Chapter - 7 : Managing Queues in Celery

The broker `RabbitMQ` is all about queues and is capable of having more than one queues, just like any other message broker out there one queue is hardly sufficient for any production level software

It may be possible that we want to separate out queues for various tasks in celery for reasons including not limited to better management, load balancing and security and the good news is that we can create multiple queues in `celery` and corresponding queues in `RabbitMQ` handle different different tasks in different queue

But what we're doing till now? Which Queue was that?

The default queue name is `Celery`, which we're using all the time

Here is the command to check all the active queues in the celery

```
celery –A task_one inspect active_queues
```

Creating named queues in Celery

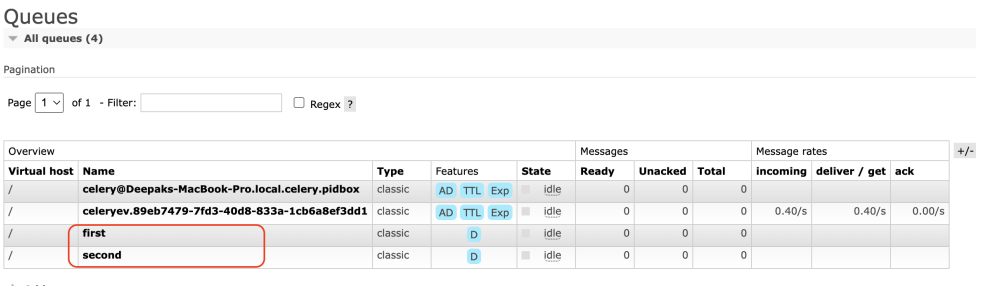
We can create named queues in celery while instantiating the workers

```
celery –A task_one worker ––loglevel=INFO –Q first
```

We can create multiple queues by giving a comma separated queue names

```
celery –A task_one worker ––loglevel=INFO –Q first,second,third
```

we can also see the same in the `RabbitMQ` Dashboard



Task In Queue

Submitting a task in a particular queue

We can use the `apply_async` to put a task in particular queue. For example, if we want to put a task in a queue name first, here is how we're going to call the function

```python
def put_task_in_first_queue():
    result = task_send_sms.apply_async((12345, "USD $5000 Deposited"), queue="first")
    result.get()

def put_task_in_second_queue():
    result = task_send_whatsapp.apply_async((12345, "USD $5000 Deposited"), queue="second")
    result.get()


if __name__ == "__main__":
    put_task_in_first_queue()
    put_task_in_second_queue()
```

Task In Queue

Example : 1

Changing the Default Queue of the Celery Tasks

When we created workers by specifying the queue name, the default queue named `celery` *no longer exists. In this case we need to explicitly mention the name of a queue while calling a task*

in the example above, if we remove the name of the queue in the `task_send_sms` *or* `task_send_whatsapp` *then the task will not be executed*

```
result = task\_send\_sms.apply\_async((12345, "USD $5000 Deposited"))
```

Example : 2

By default a queue name `Celery` *is created and the serves as a default queue. However, when we create named queues then we need to decide on which one is the default so that all the tasks where the queue name is not mentioned or called using* `delay` *can be forwarded to the default queue*

```python
app = Celery('cel_main', backend='rpc://', broker='pyamqp://')

# Mark the default queue Name
app.conf.task_default_queue = "first"
```

Task In Queue

Task Routing in Celery

We can create manual routes in celery and we can do the same by providing the queue parameter in the `app.Task` *method. In the example below, we've created 3 queues*

```
celery -A task_one worker --loglevel=INFO -Q first,second,third
```

```
app = Celery('cel_main', backend='rpc://', broker='pyamqp://')

#app.conf.task_default_queue = "first"


@app.task (queue='second')
def second_task():
    print("Second Task - CALLED")
    print("Second Task - DONE")
    return "Second Task Done."

@app.task (queue='third')
def third_task():
    print("Third Task - CALLED")
    print("Third Task - DONE")
    return "Third Task Done."

@app.task(queue='first')
def first_task():
    print("First Task - CALLED")
    print("First Task - DONE")
    return "First Task Done."
```

Example : 3