## Topic: Custom Management Command 8: Importing Large Data Using Celery & Reddis

Speaker: Udemy Instructor Rathan Kumar | Notebook: Django: Automating Common Tasks



1. After we installed REDIS, we installed REDIS as a package in our VIRTUAL ENVIRONMENT.



2. You installed CELERY.

- \$ pip install celery
- 3. Run this to create a Celery WOrker:

## windows:

```
$ celery -A autocommontasks_main worker --loglevel=info --pool=solo
```

macos:

 $\$  celery -A autocommontasks\_main worker --loglevel=info

This results to this with an error:

Rosilie@DELL MINGW64 C:/Users/Rosilie/AppData/Local/Programs/Microsoft VS Code (main)	
\$ celery -A autocommontasks_main workerloglevel=infopool=solo	
[2024-08-13 15:52:03,052: WARNING/MainProcess] No hostname was supplied. Reverting to default 'localhost'	
[tasks] . autocommontasks_main.celery.debug_task	
[2024-08-13 15:52:03,097: WARNING/MainProcess] C:\Users\Rosilie\AppData\Local\Programs\Python\Python39\lib\ _retry configuration setting will no longer determine whether broker connection retries are made during startup in Celery 6.0 and above. If you wish to retain the existing behavior for retrying connections on startup, you should set broker_connection_retry_on_startup to True. warnings.warn(	site-packages\celery\worker\consumer\consumer.py:508: CPendingDeprecationWarning: The broker_cor
[2024-08-13 15:52:05,136: ERROR/MainProcess] consumer: Cannot connect to amqp://guest:**@127.0.0.1:5672//: Trying again in 2.00 seconds (1/100)	[WinError 10061] No connection could be made because the target machine actively refused it.
<pre>[2024-08-13 15:52:09,168: ERROR/MainProcess] consumer: Cannot connect to amqp://guest:**@127.0.0.1:5672//: Trying again in 4.00 seconds (2/100)</pre>	[WinError 10061] No connection could be made because the target machine actively refused it.
[2024-08-13 15:52:15,252: ERROR/MainProcess] consumer: Cannot connect to amqp://guest:**@127.0.0.1:5672//: Trying again in 6.00 seconds (3/100)	[WinError 10061] No connection could be made because the target machine actively refused it.
<pre>[2024-08-13 15:52:23,339: ERROR/MainProcess] consumer: Cannot connect to amqp://guest:**@127.0.0.1:5672//: Trying again in 8.00 seconds (4/100)</pre>	[WinError 10061] No connection could be made because the target machine actively refused it.
[2024-08-13 15:52:33,452: ERROR/MainProcess] consumer: Cannot connect to amqp://guest:**@127.0.0.1:5672//: Trying again in 10 00 coronec (C/100)	[WinError 10061] No connection could be made because the target machine actively refused it.

## 4. To correct this, we have to update our SETTINGS.PY

4		$\leftarrow \rightarrow$			onTasks		Π
	EXPLORER ···	🔹 settings.py M 🗙	alerts.html	🕏 models.py uploads	🕏 admin.py uploads	🗘 importdata.html	\$
4		autocommontasks_m	ain > 🍦 settings.py				
Ŀ	✓ autocommontasks_m		ssage config			LCESS messages	
	>pycache	134 MESSA	AGE_TAGS = 🕇	[			
$\mathcal{A}$	🕏initpy 🛛 M	135 ···· <b>n</b>	essages.ERF	ROR: "danger",			
_	🍨 asgi.py	136	0. "critica	al".			
j 🍰	🔹 celery.py 🛛 U	- 107 ]		ر ـــ			
	🗬 settings.py 🛛 M	12/ }					
	🔹 urls.py	138					
~	🕏 views.py	139 #•Cel	ery-related.	<pre>configuration;</pre>	; setting the me	essage broker	
	🕏 wsgi.py	140 CELER	Y BROKER UF	RL = 'redis://lo	calhost:6379'		
LL2	$\checkmark$ dataentry	141					
	>pycache						
<u>–</u> 0	$\checkmark$ management \ commands						
π	>pycache						

In the bash terminal, press CTRL + C to terminate the process and try again. Your celery should be ready.



