U4BMZ90T0 : You think it's best to create a `.env` file for the user and give an example or create one if one does not exist in the directory and write to it?
U1BP42MRS : The standard is `.*rc` files, so I usually use `~/.foorc`
U1BP42MRS : At least on linux
U5S8CNULD : <@U1BP42MRS> Yes, I need implement `between` too, but first need to implmenet for current day, 2 minutes, I'll try
U1BP42MRS : Well between start of day and end of day is "current date" - which is what I am getting at
U4BMZ90T0 : <@U0NRYQNAZ> was it you that had the `example.env` file?
U0NRYQNAZ : env.example ya
U0NRYQNAZ : or `.env.example` my bad.
U4BMZ90T0 : Is that in your gists?
U0NRYQNAZ : no but i can create one.
U0NRYQNAZ : `.env.example````
export PY_ENV="development"
export GITHUB_TOKEN=""
export SOME_API_URL=""
export SOME_API_TOKEN=""
export DB_USER=""
export DB_PASSWORD=""
export DB_HOST=""
export DB_NAME=""
```

U0NRYQNAZ : then you can run `cp .env.example .env`
U0NRYQNAZ : and fill out the values
U0NRYQNAZ : then run `source .env` and be good to go.
U5S8CNULD : <@U1BP42MRS> Okay, I'm tried.. but didn't work(

For example

```
return query.filter(
            and_(
                Post.created_at > search_by_date(),
                Post.created_at < get_today()
            )
        )
```

and methods
```
def search_by_date(days=None):
    if days:
        date = datetime.utcnow() - timedelta(days=int(days))
        return date.strftime('%Y-%m-%d')
    else:
        yesterday = datetime.utcnow() - timedelta(days=1)
        return yesterday


def get_today():
    today = datetime.utcnow().strftime('%Y-%m-%d')
    return today
```
But for current day, it's not working.. I understand in which way I need to move..:disappointed:

U1BP42MRS : I think you either want `< get_today() + 1day` or to do `<= get_today()`
U5S8CNULD : ```# Query

first didn't work
```

```
return query.filter(Post.created_at == get_today())

# second
# return posts after 25th July
return query.filter(Post.created_at &gt;= get_today())

def get_today():
    today = datetime.utcnow().strftime('%Y-%m-%d')
    return today
```

U610F0WKC : Hello,
I would like to test a system with several AMQP consumer/producers that are exchanging messages with each other.
I would like to have unit-testing with those consumers. Is there a commonly used design pattern for doing so in Python?
(I'm using pytest at the moment with a fixture that look like this:

U610F0WKC : At the moment my set-up is this:My consumers inherit from multiprocessing.Process and when started set-up their AMQP environment.
When the test is finished, the terminate closes the connection and cancel everything.
The problem that I get is that even when a test completed, the Process of the consumers seems to continue running in the background even after I called terminate.

How could I have a nice clean-up when the test is over? Would it change something to have BlockingConnection instead of a SelectConnection ?

Here is the current version of my base class for those components:
<https://gist.github.com/sieben/caf39e04a83c812550d967a22b9d8584> is an obvious flaw in it? Is there an alternative from inheriting from Thread or Process ?

U4BMZ90T0 : Is there any way to run pycharm in debug with commandline arguments?
U0LSCQQNR : yes
U1BP42MRS : Yeah, you can use the same run menu with command line args and click the debug icon instead of play - or are you seeing something funky?
U0LSCQQNR : go to run configuration
U4BMZ90T0 : Is it under the script parameters