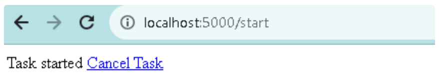
# Satya: Celery Revoke Analysis

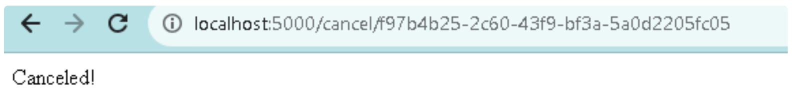
### What is the approach in code?

This is two point flask /start and /cancel, technically just need to go to /start and task will start and /cancel/<task_id> which is visible in start page.



```
localhost:5000/start
Task started Cancel Task
```

**Start Task**



```
localhost:5000/cancel/f97b4b25-2c60-43f9-bf3a-5a0d2205fc05
Canceled!
```

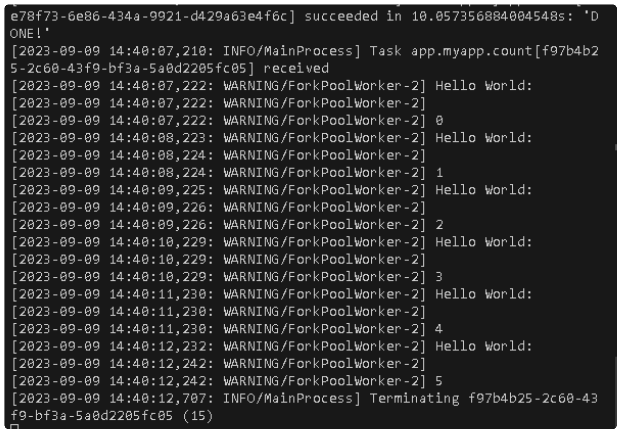
**Cancel Task**

---

1. Prefork:

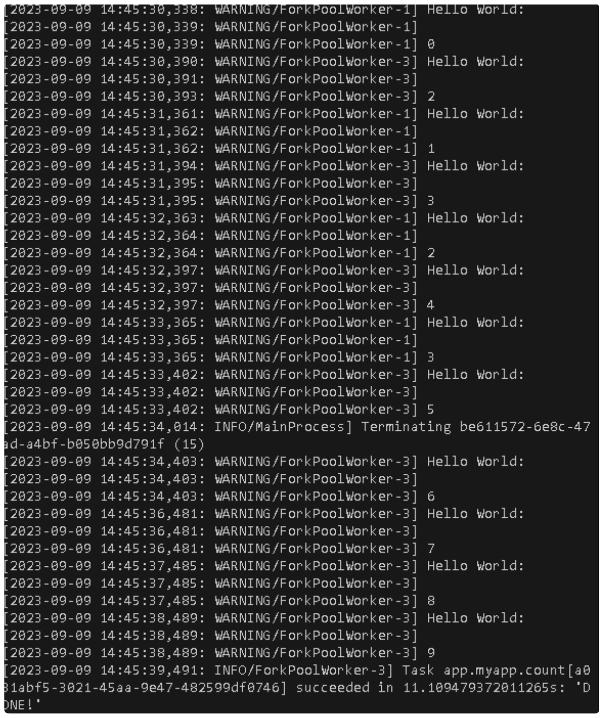
   Revoke is working correctly.

1 task only:



```
e78f73-6e86-434a-9921-d429a63e4f6c] succeeded in 10.057356884004548s: 'D
ONE!'
[2023-09-09 14:40:07,210: INFO/MainProcess] Task app.myapp.count[f97b4b2
5-2c60-43f9-bf3a-5a0d2205fc05] received
[2023-09-09 14:40:07,222: WARNING/ForkPoolWorker-2] Hello World:
[2023-09-09 14:40:07,222: WARNING/ForkPoolWorker-2]
[2023-09-09 14:40:07,222: WARNING/ForkPoolWorker-2] 0
[2023-09-09 14:40:08,223: WARNING/ForkPoolWorker-2] Hello World:
[2023-09-09 14:40:08,224: WARNING/ForkPoolWorker-2]
[2023-09-09 14:40:08,224: WARNING/ForkPoolWorker-2] 1
[2023-09-09 14:40:09,225: WARNING/ForkPoolWorker-2] Hello World:
[2023-09-09 14:40:09,226: WARNING/ForkPoolWorker-2]
[2023-09-09 14:40:09,226: WARNING/ForkPoolWorker-2] 2
[2023-09-09 14:40:10,229: WARNING/ForkPoolWorker-2] Hello World:
[2023-09-09 14:40:10,229: WARNING/ForkPoolWorker-2]
[2023-09-09 14:40:10,229: WARNING/ForkPoolWorker-2] 3
[2023-09-09 14:40:11,230: WARNING/ForkPoolWorker-2] Hello World:
[2023-09-09 14:40:11,230: WARNING/ForkPoolWorker-2]
[2023-09-09 14:40:11,230: WARNING/ForkPoolWorker-2] 4
[2023-09-09 14:40:12,232: WARNING/ForkPoolWorker-2] Hello World:
[2023-09-09 14:40:12,242: WARNING/ForkPoolWorker-2]
[2023-09-09 14:40:12,242: WARNING/ForkPoolWorker-2] 5
[2023-09-09 14:40:12,707: INFO/MainProcess] Terminating f97b4b25-2c60-43
f9-bf3a-5a0d2205fc05 (15)
```