5. To use CELERY for testing in our project, create a new URL in our root project's URLS.PY

⋈		$\leftarrow \rightarrow$	P AutomatingCommon	Tasks	
$\equiv$	EXPLORER ····	g.py 🔹 .gitignore	🕏 insertdata.py 🛛 🕏 importdata.py	🕏 exportdata.py	🕏 urls.py autocommontasks_main M 🗙
<b>~</b>		autocommontasks_mai	in > 🔹 urls.py > 🕼 urlpatterns		
	<ul> <li>✓ autocommontasks_m</li> <li>&gt; _pycache</li> <li>✓ _initpy</li> <li>M</li> <li>asgi.py</li> <li>celery.py</li> <li>U</li> <li>settings.py</li> <li>w urls.py</li> <li>M</li> <li>views.py</li> <li>wsgi.py</li> </ul>	152. 16 """ 17 from c 18 from c 19 from c 20 from c 21 from c	Add a URL to urlpatterns: django.contrib import admin django.urls import path, in import views django.conf.urls.static imp django.conf import settings	ort static	, include('blog.urls')
	<ul> <li>dataentry</li> <li>_pycache</li> <li>management \ commands</li> </ul>	22 23 urlpat	tterns = [		
-	>pycache	24 <mark>- p</mark> a	ath('admin/', admin.site.ur	ls),	
<ul> <li>∠</li> <li>∠</li></ul>	<ul> <li>exportdata.py</li> <li>greeting.py</li> <li>helloworld.py</li> <li>importdata.py</li> </ul>	25 pa 26 # 27 pa 28 pa	ath('', views.home, name='h links to our dataentry app ath('', include('dataentry. ath('celery-test/' views ce	ome'), 's·URLS.PY urls')), lerv test)	
	<ul> <li>insertdata.py</li> <li>migrations</li> <li>initpy</li> <li>admin.py</li> <li>anps.py</li> </ul>	29 ] + st 30	tatic(settings.MEDIA_URL, d	ocument_root=s	ettings.MEDIA_ROOT)

We run our django server and att the url in our browser: http://127.0.0.1:8000/celery-test/



6. Now update our CELERY\_TEST FUNCTION to perform time-consuming operations.



So, when you reload your page, in the background, it will load 10 seconds (Sleep)

ALLA.									•					
~	<b>(</b> ) c	ourse: A	Automate	🗙 🛛 🎯 ChatGPT	×	Install Redis On W	×   🕄 Persona	al Digital N 🛛 🗙	RosyCodes/Djang ×	127.0.0.1:800	0/ce ×	+		_
←	$\rightarrow$	×	(i) ht	ttp://127.0.0.1:8000/celery-	test/						,	२ 🕁	þ	

Function executed successfully.

7. We want to pass then the SLEEP(10) TASK to celery, so that our project or user can do or see other things while Celery is working on something else. To do this, in the DATAENTRY folder, create a new file, TASKS.PY and update as



8. To test all this, LOAD your URL http://127.0.0.1:8000/celery-test/ again, click on your CELERY TERMINAL, and it should RECEIVE and DISPLAY the message 'TASK EXECUTED SUCCESSFULLY' while we wee our webpage with H3 tag showing the message right away.

n Course: Auto X 🚳 ChatGPT X 💌 Install Redis X 🖓 Personal D X 🗅 ResuCode		← → 9º AutomatingCommonTasks
	EXPLORER ····	♦ views.py autocommontasks_main M ×     ♦ tasks.py U     ♦ home.html     ♦ urls.py dataentry     ♦ views.py dataentry     ↓ views.py dataentry     ↓ ×      □ □
· · · · · · · · · · · · · · · · · · ·	✓ AUTOMATINGCOMMONTASS     ✓ AUTOCOMMONTASS (m 0     ) 0     / 0     / 0     / 0     / 0     / 0     / 0     //	<pre>autocommontaks_main &gt; @ viewspy &gt; 1 from django.shortcuts.import render 2 from django.http.import HttpResponse 3 # *imports:the function.from the TASKS.PY.of DATAENTRY folder 4 from dataentry.tasks import celery_test_task 5 6 7 def home(request): 8  ····return render(request, 'home.html') 9 14 # celery_test 15 def celery_test(request): 15  ····# execute.timeconsumin.task-here.for.celery.execution 16  ···celery_test_task.delay() :# calls:the celery_test_task function.ar 17  ···# execute.the H3 message displayed right.away. 18  ····return HttpResponse('<h3>Function executed successfully.</h3>') 18 19 10 10 10 11 11 11 11 11 11 12 12 13 13 14 14 15 15 15 15 15 15 15 15 15 15 15 15 15</pre>
@	enployee_1M.cv  results resul	PROBLEMS         OUTPUT         DEBUG CONSOLE         TERMINAL         PORTS         AZURE         Image         PORTS         PORTS </td

