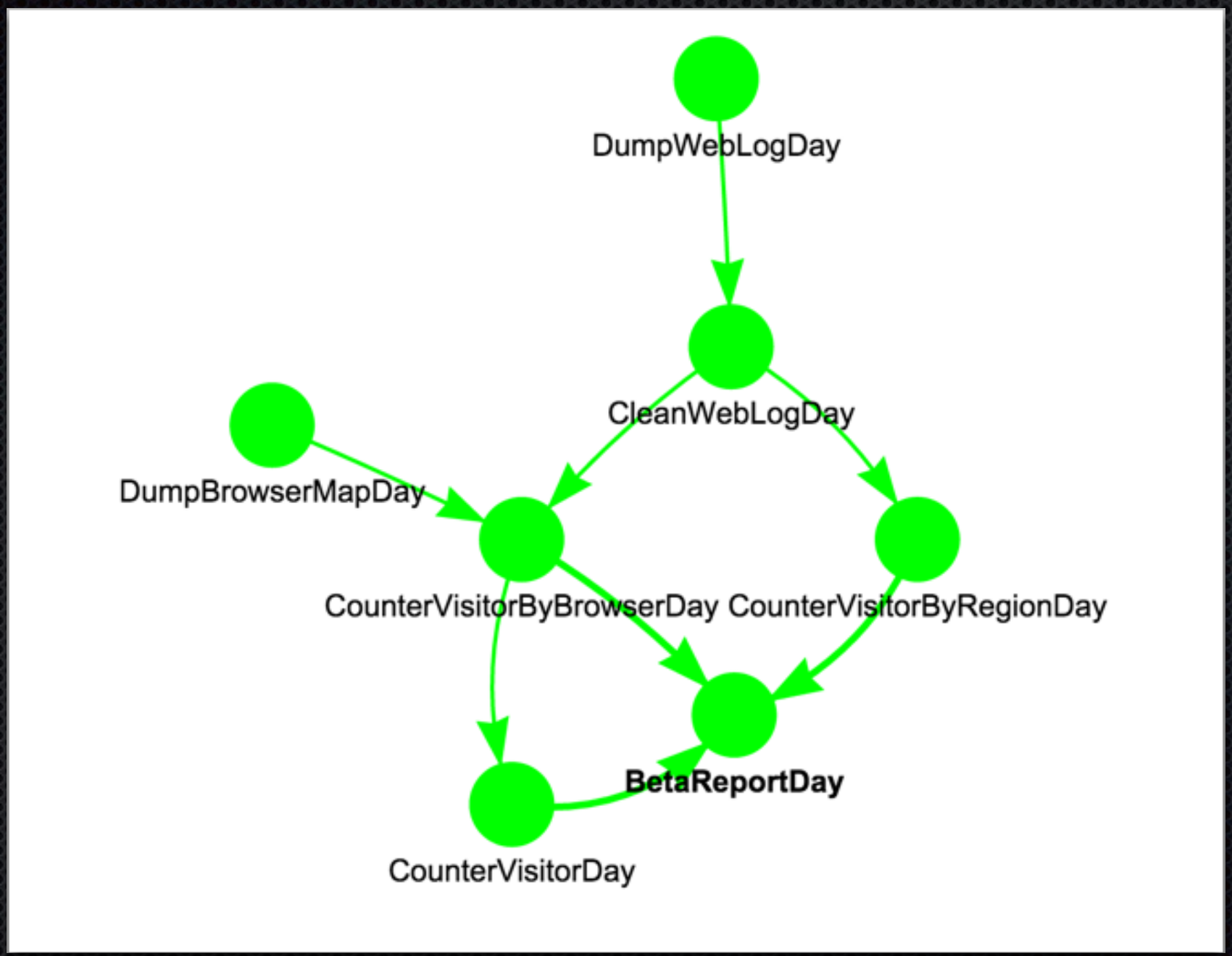
http://emhughes.com/arbitrary/observation-starfish-cool/