2 tasks running ( i am running this to ensure that worker is not crashing only independent tasks are terminated)



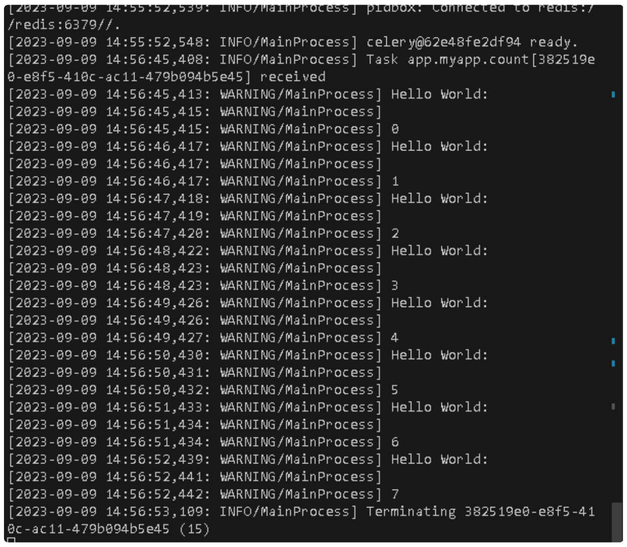
```
[2023-09-09 14:45:30,338: WARNING/ForkPoolWorker-1] Hello World:
[2023-09-09 14:45:30,339: WARNING/ForkPoolWorker-1]
[2023-09-09 14:45:30,339: WARNING/ForkPoolWorker-1] 0
[2023-09-09 14:45:30,390: WARNING/ForkPoolWorker-3] Hello World:
[2023-09-09 14:45:30,391: WARNING/ForkPoolWorker-3]
[2023-09-09 14:45:30,393: WARNING/ForkPoolWorker-3] 2
[2023-09-09 14:45:31,361: WARNING/ForkPoolWorker-1] Hello World:
[2023-09-09 14:45:31,362: WARNING/ForkPoolWorker-1]
[2023-09-09 14:45:31,362: WARNING/ForkPoolWorker-1] 1
[2023-09-09 14:45:31,394: WARNING/ForkPoolWorker-3] Hello World:
[2023-09-09 14:45:31,395: WARNING/ForkPoolWorker-3]
[2023-09-09 14:45:31,395: WARNING/ForkPoolWorker-3] 3
[2023-09-09 14:45:32,363: WARNING/ForkPoolWorker-1] Hello World:
[2023-09-09 14:45:32,364: WARNING/ForkPoolWorker-1]
[2023-09-09 14:45:32,364: WARNING/ForkPoolWorker-1] 2
[2023-09-09 14:45:32,397: WARNING/ForkPoolWorker-3] Hello World:
[2023-09-09 14:45:32,397: WARNING/ForkPoolWorker-3]
[2023-09-09 14:45:32,397: WARNING/ForkPoolWorker-3] 4
[2023-09-09 14:45:33,365: WARNING/ForkPoolWorker-1] Hello World:
[2023-09-09 14:45:33,365: WARNING/ForkPoolWorker-1]
[2023-09-09 14:45:33,365: WARNING/ForkPoolWorker-1] 3
[2023-09-09 14:45:33,402: WARNING/ForkPoolWorker-3] Hello World:
[2023-09-09 14:45:33,402: WARNING/ForkPoolWorker-3]
[2023-09-09 14:45:33,402: WARNING/ForkPoolWorker-3] 5
[2023-09-09 14:45:34,014: INFO/MainProcess] Terminating be611572-6e8c-47
ad-a4bf-b050bb9d791f (15)
[2023-09-09 14:45:34,403: WARNING/ForkPoolWorker-3] Hello World:
[2023-09-09 14:45:34,403: WARNING/ForkPoolWorker-3]
[2023-09-09 14:45:34,403: WARNING/ForkPoolWorker-3] 6
[2023-09-09 14:45:36,481: WARNING/ForkPoolWorker-3] Hello World:
[2023-09-09 14:45:36,481: WARNING/ForkPoolWorker-3]
[2023-09-09 14:45:36,481: WARNING/ForkPoolWorker-3] 7
[2023-09-09 14:45:37,485: WARNING/ForkPoolWorker-3] Hello World:
[2023-09-09 14:45:37,485: WARNING/ForkPoolWorker-3]
[2023-09-09 14:45:37,485: WARNING/ForkPoolWorker-3] 8
[2023-09-09 14:45:38,489: WARNING/ForkPoolWorker-3] Hello World:
[2023-09-09 14:45:38,489: WARNING/ForkPoolWorker-3]
[2023-09-09 14:45:38,489: WARNING/ForkPoolWorker-3] 9
[2023-09-09 14:45:39,491: INFO/ForkPoolWorker-3] Task app.myapp.count[a0
31abf5-3021-45aa-9e47-482599df0746] succeeded in 11.109479372011265s: 'D
ONE!'
```