## IMPORTANT REMINDER:

So, when you run your django project with celery and redis, THERE SHOULD BE 3 BASH TERMINALS AND YOU NEED TO NAME THEM APPROPRIATELY:

- 1. DJANGO-SERVER- this is where you will run your python server to run your Django project:
- \$ python manage.py runserver



2. REDIS - this is where you run your REDIS-SERVER. Every time you start this, it will create a new process of REDIS, so you can simply kill it or ignore it. As long as when you PING your REDIS SERVER, it returns a PONG message, then you are fine. You have multiple processes (TCP) here because you have started your REDIS-SERVER multiple times.

\$ redis-server

- \$ netstat -ano | findstr :6379 (this is to see what process is listening to the port 6379 (Redis)
- \$ redis-cli ping (Run this before you run the REDIS-SERVER so you wont have several TCPs)

PROBLEMS OUTPUT		IAL PORTS AZ	ZURE			+~ ··· ^ ×
[26052] 13 Aug 17:16	:42.613 # Could not o	reate server TC	P listening soc	ket *:6379: bind	: An operation was	▶ django-server 🔬
attempted on somethi	ng that is not a sock	et.				▶ redis Microsoft VS C
(env)						Celery Microsoft VS C
[26052] 13 Aug 17:16	:42.613 # Could not o	reate server TC	P listening soc	ket *:6379: bind	1: An operation was	
Eacompted on something 17:16	ng that is not a sock	et.	D listoning cos	kot * 6270, hin	. An openation was	
attempted on somethi	ng that is not a sock	ot	P IISCENING SOC	.Ket *:0379: D1nd	. An operation was	
	ING CHAC IS NOT A SUCK	cc.				
Rosilie@DELL MINGW64	C:/Users/Rosilie/Apr	Data/Local/Prog	rams/Microsoft	VS Code (main)		
<pre>\$ netstat -ano   fin</pre>	dstr :6379					
TCP 0.0.0.0:637	9 0.0.0.0:0	L	ISTENING	30012		
TCP [::]:6379	[::]:0	L	ISTENING	30012		
TCP [::1]:6379	[::1]:654	97 E	STABLISHED	30012		
TCP [::1]:6379	[::1]:654	98 E	STABLISHED	30012		
TCP [::1]:65497	[::1]:637	9 E	STABLISHED	16768		
TCP [::1]:65498	[::1]:637	9 E	STABLISHED	16768		
Posilio/DELL_MINCH64	Lu/Ucons/Posilio/Anr	Doto /Local /Drog	mame ///icnocoft	VE Code (main)		
<pre></pre>	osers/ kostite/ App	Data/Local/Prog	railis/futrosoft	vs coue (marn)		
PONG						
(env)	-					
Rosilie@DELL MINGW64	C:/Users/Rosilie/App	Data/Local/Prog	rams/Microsoft	VS Code (main)		
o \$ [						

3. CELERY- this is where you will run this code whenever you need to launch a new task or when you reload your Django page. Meaning, you have to launch this command again if you have made new changes to your CELERY TASKS like changing 10 seconds to 5 seconds to see the effect.

\*autocommontasks\_main - this is your Django Project Name

\$ celery -A autocommontasks\_main worker --loglevel=info --pool=solo



Copyright © Personal Digital Notebooks | By Rosilie | Date Printed: July 11, 2025, 1:42 a.m.