Hierarchical Data Warehouse

Data Flow

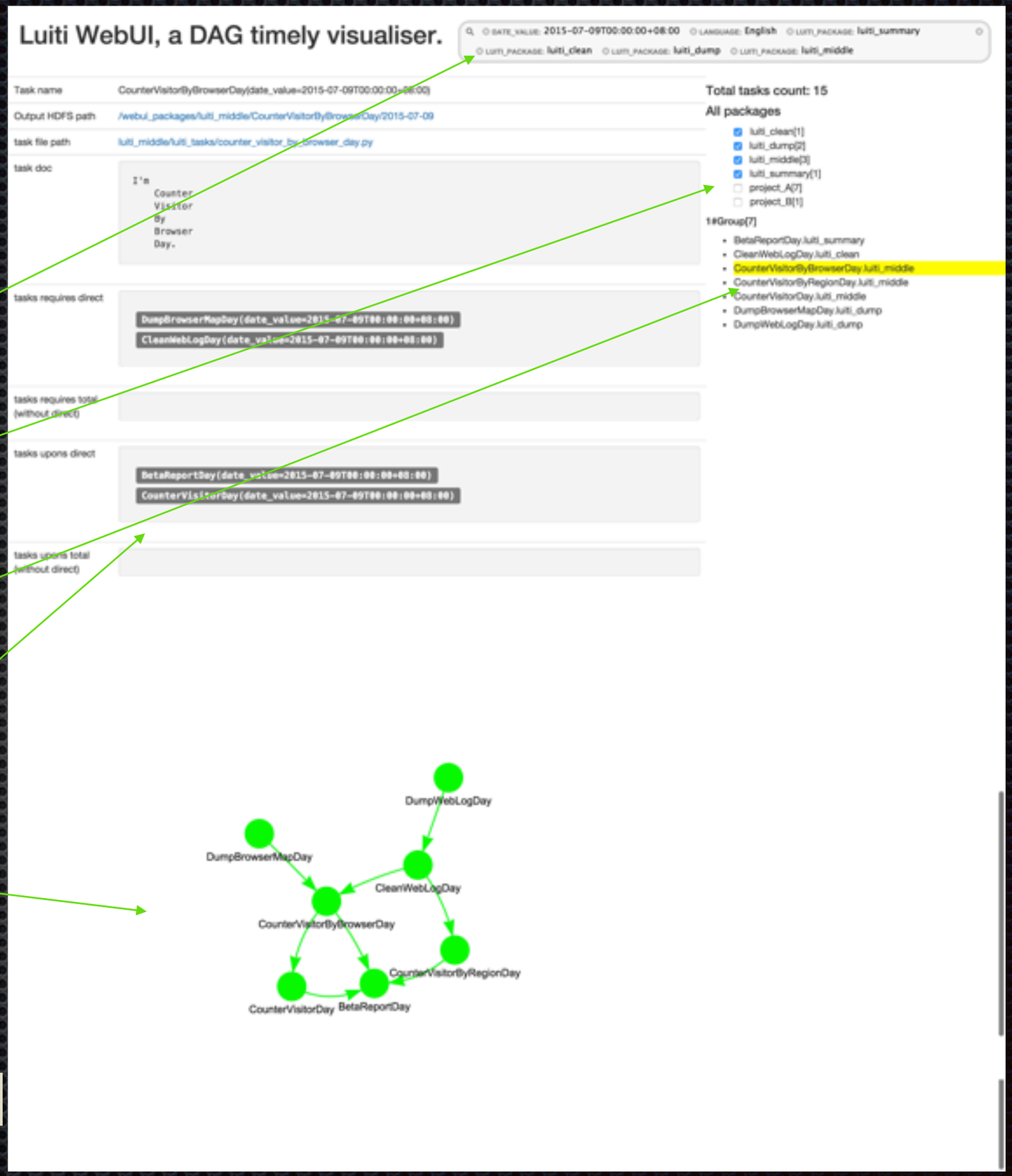


DAG (Directed acyclic graph)



Luiti WebUI Tasks List

1. Parameters
2. Packages
3. Tasks
4. Selected Task Info
5. Tasks DAG




```
~/github/17zuoye/luiti on master 15:43:06
$ ./example_webui_run.py
```

http://localhost:8082/luiti/dag_visualiser?date_value=2015-07-09T00%3A00%3A00%2B08%3A00&language=English&luiti_package=luiti_summary&luiti_package=luiti_clean&luiti_package=luiti_dump&luiti_package=luiti_middle

Luiti WebUI Task Show

1. Task name

2. Output path link

3. Source code link

4. Current task document

5. Current task relations

6. Current task DAG

http://localhost:8082/luiti/dag_visualiser?date_value=2015-07-09T00%3A00%3A00%2B08%3A00&language=English&luiti_package=luiti_summary&luiti_package=luiti_clean&luiti_package=luiti_dump&luiti_package=luiti_middle&task_cls=BetaReportDay

Luiti is built on top of Luigi

Luigi was built at Spotify®, mainly

http://nerdreactor.com/2012/01/26/new-2d-super-mario-bros-game-headed-to-3ds/

http://1.bp.blogspot.com/--i37qQi6WBc/UHPoISo7Y6I/AAAAAAAAKnU/XIHIo223-pE/s1600/mario-bross-493985.jpg

Luigi's Task Class

1. Output	**Atomic** LocalTarget or hdfs.HdfsTarget
2. Input	Other luigi tasks or none
3. Parameters	luigi.Parameter, e.g. DateParameter
4. Execute Logic	`run` or (`mapper`, `reducer`)

http://github.com/spotify/luigi
http://github.com/17zuoye/luiti#a-simple-guide-to-luigi

A Luigi Example (1)

Luigi Task – breakdown

```
import luigi

class MyTask(luigi.Task):
    param = luigi.Parameter(default=42)

    def requires(self):
        return SomeOtherTask(self.param)

    def run(self):
        f = self.output().open('w')
        print >>f, "hello, world"
        f.close()

    def output(self):
        return luigi.LocalTarget('/tmp/foo/bar-%s.txt' % self.param)

if __name__ == '__main__':
    luigi.run()
```

The business logic of the task

Where it writes output

What other tasks it depends on

Parameters for this task

From Luigi's Presentation

http://www.slideshare.net/erikbern/luigi-presentation-nyc-data-science

A Luigi Example (2)