---

2. Eventlet:

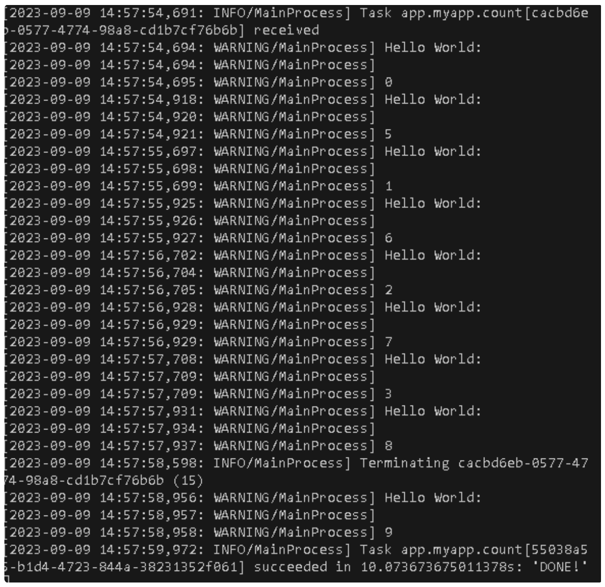
   Revoke is working correctly.

1 task only:



```
[2023-09-09 14:55:52,539: INFO/MainProcess] pidbox: Connected to redis:/
/redis:6379//.
[2023-09-09 14:55:52,548: INFO/MainProcess] celery@62e48fe2df94 ready.
[2023-09-09 14:56:45,408: INFO/MainProcess] Task app.myapp.count[382519e
0-e8f5-410c-ac11-479b094b5e45] received
[2023-09-09 14:56:45,413: WARNING/MainProcess] Hello World:
[2023-09-09 14:56:45,415: WARNING/MainProcess]
[2023-09-09 14:56:45,415: WARNING/MainProcess] 0
[2023-09-09 14:56:46,417: WARNING/MainProcess] Hello World:
[2023-09-09 14:56:46,417: WARNING/MainProcess]
[2023-09-09 14:56:46,417: WARNING/MainProcess] 1
[2023-09-09 14:56:47,418: WARNING/MainProcess] Hello World:
[2023-09-09 14:56:47,419: WARNING/MainProcess]
[2023-09-09 14:56:47,420: WARNING/MainProcess] 2
[2023-09-09 14:56:48,422: WARNING/MainProcess] Hello World:
[2023-09-09 14:56:48,423: WARNING/MainProcess]
[2023-09-09 14:56:48,423: WARNING/MainProcess] 3
[2023-09-09 14:56:49,426: WARNING/MainProcess] Hello World:
[2023-09-09 14:56:49,426: WARNING/MainProcess]
[2023-09-09 14:56:49,427: WARNING/MainProcess] 4
[2023-09-09 14:56:50,430: WARNING/MainProcess] Hello World:
[2023-09-09 14:56:50,431: WARNING/MainProcess]
[2023-09-09 14:56:50,432: WARNING/MainProcess] 5
[2023-09-09 14:56:51,433: WARNING/MainProcess] Hello World:
[2023-09-09 14:56:51,434: WARNING/MainProcess]
[2023-09-09 14:56:51,434: WARNING/MainProcess] 6
[2023-09-09 14:56:52,439: WARNING/MainProcess] Hello World:
[2023-09-09 14:56:52,441: WARNING/MainProcess]
[2023-09-09 14:56:52,442: WARNING/MainProcess] 7
[2023-09-09 14:56:53,109: INFO/MainProcess] Terminating 382519e0-e8f5-41
0c-ac11-479b094b5e45 (15)
```

