

5 - (8) custom operator



Figure 8.1 Building a simple pilot MVP for movie recommender project

- 1.. rating api 5000 .
2. airflow rating api , .

chapter 8 .

docker-compose up -d --build

5000 rating api .

5000 rating api .



사이트에 연결할 수 없음

ec2-13-209-17-10.ap-northeast-2.compute.amazonaws.com에서 연결을 거부했습니다.

다음 방법을 시도해 보세요.

- 연결 확인
- 프록시 및 방화벽 확인

ERR_CONNECTION_REFUSED

새로고침

세부정보

rating api .

Traceback (most recent call last):

```
File "/app.py", line 6, in <module>
    from flask import Flask, jsonify, request
File "/usr/local/lib/python3.8/site-packages/flask/__init__.py", line 14, in <module>
    from jinja2 import escape
ImportError: cannot import name 'escape' from 'jinja2' (/usr/local/lib/python3.8/site-packages/jinja2/__init__.py)
```

<https://stackoverflow.com/questions/71718167/importerror-cannot-import-name-escape-from-jinja2>

Flask .

```
Flask==2.1.0
pandas==0.25.2
Flask-HTTPAuth==4.6.0
click==8.0.4
```

Hello from the Movie Rating API!

```
← → ↻ 주의 요약 | ec2-13-209-17-10.ap-northeast-2.compute.amazonaws.com:5000/ratings
{"limit":100,"offset":0,"result":[{"movieid":196997,"rating":4.0,"timestamp":1546301166,"userid":276677}, {"movieid":1213,"rating":5.0,"timestamp":1546301290,"userid":276677}, {"movieid":59820,"rating":4.0,"timestamp":1546301706,"userid":276677}, {"movieid":2329,"rating":4.0,"timestamp":1546301788,"userid":276677}, {"movieid":4878,"rating":4.0,"timestamp":1546301842,"userid":276677}, {"movieid":589,"rating":4.0,"timestamp":1546302155,"userid":128817}, {"movieid":119145,"rating":2.0,"timestamp":1546302225,"userid":276677}, {"movieid":2716,"rating":4.0,"timestamp":1546302513,"userid":276677}, {"movieid":165,"rating":4.0,"timestamp":1546302549,"userid":276677}, {"movieid":57669,"rating":4.0,"timestamp":1546302549,"userid":276677}, {"movieid":5418,"rating":4.0,"timestamp":1546302687,"userid":276677}, {"movieid":377,"rating":4.5,"timestamp":1546302763,"userid":276677}, {"movieid":1387,"rating":3.5,"timestamp":1546303369,"userid":276677}, {"movieid":70286,"rating":4.0,"timestamp":1546303445,"userid":276677}, {"movieid":56782,"rating":3.0,"timestamp":1546303670,"userid":276677}, {"movieid":116797,"rating":4.0,"timestamp":1546303832,"userid":276677}, {"movieid":114662,"rating":5.0,"timestamp":1546304182,"userid":123250}, {"movieid":4710,"rating":3.5,"timestamp":1546305412,"userid":101489}, {"movieid":122305,"rating":2.0,"timestamp":1546306524,"userid":119693}, {"movieid":7346,"rating":4.0,"timestamp":1546307797,"userid":43168}, {"movieid":54286,"rating":4.5,"timestamp":1546307807,"userid":43168}, {"movieid":168266,"rating":5.0,"timestamp":1546308239,"userid":279311}, {"movieid":2858,"rating":4.0,"timestamp":1546308316,"userid":125474}, {"movieid":163134,"rating":5.0,"timestamp":1546308976,"userid":86604}, {"movieid":168477,"rating":5.0,"timestamp":1546308990,"userid":86604}, {"movieid":356,"rating":5.0,"timestamp":1546309069,"userid":86604}, {"movieid":196997,"rating":5.0,"timestamp":1546309064,"userid":86604}, {"movieid":2724,"rating":4.0,"timestamp":1546309194,"userid":43168}, {"movieid":3000,"rating":5.0,"timestamp":1546309308,"userid":86604}, {"movieid":94677,"rating":3.0,"timestamp":1546309698,"userid":43168}, {"movieid":296,"rating":5.0,"timestamp":1546309714,"userid":138953}, {"movieid":1193,"rating":4.5,"timestamp":1546309729,"userid":138953}, {"movieid":165347,"rating":4.5,"timestamp":1546309734,"userid":43168}, {"movieid":97921,"rating":5.0,"timestamp":1546309806,"userid":86604}, {"movieid":1221,"rating":4.0,"timestamp":1546309848,"userid":138953}, {"movieid":197039,"rating":2.5,"timestamp":1546310326,"userid":78859}, {"movieid":12291,"rating":5.0,"timestamp":1546310751,"userid":86604}, {"movieid":2282,"rating":2.5,"timestamp":1546311490,"userid":43168}, {"movieid":165611,"rating":3.5,"timestamp":1546311682,"userid":126132}, {"movieid":5952,"rating":1.5,"timestamp":1546312215,"userid":144369}, {"movieid":4886,"rating":2.0,"timestamp":1546312229,"userid":144369}, {"movieid":6377,"rating":2.0,"timestamp":1546312230,"userid":144369}, {"movieid":3798,"rating":0.5,"timestamp":1546312232,"userid":144369}, {"movieid":48516,"rating":4.0,"timestamp":1546312247,"userid":144369}, {"movieid":57669,"rating":4.0,"timestamp":1546312535,"userid":30608}, {"movieid":45761,"rating":4.5,"timestamp":1546316081,"userid":45761}, {"movieid":2600,"rating":3.0,"timestamp":1546317516,"userid":459341}, {"movieid":79132,"rating":5.0,"timestamp":1546318013,"userid":13801}, {"movieid":1198,"rating":3.0,"timestamp":1546318113,"userid":13801}, {"movieid":99114,"rating":4.0,"timestamp":1546318168,"userid":13801}, {"movieid":79132,"rating":5.0,"timestamp":1546318204,"userid":13801}, {"movieid":3753,"rating":5.0,"timestamp":1546320735,"userid":144369}, {"movieid":195161,"rating":1.5,"timestamp":1546322469,"userid":161037}, {"movieid":143899,"rating":3.0,"timestamp":1546323932,"userid":34164}, {"movieid":5618,"rating":5.0,"timestamp":1546327224,"userid":147049}, {"movieid":51077,"rating":4.0,"timestamp":1546327268,"userid":147049}, {"movieid":28662,"rating":5.0,"timestamp":1546327375,"userid":147049}
```