Easy command line integration

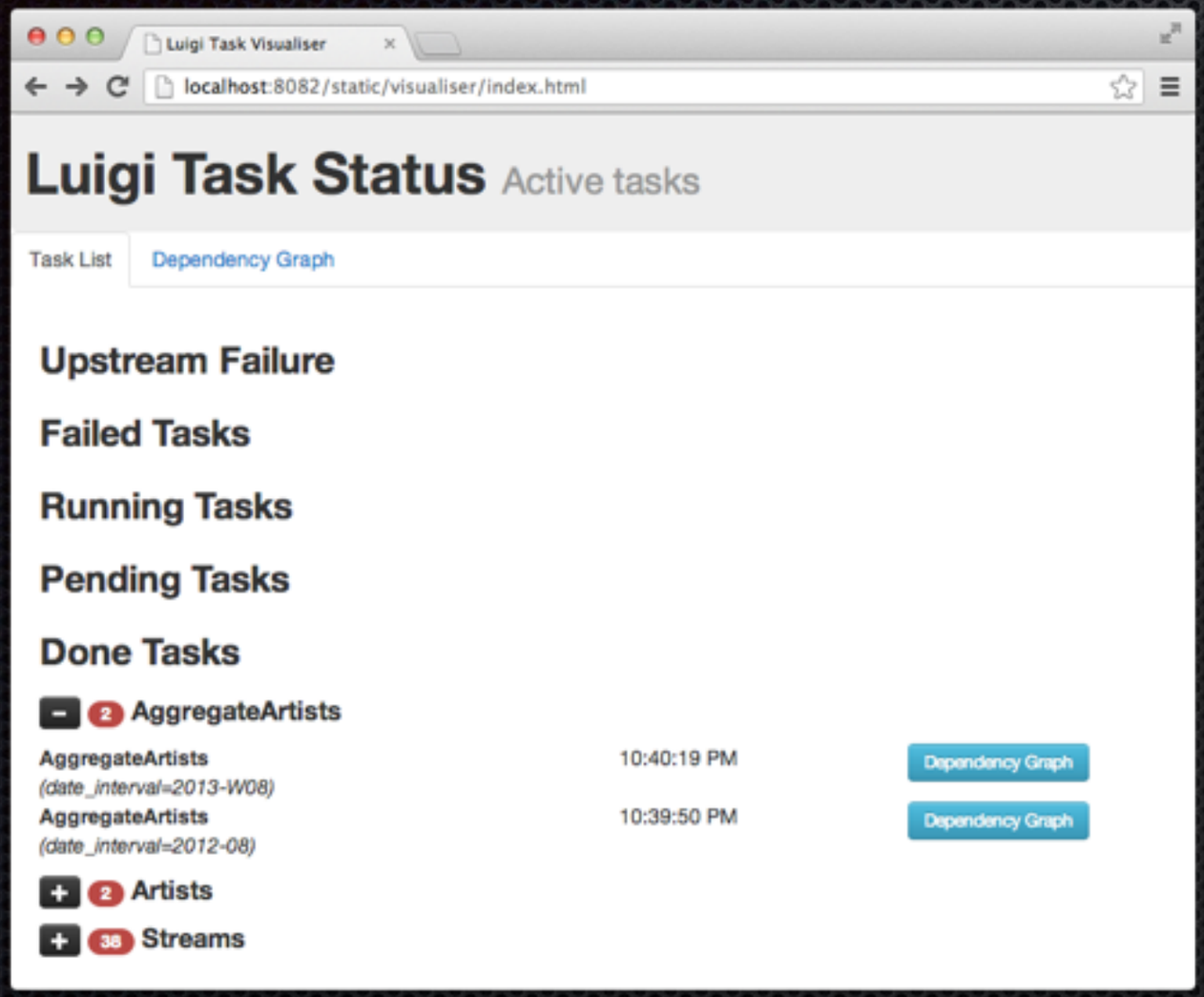
So easy that you **want** to use Luigi for it

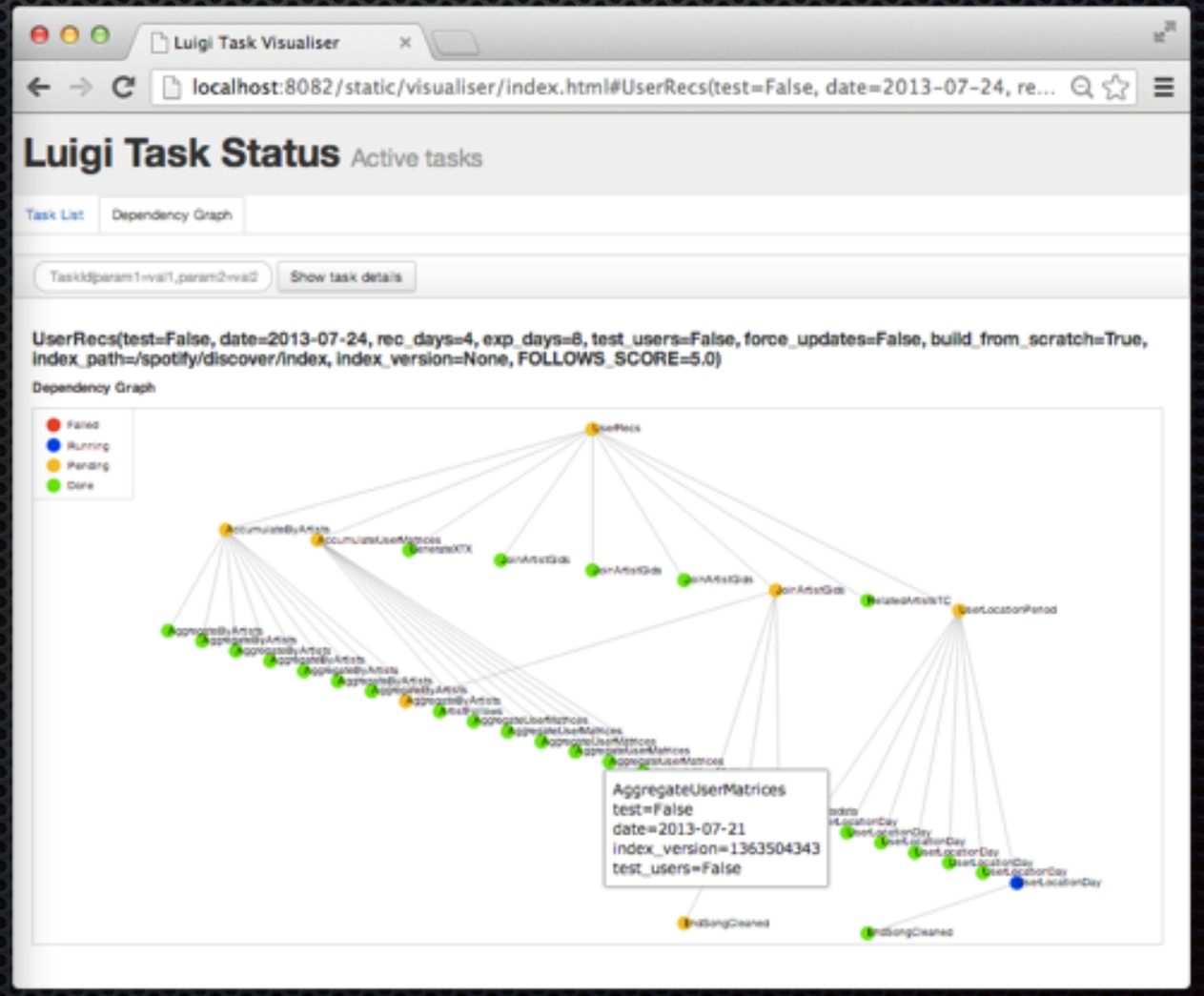
```
$ python my_task.py MyTask --param 43
INFO: Scheduled MyTask(param=43)
INFO: Scheduled SomeOtherTask(param=43)
INFO: Done scheduling tasks
INFO: [pid 20235] Running   SomeOtherTask(param=43)
INFO: [pid 20235] Done      SomeOtherTask(param=43)
INFO: [pid 20235] Running   MyTask(param=43)
INFO: [pid 20235] Done      MyTask(param=43)
INFO: Done
INFO: There are no more tasks to run at this time
INFO: Worker was stopped. Shutting down Keep-Alive thread
$ cat /tmp/foo/bar-43.txt
hello, world
$
```

