Optimizing Data Retrieval with Python Celery

Jenna Conn & Hannah Cline

Current Outline

Jenna Conn & Hannah Cline

Grandma's Chicken Salad Sandwich
Grilled or Blackened Trigger
Mediterranean Chicken Salad

Background

When do you use data?

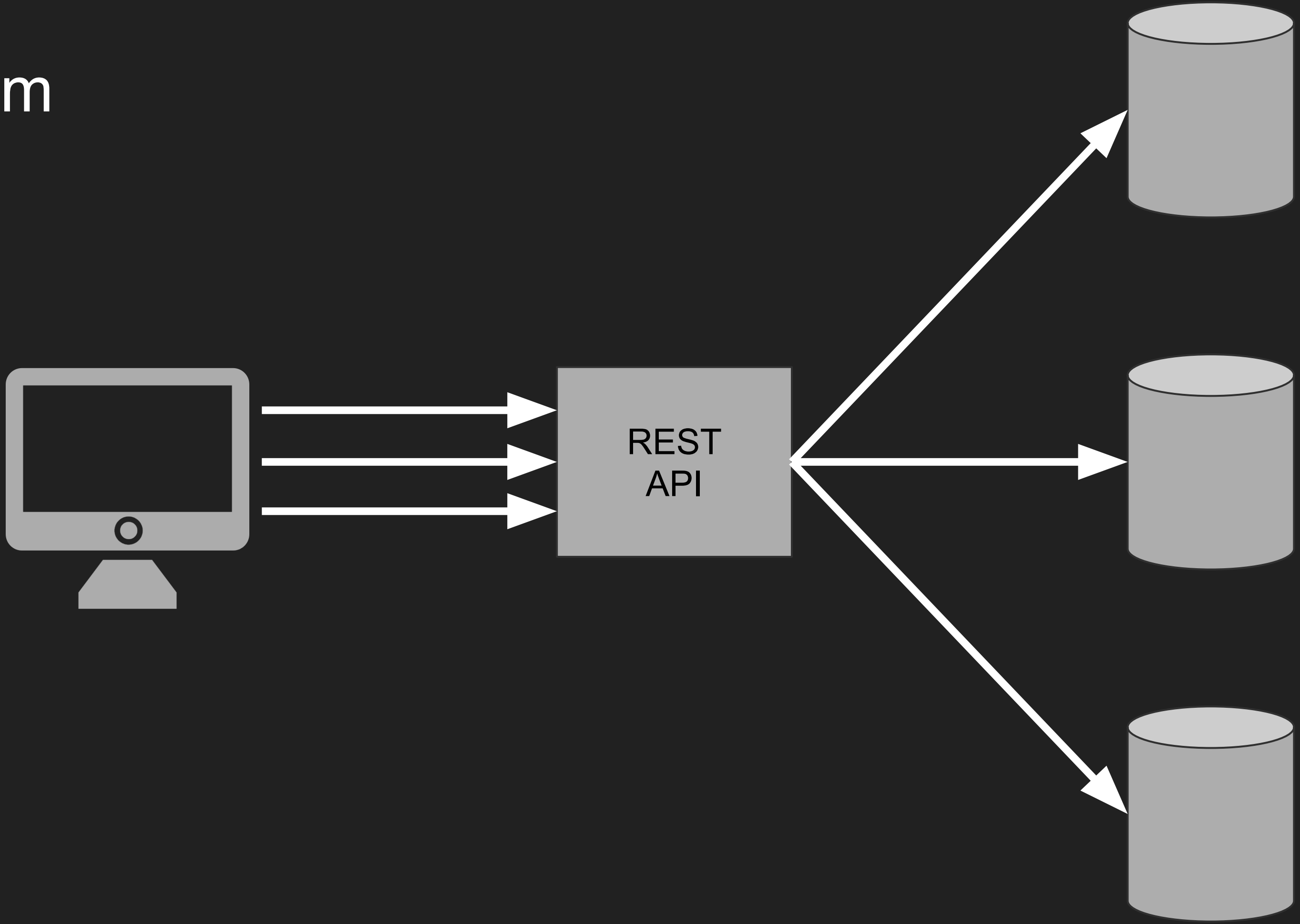
Choosing what to eat

Online shopping

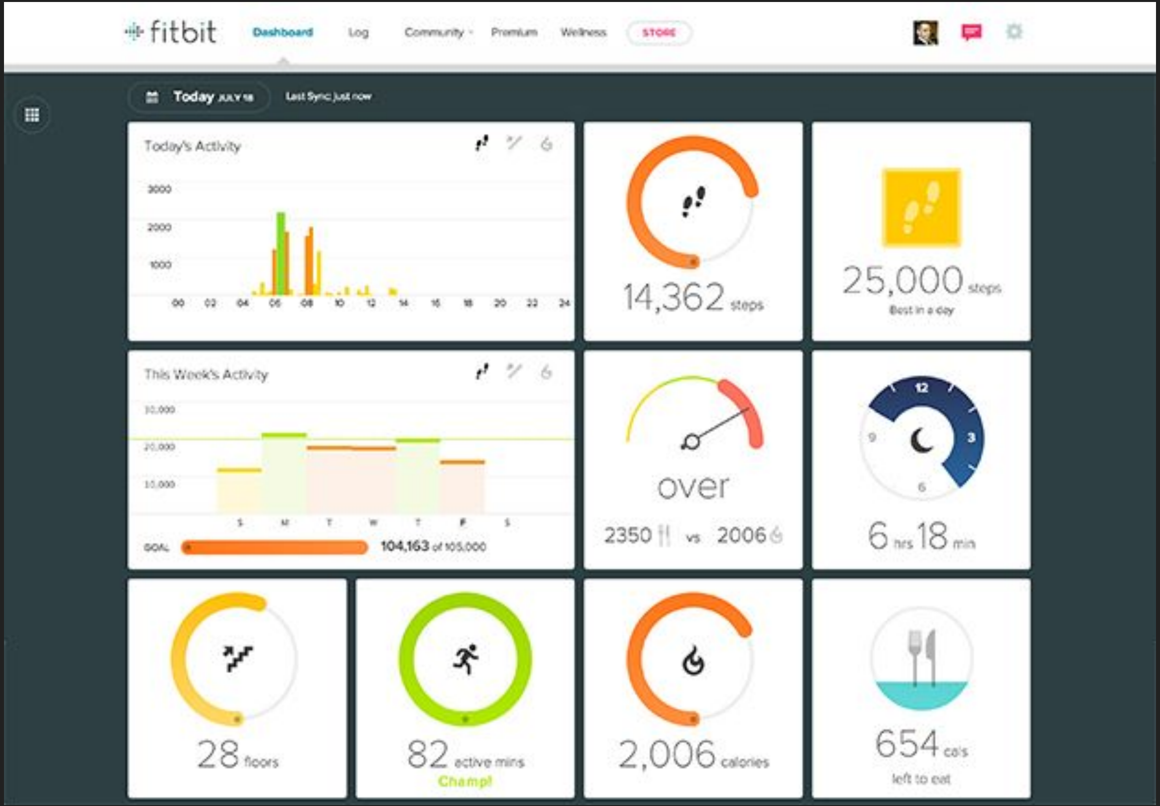
Budget Apps

Data Driven Decision Making – See 10 Tips For Your Business Success. datapine. (2020, August 17). https://www.datapine.com/blog/data-driven-decision-making-in-businesses/.

Problem



Dashboards



Staff, F. (2017, March 14). *Fitbit Dashboard Updated with Weekly Activity and More*. Fitbit Blog. https://blog.fitbit.com/fitbit-dashboard-updated-with-weekly-activity-and-more/.

Solutions

- Optimize DB queries
- Batch queries together
- Make more specialized API’s (microservices)
- Create a task queue

DON'T BRING ME PROBLEMS, BRING ME SOLUTIONS.
WHEN I BRING YOU SOLUTIONS, YOU TELL ME THEY WON'T WORK AND TREAT ME LIKE A MORON.
WHAT'S THE PROBLEM WITH THAT?
www.dilbert.com scottadams@aol.com
© 2007 Scott Adams, Inc./Dist. by UFS, Inc.