ratings api .

```
/ratings?start_date=2022-10-01&end_date=2022-01-02
```

8.1 fetch +rating DAG

AIRFLOW rating api fetch , .

github .

https://github.com/BasPH/data-pipelines-with-apache-airflow/blob/master/chapter08/dags/01_python.py

```
def _get_ratings(start_date, end_date, batch_size=100):
    session, base_url = _get_session()

    yield from _get_with_pagination(
        session=session,
        url=base_url + "/ratings",
        params={"start_date": start_date, "end_date": end_date},
        batch_size=batch_size,
    )

yield from .
```

<https://docs.python.org/3/reference/expressions.html#yieldexpr>

asynchronous

DAG: 01_python Fetches ratings from the Movielens API using the Python Operator.

Tree View | Graph View | Task Duration | Task Tries | Landing Times | Ga

2019-01-10T00:00:00Z | Runs: 25 | Update

PythonOperator



output

, airflow scheduler webserver container csv output

(/data/python/ratings/{{ds}}.json)

/var/lib/docker/volumes ,output

ratings rankings

```
root@ip-172-31-29-78: /var/lib/docker/volumes/chapter08_data/_data/python/ratings
{"movieId": 196997, "rating": 4.0, "timestamp": 1546301166, "userId": 27667}, {"
"movieId": 1213, "rating": 5.0, "timestamp": 1546301290, "userId": 27667}, {"mov
ieId": 55820, "rating": 4.0, "timestamp": 1546301706, "userId": 27667}, {"movieI
d": 2329, "rating": 4.0, "timestamp": 1546301788, "userId": 27667}, {""movieId":
4878, "rating": 4.0, "timestamp": 1546301842, "userId": 27667}, {""movieId": 589,
"rating": 4.0, "timestamp": 1546302155, "userId": 128817}, {""movieId": 119145,
"rating": 2.0, "timestamp": 1546302225, "userId": 27667}, {""movieId": 2716, "rat
ing": 4.0, "timestamp": 1546302513, "userId": 27667}, {""movieId": 165, "rating":
4.0, "timestamp": 1546302548, "userId": 27667}, {""movieId": 57669, "rating": 4.
0, "timestamp": 1546302649, "userId": 27667}, {""movieId": 5418, "rating": 4.0,
"timestamp": 1546302687, "userId": 27667}, {""movieId": 377, "rating": 4.5, "times
tamp": 1546302763, "userId": 27667}, {""movieId": 1387, "rating": 3.5, "timestamp
": 1546303369, "userId": 27667}, {""movieId": 70286, "rating": 4.0, "timestamp":
1546303445, "userId": 27667}, {""movieId": 56782, "rating": 3.0, "timestamp": 154
6303870, "userId": 27667}, {""movieId": 116797, "rating": 4.0, "timestamp": 15463
03982, "userId": 27667}, {""movieId": 114662, "rating": 5.0, "timestamp": 1546304
182, "userId": 123850}, {""movieId": 4710, "rating": 3.5, "timestamp": 1546305412
, "userId": 101489}, {""movieId": 122906, "rating": 2.0, "timestamp": 1546306524,
"userId": 119993}, {""movieId": 7346, "rating": 4.0, "timestamp": 1546307797, "u
serId": 43168}, {""movieId": 54286, "rating": 4.5, "timestamp": 1546307807, "user
Id": 43168}, {""movieId": 168266, "rating": 5.0, "timestamp": 1546308239, "userId
": 27931}, {""movieId": 2858, "rating": 4.0, "timestamp": 1546308316, "userId": 1
25474}, {""movieId": 163134, "rating": 5.0, "timestamp": 1546308975, "userId": 86
"2019-01-01.json" [noeol] 1L, 36326C 1,1 All
```

rankings:

```
root@ip-172-31-29-78: /var/lib/docker/volumes/chapter08_data/_data/py
movieId,avg_rating,num_ratings
5618,4.666666666666667,3
56367,4.666666666666667,3
3147,4.5,4
98154,4.5,3
164179,4.5,3
296,4.333333333333333,3
4720,4.166666666666667,3
4993,4.166666666666667,3
196997,4.0,4
356,4.0,3
7451,4.0,3
194951,4.0,3
4886,3.3333333333333335,3
3623,2.5,3
~
~
```

hook

hook .

hook .(.)

https://github.com/BasPH/data-pipelines-with-apache-airflow/blob/master/chapter08/dags/02_hook.py

connection python DAG .

hook .

hook is a place and usually an interface provided in packaged code that allows a programmer to insert customized programming

customizing .

custom operator

.

https://github.com/BasPH/data-pipelines-with-apache-airflow/blob/master/chapter08/dags/03_operator.py

output path .

hook .

sensor

<https://github.com/BasPH/data-pipelines-with-apache-airflow/blob/40b52ba6fcd3203b194be9e1c2850135997215a/chapter08/dags/custom/sensors.py>