From Luigi's Presentation

http://www.slideshare.net/erikbern/luigi-presentation-nyc-data-science

A Luigi Example (3)





https://github.com/spotify/luigi/blob/master/doc/web_server.png

https://github.com/spotify/luigi/blob/master/doc/user_recs.png

the Power of Python!

a Functional programming language.

https://docs.python.org/2/howto/functional.html

http://www.clubfauna.com/articles/reptiles/green-tree-python-care-sheet/

builtin @property in Python

```
class C(object):
    def __init__(self):
        self._x = None

    def getx(self):
        return self._x

    def setx(self, value):
        self._x = value

    def delx(self):
        del self._x

    x = property(getx, setx, delx, "I'm the 'x' property.")
```

1. First-class function

2. Decorator

```
class Parrot(object):
    def __init__(self):
        self._voltage = 100000

    @property
    def voltage(self):
        """Get the current voltage."""
        return self._voltage
```

https://docs.python.org/2/library/functions.html#property

@cached_property

cached-property/cached_p ×

mvj3

GitHub, Inc. [US] https://github.com/pydanny/cached-property/blob/master/cached_property.py

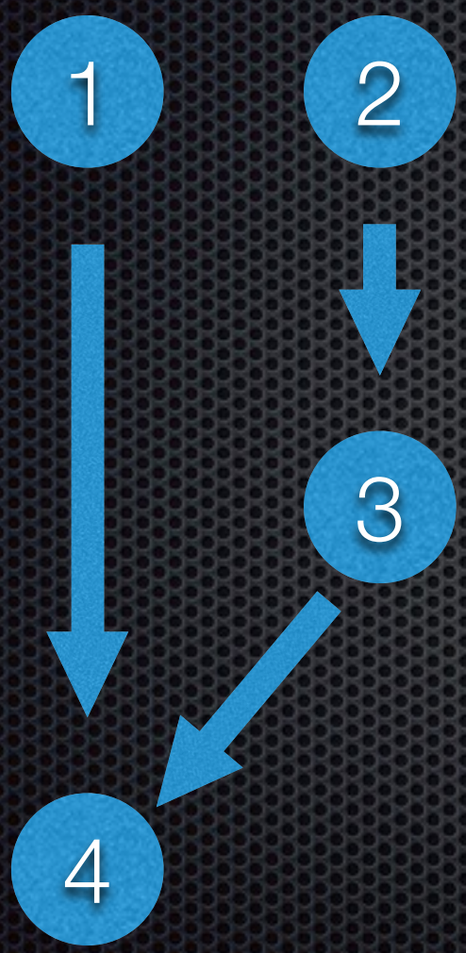
```
11
12  class cached_property(object):
13      """
14      A property that is only computed once per instance and then replaces itself
15      with an ordinary attribute. Deleting the attribute resets the property.
16      Source: https://github.com/bottlepy/bottle/commit/fa7733e075da0d790d809aa3d2f53071897e6f76
17      """  # noqa
18
19      def __init__(self, func):
20          self.__doc__ = getattr(func, '__doc__')
21          self.func = func
22
23      def __get__(self, obj, cls):
24          if obj is None:
25              return self
26          value = obj.__dict__[self.func.__name__] = self.func(obj)
27          return value
28
```