Adams, S. (1970, November 7). *Dilbert Comic Strip on November 03, 2007*. Dilbert. https://dilbert.com/strip/2007-11-03.

Task Queues

- Distribute tasks across threads and/or machines
- Each thread carries out one function of code
- UI can add a task to the queue and the queue return an async Id back
- UI can then poll to see if results are ready
- Types
  - RabbitMQ
  - Beanstalk
  - Celery

Python Celery

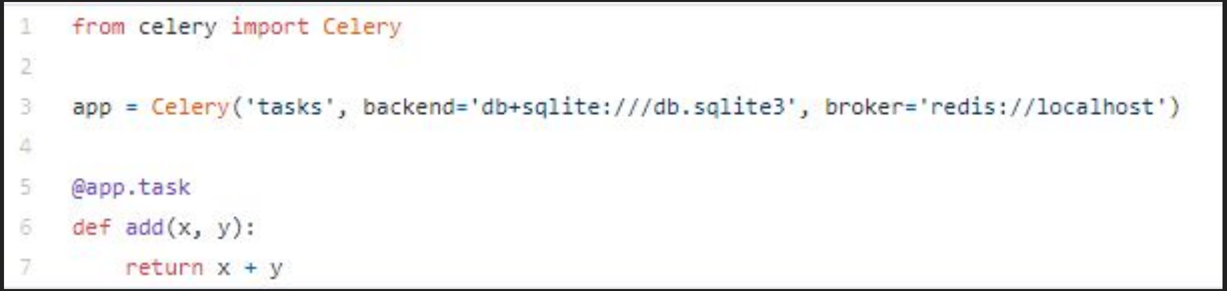
Why developers like Celery. StackShare. (n.d.). https://stackshare.io/celery.

Simple task queue that requires no configuration files

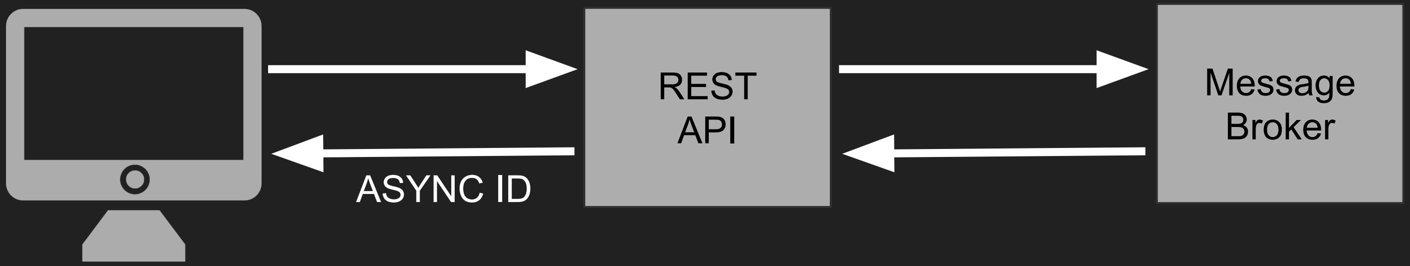
Can manage tasks across a single machine or multiple machines.

