**Django Mail Queue**

Mail Queue provides an easy and simple way to send email. Each email is saved and queued up either in real time or with Celery.

### Download and install :

**pip3 install django-mail-queue**

**pip3 install django-celery**

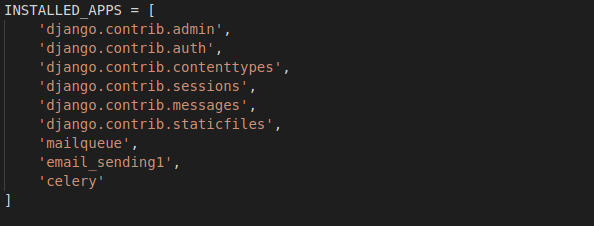
**pip3 install redis**

**Quickstart**

### Basic configuration:-

### First install the package into your project (see above).

Open settings.py and add in INSTALLED\_APPS:

Add the below settings, and adjust as needed:



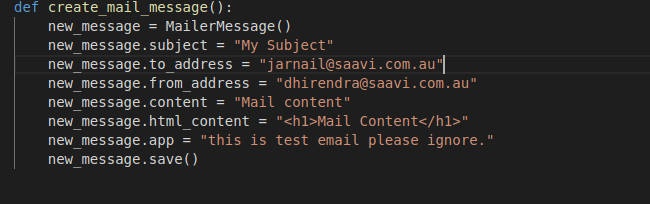
**Running the migrations**

Once you've added mailqueue to your INSTALLED\_APPS plus the basic config in settings.py, run the migrations to create the tables needed:

python3 manage.py migrate

## **To Send an Email**

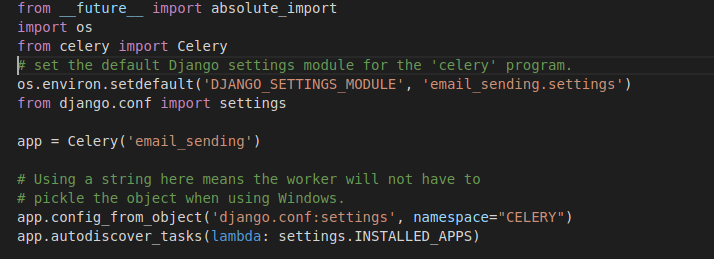
Create a new MailerMessage() object in task.py file



When save is called, Django will immediately try to send the email. Should it fail, it will be marked as unsent, and wait for the next time the job is called. Of course, the BCC address is optional.

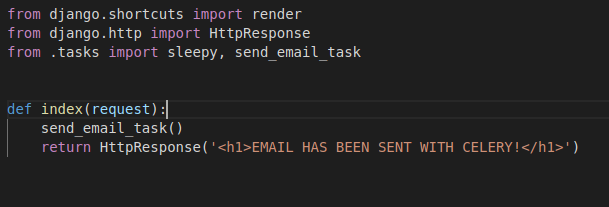
## **Celery**: [Celery](http://www.celeryproject.org/), a powerful asynchronous job queue used for running tasks in the background. Add celery.py file in main Django project and create tasks (task is responsible to send email or any other job).

celery.py



View.py

Add task here



**Redis:**

Redis is also requiered : "Redis is an open source (BSD licensed), in-memory data structure store, used as a database, cache, and message broker. Redis provides data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, hyperloglogs, geospatial indexes, and streams. Redis has built-in replication, Lua scripting, LRU eviction, transactions, and different levels of on-disk persistence, and provides high availability via Redis Sentinel and automatic partitioning with Redis Cluster."

Redis : here is show which email messgage is send or not and store all user details . """