https://github.com/pydanny/cached-property

https://docs.python.org/2/reference/datamodel.html#object.__get__

Data processing in Python

Actually,
It's a DAG !




```python
class DAG(object):
    """ Computing line by line in old style. """

    def run(self):
        self.val_1 = Node(value=1).compute()
        self.val_2 = Node(value=2).compute()
        self.val_3 = Node(value=3, other_node=self.val_2).compute()
        self.val_4 = Node(value=4, other_node=[self.val_1, self.val_3]).compute()
        return self.val_4


class DAG(object):
    """ Computing in @property style. """

    @property
    def val_1(self):
        return Node(value=1).compute()

    @property
    def val_2(self):
        return Node(value=2).compute()

    @property
    def val_3(self):
        return Node(value=3, other_node=self.val_2).compute()

    @property
    def val_4(self):
        return Node(value=4, other_node=[self.val_1, self.val_3]).compute()

    def run(self):
        return self.val_4
```

```python
class DAGSubclass(DAG):
    """
    Two ways to overwrite property in subclass.
    """

    @property
    def val_1(self):
        10

    val_3 = 30
```

Data processing in Luigi

```
60
61 import luigi
62
63 class Node4(luigi.Task):
64
65     value = luigi.Parameter
66
67     def require(self):
68         return [Node1(value=1), Node3(value=3)]
69
70     def run(self):
71         with self.output().open("w") as file_write:
72             result = self.compute(self.input())
73             file_write.write(result)
74
75     def output(self):
76         return luigi.LocalTarget("/tmp/node_4/%s" % self.value)
77
78 """""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
79
80 class DAG(object):
81     """ Computing in @property style. """
82
83     @property
84     def val_4(self):
85         return Node(value=4, other_node=[self.val_1, self.val_3]).compute()
86
87     def run(self):
88         return self.val_4
89
```

Luiti Command Line

mvj3 — mvj3@Davids-MacBook-Pro: ~ — ~ — zsh

```
~ 14:32:15
$ luiti
usage: luiti [-h] {ls,new,generate,info,clean,run,webui} ...

Luiti tasks manager.

optional arguments:
  -h, --help            show this help message and exit

subcommands:
  valid subcommands

  {ls,new,generate,info,clean,run,webui}
    ls                  list all current luiti tasks.
    new                 create a new luiti project.
    generate            generate a new luiti task python file.
    info                show a detailed task.
    clean               manage files that outputed by luiti tasks.
    run                 run a luiti task.
    webui               start a luiti DAG visualiser.

~ 14:32:18
$
```

Data processing in Luiti(1)

```
~/bitbucket/mvj3_/luiti_keynote on  master  13:46:22
$ luiti new --project-name dag_keynote
[info] generate dag_keynote/README.markdown file.
[info] generate dag_keynote/setup.py file.
[info] generate dag_keynote/dag_keynote/__init__.py file.
[info] generate dag_keynote/dag_keynote/luiti_tasks/__init__.py file.
[info] generate dag_keynote/dag_keynote/luiti_tasks/__init_luiti.py file.
[info] generate dag_keynote/tests/test_main.py file.
~/bitbucket/mvj3_/luiti_keynote/dag_keynote on  master!  13:46:58
$ luiti generate --task-name Node4Day
[info] generate /Users/mvj3/bitbucket/mvj3_/luiti_keynote/dag_keynote/dag_keynote/luiti_tasks/node4_day.py file.
~/bitbucket/mvj3_/luiti_keynote/dag_keynote on  master!  13:50:18
$ luiti generate --task-name Node1Day
[info] generate /Users/mvj3/bitbucket/mvj3_/luiti_keynote/dag_keynote/dag_keynote/luiti_tasks/node1_day.py file.
~/bitbucket/mvj3_/luiti_keynote/dag_keynote on  master!  13:50:21
$ luiti generate --task-name Node2Day
[info] generate /Users/mvj3/bitbucket/mvj3_/luiti_keynote/dag_keynote/dag_keynote/luiti_tasks/node2_day.py file.
~/bitbucket/mvj3_/luiti_keynote/dag_keynote on  master!  13:50:24
$ luiti generate --task-name Node3Day
[info] generate /Users/mvj3/bitbucket/mvj3_/luiti_keynote/dag_keynote/dag_keynote/luiti_tasks/node3_day.py file.
~/bitbucket/mvj3_/luiti_keynote/dag_keynote on  master!  13:54:02
$ luiti webui
( \       |\      /|\__    __/\__    __/\__    __/
| (       | )    ( |   ) (       ) (       ) (
| |       | |    | |   | |       | |       | |
| |       | |    | |   | |       | |       | |
| |       | |    | |   | |       | |       | |
| (____/\| (___) |___) (___    | |    ___) (___
(_______/(_______)\_______/    )_(    \_______/
Luiti WebUI is mounted on http://localhost:8082
[I 150717 13:54:02 server:65] Scheduler starting up
[I 150717 13:54:04 web:1811] 304 GET /luiti/dag_visualiser (127.0.0.1) 1.37ms
[I 150717 13:54:04 web:1811] 200 GET /luiti/init_data.json (127.0.0.1) 13.84ms
```

Data processing in Luiti(2)