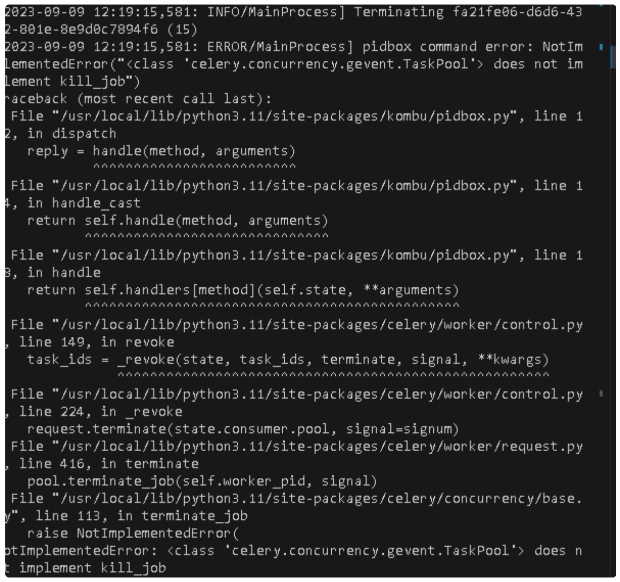
2 tasks :



```
[2023-09-09 14:57:54,691: INFO/MainProcess] Task app.myapp.count[cacbd6e
b-0577-4774-98a8-cd1b7cf76b6b] received
[2023-09-09 14:57:54,694: WARNING/MainProcess] Hello World:
[2023-09-09 14:57:54,694: WARNING/MainProcess]
[2023-09-09 14:57:54,695: WARNING/MainProcess] 0
[2023-09-09 14:57:54,918: WARNING/MainProcess] Hello World:
[2023-09-09 14:57:54,920: WARNING/MainProcess]
[2023-09-09 14:57:54,921: WARNING/MainProcess] 5
[2023-09-09 14:57:55,697: WARNING/MainProcess] Hello World:
[2023-09-09 14:57:55,698: WARNING/MainProcess]
[2023-09-09 14:57:55,699: WARNING/MainProcess] 1
[2023-09-09 14:57:55,925: WARNING/MainProcess] Hello World:
[2023-09-09 14:57:55,926: WARNING/MainProcess]
[2023-09-09 14:57:55,927: WARNING/MainProcess] 6
[2023-09-09 14:57:56,702: WARNING/MainProcess] Hello World:
[2023-09-09 14:57:56,704: WARNING/MainProcess]
[2023-09-09 14:57:56,705: WARNING/MainProcess] 2
[2023-09-09 14:57:56,928: WARNING/MainProcess] Hello World:
[2023-09-09 14:57:56,929: WARNING/MainProcess]
[2023-09-09 14:57:56,929: WARNING/MainProcess] 7
[2023-09-09 14:57:57,708: WARNING/MainProcess] Hello World:
[2023-09-09 14:57:57,709: WARNING/MainProcess]
[2023-09-09 14:57:57,709: WARNING/MainProcess] 3
[2023-09-09 14:57:57,931: WARNING/MainProcess] Hello World:
[2023-09-09 14:57:57,934: WARNING/MainProcess]
[2023-09-09 14:57:57,937: WARNING/MainProcess] 8
[2023-09-09 14:57:58,598: INFO/MainProcess] Terminating cacbd6eb-0577-47
74-98a8-cd1b7cf76b6b (15)
[2023-09-09 14:57:58,956: WARNING/MainProcess] Hello World:
[2023-09-09 14:57:58,957: WARNING/MainProcess]
[2023-09-09 14:57:58,958: WARNING/MainProcess] 9
[2023-09-09 14:57:59,972: INFO/MainProcess] Task app.myapp.count[55038a5
5-b1d4-4723-844a-38231352f061] succeeded in 10.073673675011378s: 'DONE!'
```