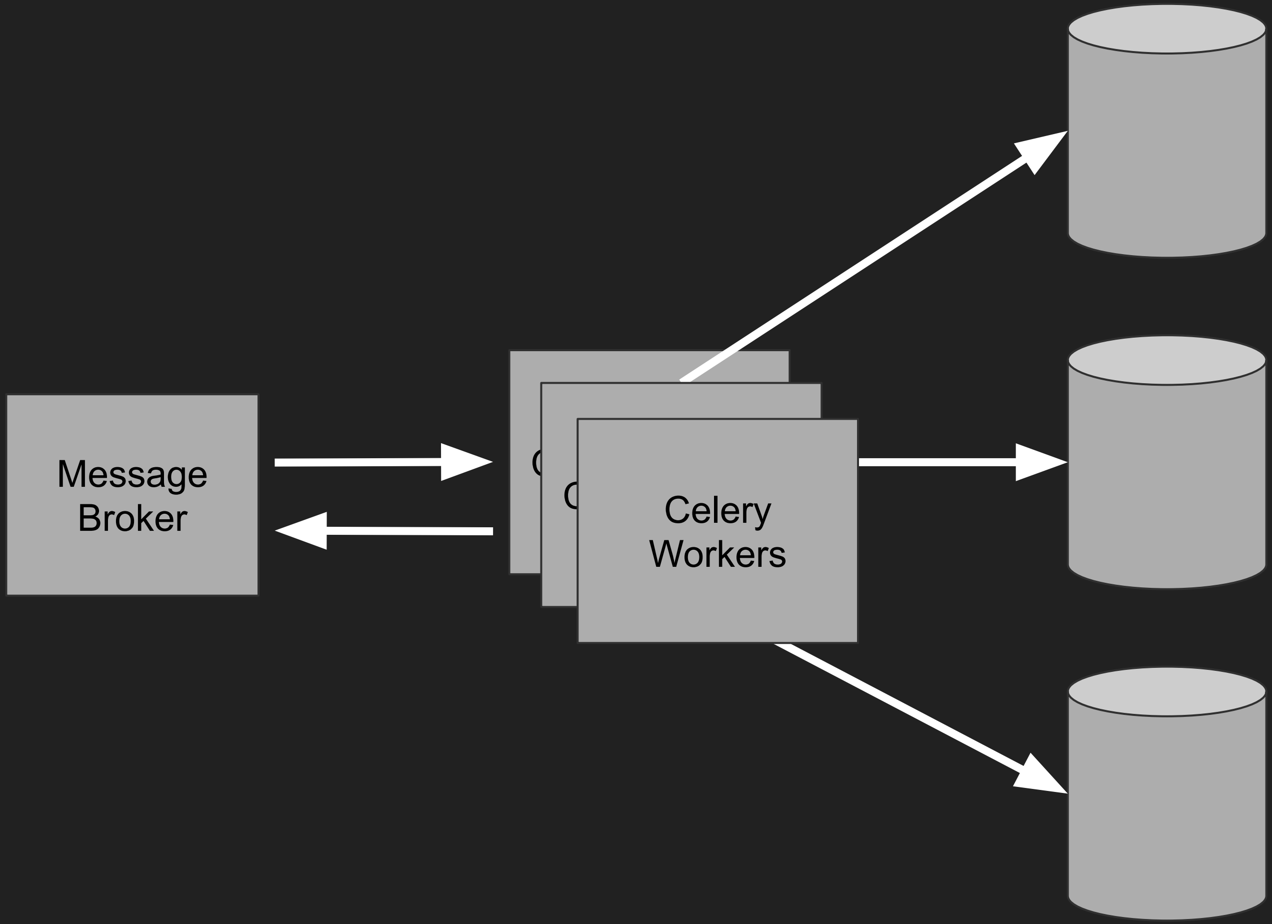
Easy to integrate with web frameworks such as Django and Flask