Convention over Configuration

node4_day.py (~/bitbucket/mvj3_/luiti_ke...ag_keynote/dag_keynote/luiti_tasks) - VIM3

```
</luiti_keynote/dag_keynote/
▾ dag_keynote/
  ▾ luiti_tasks/
      __init__.py
      __init_luiti.py
      node1_day.py
      node2_day.py
      node3_day.py
      node4_day.py
    __init__.py
  ▸ tests/
  README.markdown
  setup.py
```

```
 6 class Node1Day(TaskDay):
 7
 8     root_dir = "/tmp"
<ynote/luiti_tasks/node1_day.py  Line:8/8[100%]Col:21Buf:#6[34][0x22]
 6 class Node2Day(TaskDay):
 7
 8     root_dir = "/tmp"
<ynote/luiti_tasks/node2_day.py  Line:8/8[100%]Col:21Buf:#7[34][0x22]
 2
 3 from .__init_luiti import luigi, TaskDay
 4
 5
 6 @luigi.ref_tasks("Node2Day")
 7 class Node3Day(TaskDay):
 8
 9     root_dir = "/tmp"
10
11     def requires(self):
12         return self.Node2Day_task
<ynote/luiti_tasks/node3_day.py  Line:6/12[50%]Col:24Buf:#8[68][0x44]
 1 # -*-coding:utf-8-*-
 2
 3 from .__init_luiti import TaskDay, luigi, IOUtils
 4
 5
 6 @luigi.ref_tasks("Node1Day", "Node3Day")
 7 class Node4Day(TaskDay):
 8
 9     root_dir = "/tmp"
10
11     def requires(self):
12         return [self.Node1Day_task, self.Node3Day_task]
13
14     def run(self):
15         result = self.compute(self.input())
16         IOUtils.write_json_to_output(result, self.output())
<vj3_/luiti_keynote/dag_keynote <note/luiti_tasks/node4_day.py  Line:16/16[100%]Col:1Buf:#1[32][0x20]
```

Task name	Node4Day(date_value=2015-07-16T00:00:00+08:00)
Output HDFS path	/tmp/2015-07-16/node4_day.json
task file path	dag_keynote/luiti_tasks/node4_day.py
task doc	
tasks requires direct	Node3Day(date_value=2015-07-16T00:00:00+08:00) Node1Day(date_value=2015-07-16T00:00:00+08:00)

Luiti = Luigi + ?

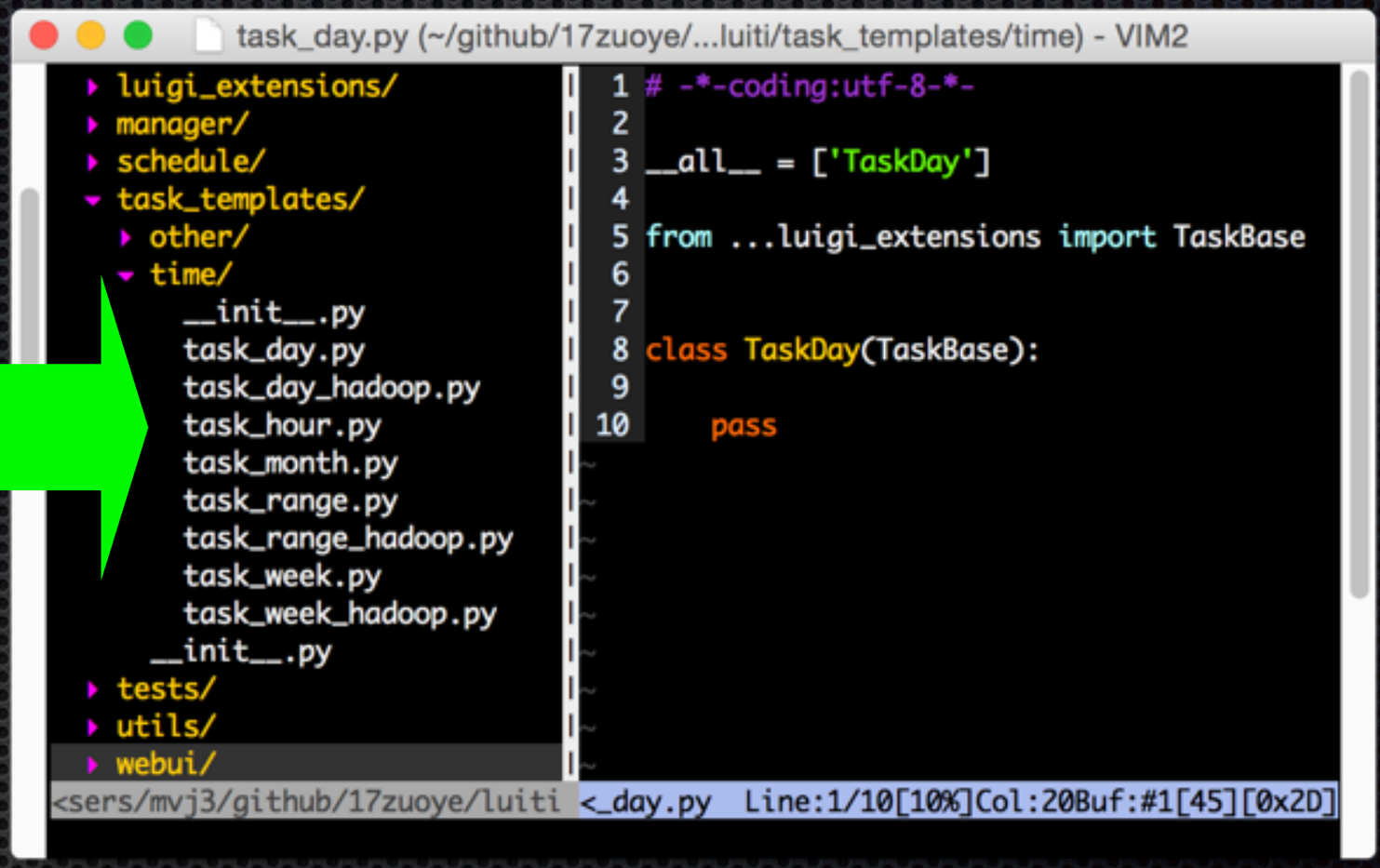
task_base.py + (~/github/17zuoye/luiti/luiti/luigi_extensions) - VIM1

