U1BP42MRS : probably a flask devserver oddity

U5CGPBF0U : Well, that never would have occurred to me. I'll have to remember that.

U1BP42MRS : sorry man

U5CGPBF0U : Sorry? You solved my problem! :stuck_out_tongue: Thanks man!

U1BP42MRS : after spinning for an hour or so

U5CGPBF0U : Haha, no problem. It would've taken me alone at least 6 hours. :wink:

U5CGPBF0U : The only - quite odd - problem I'm having now is that all my endpoints are returning 404s. There are no Python errors reported whatsoever though. :thinking_face: I'm not sure what to make of it. I inserted some `print()`s to check everything was running in the order expected and it was.

U1BP42MRS : Can you give a sample of how you are registering your blueprints?

U5CGPBF0U : Um, no blueprints :sweat_smile:

U1BP42MRS : how are you registering routes

U5CGPBF0U : In __main__.py:```...

```
from broadway.external import api, auth
app = create_app(os.environ['APP_SETTINGS'])
class API_Root(Resource):
    def get(self):
        return 'This page intentionally left blank.'
api.add_resource(API_Root, '/')```
```

U5CGPBF0U : (Just one of many endpoints, but I chose the simplest one)

U5CGPBF0U : Changing the port number in `app.run(host='0.0.0.0', port=port)` has no effect. The server restarts on the default port. :thinking_face:

U69GQ27K8 : Hey guys, have some questions for you Django users.. Im a fairly novice web developer and have started a small project with Django. I need it to have an updated list of Netflix shows and movies at all times. Since Netflix doesn't have an open API, would the most logical thing to do would be to create a scraper that gets that information from their website? I am a bit stuck on what would be the proper way to do this. Thank you for your help!

U1BP42MRS : That's probably your only option if there's no unofficial source either. Be sure to check that you aren't breaking their terms of service with that, however. Scraping is often times okay, but some sites explicitly forbid it - especially if intending to resell the info

U0PKTUWKH : hello again, looking for help with a boto3 based script. I am getting en error ```

```
Traceback (most recent call last):
  File "backup-to-s3.py", line 58, in &lt;module&gt;
    upload = push_to_s3(args.bucket)
  File "backup-to-s3.py", line 36, in push_to_s3
    uploaded = s3.Bucket(bucket_name).upload_file(file_name)
TypeError: bucket_upload_file() takes at least 3 arguments (2 given)
```
```

this is code <https://gist.github.com/pleegor/ae8086286533ce33cc92095d7049dce5>

U0PKTUWKH : it appears im missing something in `s3.Bucket(bucket_name).upload_file(file_name)`

U0PKTUWKH : i think i sorted things out (please correct me if im wrong)  upload_file requires the following: `Filename, Bucket, Key, ExtraArgs=None, Callback=None, Config=None` and key was missing

U5VCDE5SS : anybody know how I can use jquery to target specific elements that Django automatically brings up?

U5VCDE5SS : specifically I'm trying to click on a button if the Django generic form error element is visible to the user, but targeting the element directly always brings up the modal that is triggered by the button, or doesn't trigger at all if I use the visible selector

U686SVA9X : Guys am having a problem deciding whether to use function or class base view with minimal angular or jquery for frontend

U686SVA9X : or start with function base then refactor

U0NRYQNAZ : it all depends on what you like.

U0NRYQNAZ : i prefer to stay as close to a function as possible before creating a class. however, there are great times to use a class and really depends on the use case. if you offer up some examples or ideas of what you are trying to do maybe someone has a way they have done it before.

U663SK5K2 : i don't understand the 2nd elif condition```

```
choice = int(input("Choose a number in the list: "))
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
new_list = []
for i in a:
    if choice not in a:
```

```
            print("Choice should be in the list.")
            break
        elif choice > i:
            my_list.append(i)
        else:
            print(my_list, ",\nelements containing this list is smaller than", choice)
            break
```