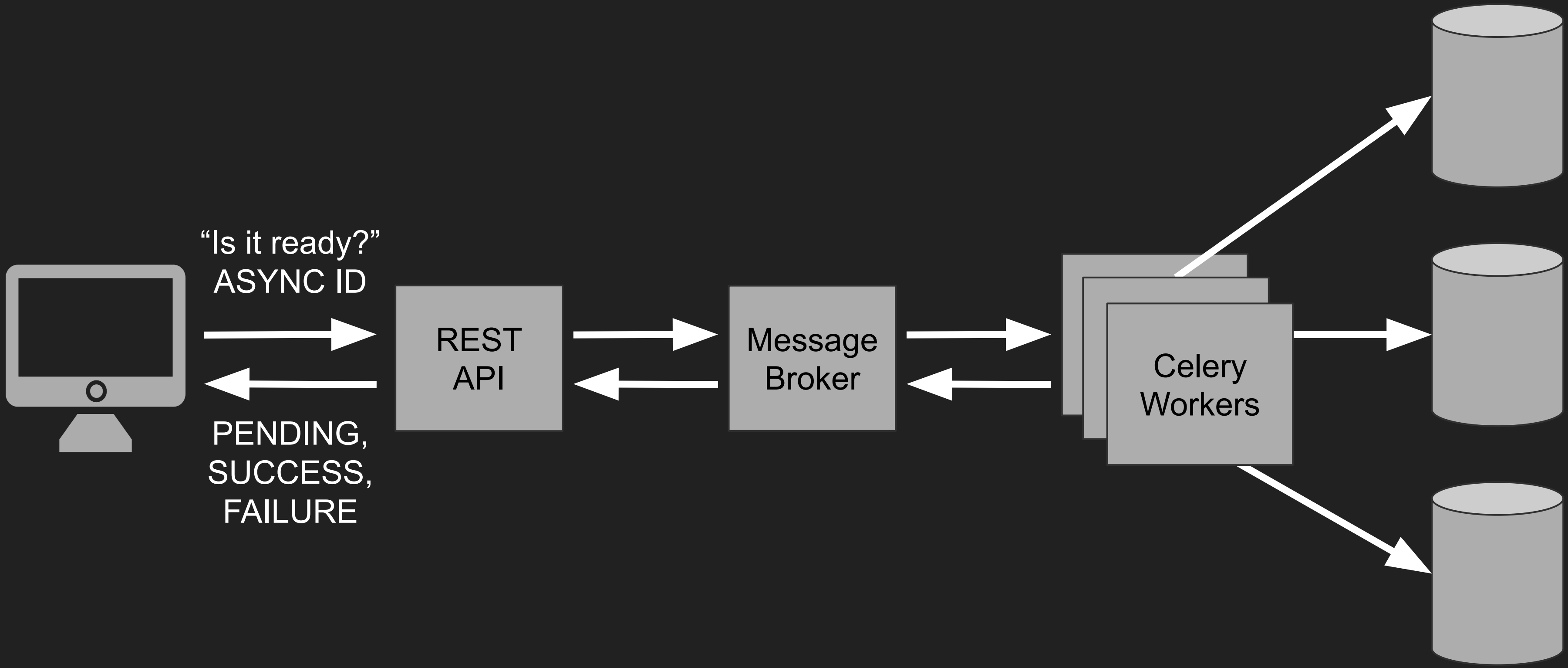
Has an active community



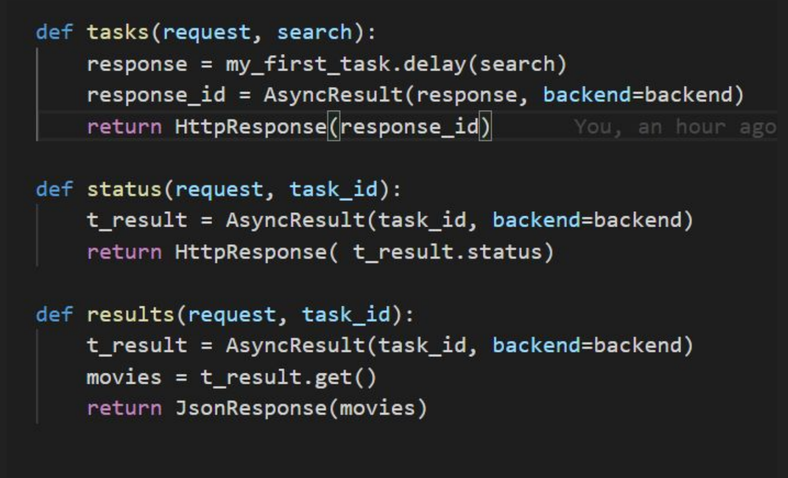
```python
from celery import Celery

app = Celery('tasks', backend='db+sqlite:///db.sqlite3', broker='redis://localhost')

@app.task
def add(x, y):
    return x + y
```


REST API
Message Broker
ASYNC ID

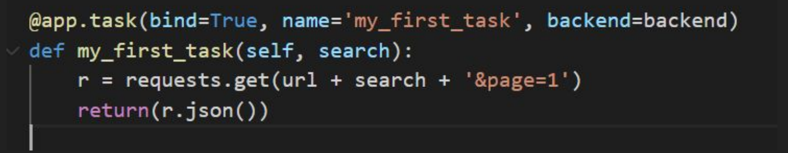

Message Broker
Celery Workers


"Is it ready?"
ASYNC ID
PENDING,
SUCCESS,
FAILURE
REST
API
Message
Broker
Celery
Workers