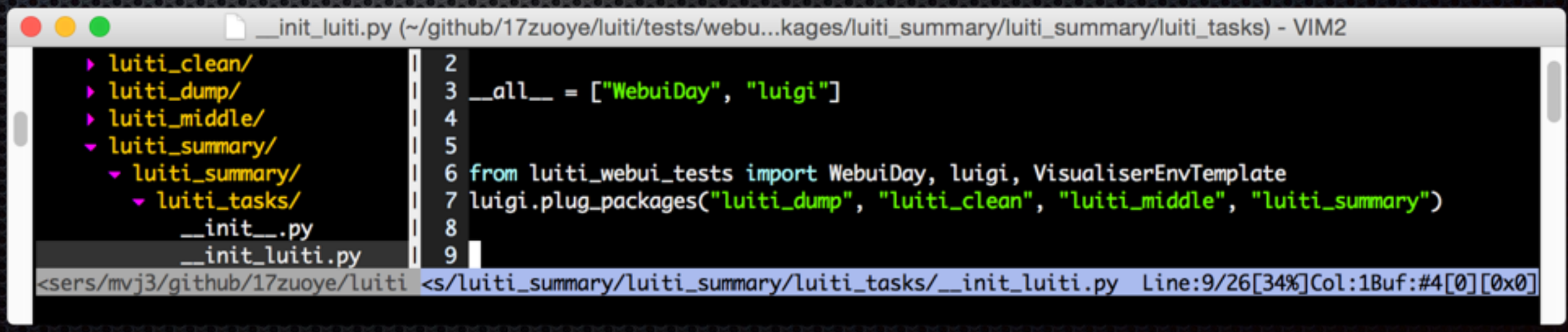
```
144 class TaskBase(luigi.Task, ExtUtils.ExtendClass):
145
146     date_value = ArrowParameter()
147
148     @cached_property
149     def data_file(self):
150         return os.path.join(self.data_dir, self.data_name + ".json")
151
152     @cached_property
153     def date_str(self):
154
155     @cached_property
156     def date_type(self):
157         return luiti_config.get_date_type(self.__class__.__name__)
158
159     @cached_property
160     def date_value_by_type_in_last(self):
161
162     @cached_property
163     def date_value_by_type_in_begin(self):
164
165     @cached_property
166     def date_value_by_type_in_end(self):
167
168     @cached_property
169     def pre_task_by_self(self):
170
171     @cached_property
172     def task_clsname(self):
173
174     @cached_property
175     def package_name(self):
~
~
<iti/luigi_extensions/task_base.py [+] Line:175/175[100%]Col:1Buf:#9[32][0x20]
```

https://github.com/17zuoye/luiti#task-builtin-properties

Luiti = Luigi +TaskDay + Package

Luiti = Luigi +Time





Context

Abstract Machine

package

class

object

function

CPU

IO

luiti.TaskBase

luigi.Task

@cached_property

@property

Luiti Code Architecture

task templates

luiti task

WebUI

```
/Users/mvj3/github/17zuoye/luiti/
▸ bin/
▾ luiti/
  ▸ daemon/
  ▸ java/
  ▸ luigi_decorators/
  ▸ luigi_extensions/
  ▸ manager/
  ▸ schedule/
  ▸ task_templates/
  ▸ tests/
  ▸ utils/
  ▸ webui/
    __init__.py
▸ screenshots/
▸ tests/
  changelog.markdown
```