---

3. Gevent:
    a. Revoke is not working.
    b. Its not implemented.



```
2023-09-09 12:19:15,581: INFO/MainProcess] Terminating fa21fe06-d6d6-43
2-801e-8e9d0c7894f6 (15)
2023-09-09 12:19:15,581: ERROR/MainProcess] pidbox command error: NotIm
lementedError("<class 'celery.concurrency.gevent.TaskPool'> does not im
lement kill_job")
raceback (most recent call last):
 File "/usr/local/lib/python3.11/site-packages/kombu/pidbox.py", line 1
2, in dispatch
   reply = handle(method, arguments)
           ^^^^^^^^^^^^^^^^^^^^^^^^^
 File "/usr/local/lib/python3.11/site-packages/kombu/pidbox.py", line 1
4, in handle_cast
   return self.handle(method, arguments)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
 File "/usr/local/lib/python3.11/site-packages/kombu/pidbox.py", line 1
3, in handle
   return self.handlers[method](self.state, **arguments)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
 File "/usr/local/lib/python3.11/site-packages/celery/worker/control.py
, line 149, in revoke
   task_ids = _revoke(state, task_ids, terminate, signal, **kwargs)
              ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
 File "/usr/local/lib/python3.11/site-packages/celery/worker/control.py
, line 224, in _revoke
   request.terminate(state.consumer.pool, signal=signum)
 File "/usr/local/lib/python3.11/site-packages/celery/worker/request.py
, line 416, in terminate
   pool.terminate_job(self.worker_pid, signal)
 File "/usr/local/lib/python3.11/site-packages/celery/concurrency/base.
y", line 113, in terminate_job
   raise NotImplementedError(
otImplementedError: <class 'celery.concurrency.gevent.TaskPool'> does n
t implement kill_job
```

---

Gevent and Eventlet are both concurrency libraries for Python that allow you to write asynchronous and non-blocking code. They provide alternatives to the traditional thread-based or process-based concurrency models. Here, I'll explain the differences between Gevent and Eventlet in detail, including their underlying mechanisms, thread usage, and other aspects.

**1. Underlying Mechanism:**

- **Gevent:** Gevent is built on top of the `greenlet` library and uses cooperative multitasking. It employs a cooperative concurrency model, meaning that it relies on coroutines (greenlets) and does not create a new thread or process for each task. Greenlets yield control voluntarily to other greenlets, allowing non-blocking I/O operations.
- **Eventlet:** Eventlet also uses a cooperative multitasking approach, but it's built on the `greenthread` library. Like Gevent, it relies on coroutines (greenthreads) for concurrency. It employs a similar cooperative model but uses different internal mechanisms than Gevent.

**2. Threading:**

- **Gevent:** Gevent can be thought of as a higher-level library that provides a simple and Pythonic API for asynchronous programming. It manages its own greenlet pool. It may use a small number of OS-level threads (by default, just one) to manage these greenlets but doesn't create an OS-level thread for each greenlet.
- **Eventlet:** Eventlet also employs a cooperative model but doesn't use OS-level threads for managing greenthreads. Instead, it relies on a single OS-level thread (by default) and uses non-blocking socket operations for concurrency. Eventlet is known for its simplicity and lightweight approach.

**3. Compatibility:**

- **Gevent:** Gevent is compatible with a broader range of third-party libraries and frameworks due to its more extensive monkey-patching capabilities. It can patch many standard libraries and third-party modules to work cooperatively.
- **Eventlet:** Eventlet is more conservative in its monkey-patching and may require additional effort to work with certain libraries and frameworks. While this can be an advantage for code isolation, it might require more manual patching for compatibility.

**4. Performance:**

- **Gevent:** Gevent is known for its high performance, especially for I/O-bound tasks. It efficiently manages thousands of greenlets with a small number of OS-level threads, reducing the overhead of thread management.
- **Eventlet:** Eventlet also offers good performance for I/O-bound tasks, but its performance characteristics may differ slightly from Gevent. Performance may depend on the specific use case and the libraries being used.

**5. Community and Ecosystem:**

- **Gevent:** Gevent has a larger user base and a more active community, which results in more frequent updates, bug fixes, and third-party extensions. It has been widely adopted in various projects.
- **Eventlet:** Eventlet has a smaller community compared to Gevent. While it's stable and reliable, it may not have the same level of community support and ecosystem as Gevent.

**6. Licensing:**

- **Gevent:** Gevent is distributed under the MIT License, which is very permissive and allows for commercial and open-source use.
- **Eventlet:** Eventlet is distributed under the MIT License as well, making it suitable for both commercial and open-source projects.

In summary, both Gevent and Eventlet offer cooperative concurrency models and are suitable for I/O-bound tasks. The choice between them often depends on factors like library compatibility, project requirements, and personal preference. Gevent is more widely used and has a larger ecosystem, while Eventlet is known for its simplicity and lightweight design. The choice between them should be based on your specific project needs and constraints.

## Our use case is strictly not possible in Gevent.