Video

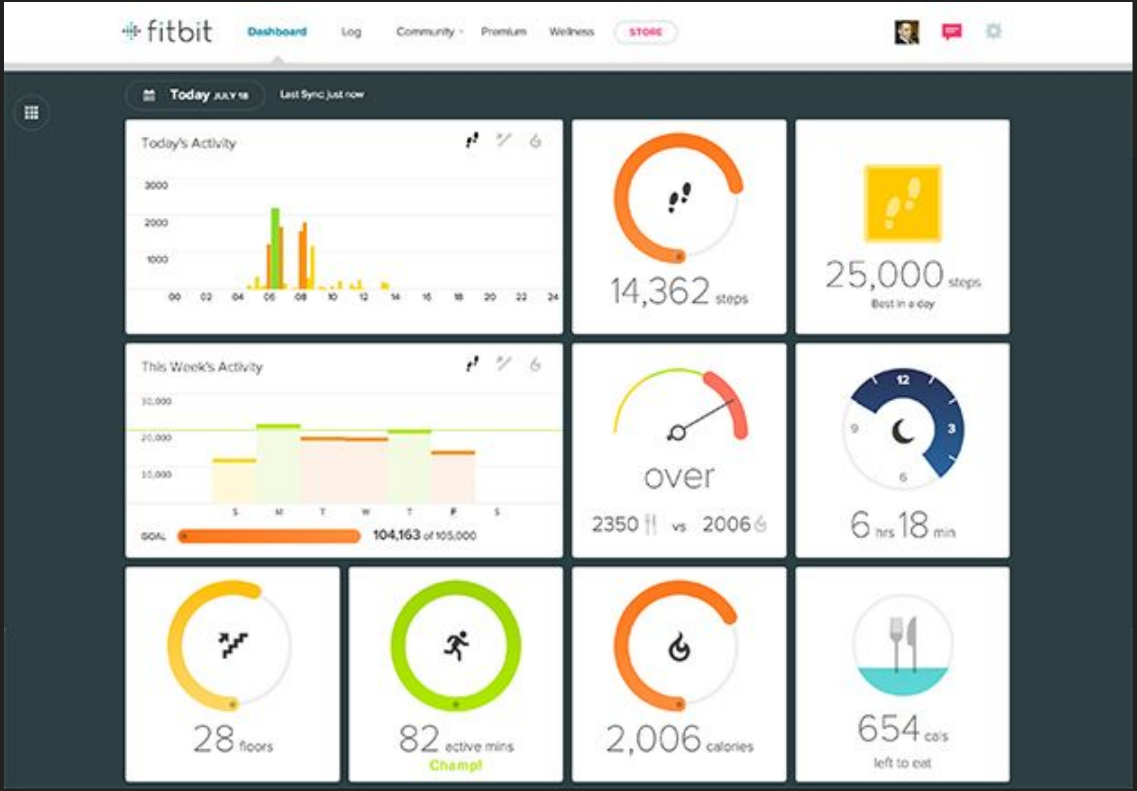


```
def tasks(request, search):
    response = my_first_task.delay(search)
    response_id = AsyncResult(response, backend=backend)
    return HttpResponse(response_id)        You, an hour ago

def status(request, task_id):
    t_result = AsyncResult(task_id, backend=backend)
    return HttpResponse( t_result.status)

def results(request, task_id):
    t_result = AsyncResult(task_id, backend=backend)
    movies = t_result.get()
    return JsonResponse(movies)
```



```
@app.task(bind=True, name='my_first_task', backend=backend)
def my_first_task(self, search):
    r = requests.get(url + search + '&page=1')
    return(r.json())
```

Dashboard



Staff, F. (2017, March 14). *Fitbit Dashboard Updated with Weekly Activity and More*. Fitbit Blog. https://blog.fitbit.com/fitbit-dashboard-updated-with-weekly-activity-and-more/.

Extras

Lots of additional configuration is available

Schedule tasks for different times

Celery workers can be spread across multiple servers to enable scaling with minimal setup

Can cache results in Redis to further increase speeds when duplicate queries are requested