daemon

ptm

query engine

web

luigi extensions

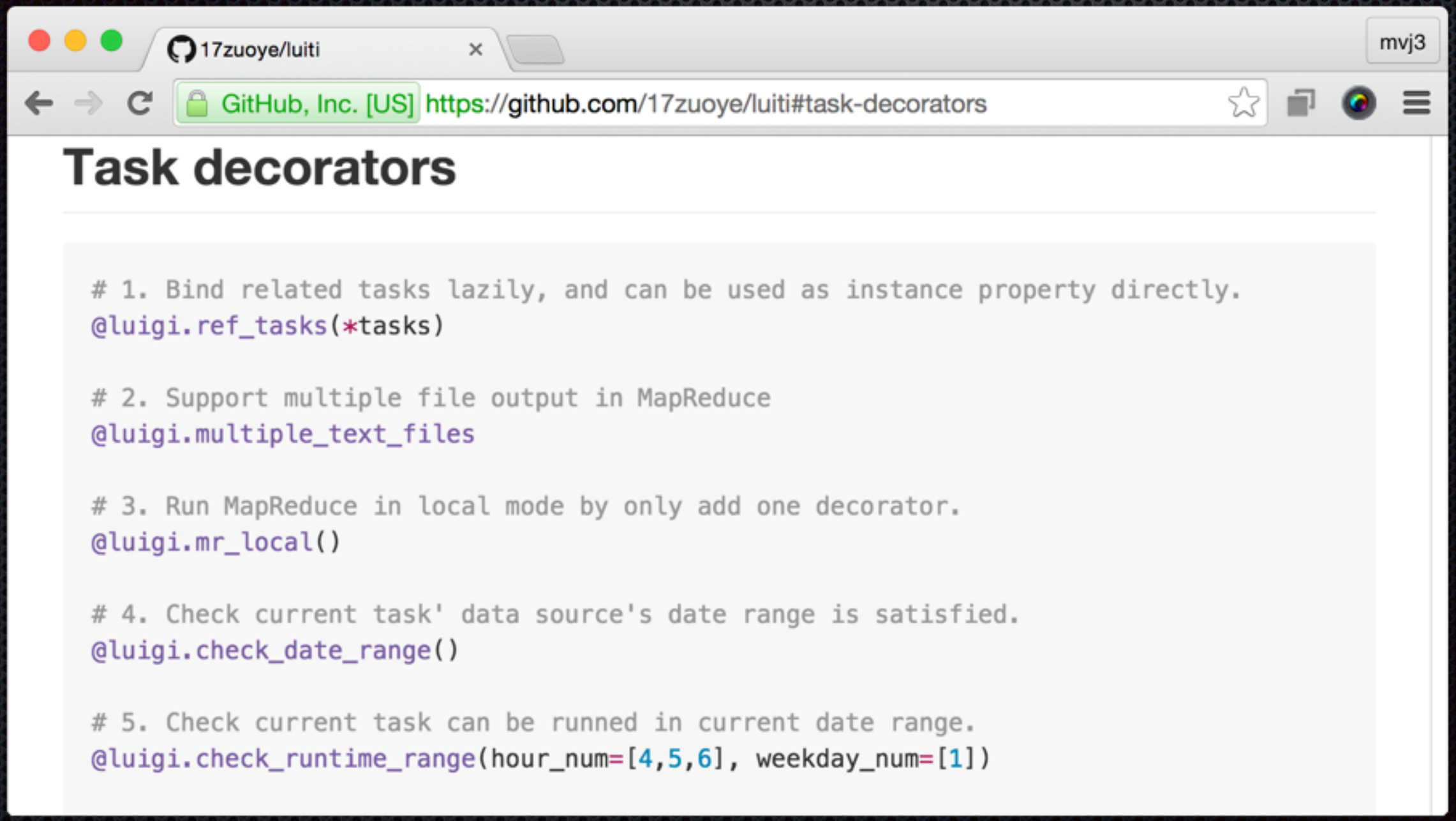
manager

luiti cmd

luigi decorators

luigi task

Luiti Decorators



https://github.com/17zuoye/luiti#task-decorators

MapReduce unit test

foobar_day.py (~/github/17zuoye/luiti/tests/project_A/luiti_tasks) - VIM1

```
5
6  class FoobarDay(TaskDayHadoop):
7      """
8      A MapReduce Python Program written in Luiti Task Style, including test case.
9      """
10
11     root_dir = "/foobar"
12
13     def mapper(self, line1):
14         d2 = MRUtils.json_parse(line1)
15         yield d2['uid'], d2
16
17     def reducer(self, uid1, d1):
18         yield '', MRUtils.str_dump({
19             "uid": uid1,
20             "total": sum([i2['count'] for i2 in d1]),
21             "ref": self.ref,
22         })
23
24     ref = NotImplementedError
25
26     def mrtest_input(self):
27         return u"""
28 {"uid": 1, "count": 2}
29 {"uid": 1, "count": 3}
30 {"uid": 2, "count": 1}
31 """
32
33     def mrtest_output(self):
34         return u"""
35 {"uid": 1, "total": 5, "ref": "foobar"}
36 {"uid": 2, "total": 1, "ref": "foobar"}
37 """
38
39     def mrtest_attrs(self):
40         return {
41             "ref": "foobar",
42         }
~
tests/project_A/luiti_tasks/foobar_day.py  Line:5/42[11%]Col:1Buf:#3[0][0x0]
```

test_mr_test_case.py (~/github/17zuo...

```
12 import unittest
13 from luiti import MrTestCase
14
15
16 @MrTestCase
17 class TestMrTestCase(unittest.TestCase):
18
19     mr_task_names = [
20         'FoobarDay',
21     ]
22
23 if __name__ == '__main__':
24     unittest.main()
<case.py  Line:24/24[100%]Col:1Buf:#1[32][0x20]
```

Extend Luiti

luigi_ext.py (~/github/17zuoye/luiti) - VIM2

```python
# -*-coding:utf-8-*-

__all__ = ["TaskDayHadoop"]

from luiti import luigi, TaskDayHadoop
from etl_utils import cached_property

namenode = luigi.configuration.get_config().get("hdfs", "namenode_host", "localhost")


def jobconfs_opts(self):
    return ["fs.defaultFS=hdfs://" + namenode + ":8020",
            "mapreduce.jobtracker.address=" + namenode + ":8031",
            "yarn.resourcemanager.address=" + namenode + ":8032",
            "yarn.resourcemanager.scheduler.address=" + namenode + ":8030",
            "yarn.resourcemanager.admin.address=" + namenode + ":8033",
            "yarn.resourcemanager.webapp.address=" + namenode + ":8088",
            "mapreduce.framework.name=yarn",
            'mapreduce.job.reduces=%s' % self.n_reduce_tasks, ]


@cached_property
def an_exchange_dict(self):
    return dict()  # read some data.


TaskDayHadoop.extend({
    "jobconfs_opts": jobconfs_opts,
    "data_interchange_format": "json",
    "an_exchange_dict": an_exchange_dict,
})
```

luigi_ext.py Line:31/31[100%]Col:2Buf:#1[41][0x29]

https://github.com/17zuoye/luiti#extend-luiti

Data Pipelines Frameworks

Startup	Service	Framework	Github stars
Sportify	Music	luigi	2,908
Airbnb	Travel	airflow	669
Pinterest	Photo	pinball	386
17zuoye	Education	luiti	14
...	...	...	...

They're all hosted on Github, and written in Python!

Thanks!

Question?

@mvj3

http://mvj3.github.io/