

U63VCB8AH : is this pythonic? <@U13L8J76J>

U5NMSURQAQ : `self.json = json` I think there is a better name for this parameter

U63VCB8AH : Or would you just use the `json` module without specifying it as a dep in the constructor?

U5NMSURQAQ : yes, this!

U63VCB8AH : why? just interested :slightly\_smiling\_face:

U63VCB8AH : because it is a built-in?

U5NMSURQAQ : `to\_json` only makes sense with json, and json is always available

U63VCB8AH : ```import json

```
class User():
    name = ""
    email = ""

    def __init__(self, name, email):
        self.name = name
        self.email = email

    def to_json(self):
        return json.dumps({
            'name': self.name,
            'email': self.email,
        })
...`
```

U63VCB8AH : like this?

U5NMSURQAQ : I think it's quite nice

U5NMSURQAQ : btw, you don't need```

```
    name = ""
    email = ""
...`
```

in your class definition

U5NMSURQAQ : `self.name` sets an attribute for every new instance already

U63VCB8AH : I know but I think it's more explicit

U63VCB8AH : What would I use if I would to serialize SQLAlchemy model result?

U63VCB8AH : would like to be able to encode the results as JSON or cast it to a dict

U63VCB8AH : Is `marshmallow` any good or is there anyway to it with pure python easily? Can I implement a special attribute like `\_\_json\_\_`?

U5NMSURQAQ : marshmallow is the tool to go

U63VCB8AH : what should I use for web dev?

U63VCB8AH : Flask? aiohttp?

U5NMSURQAQ : oh, it's up to you completely, there are too many options for one to be considered the only way :slightly\_smiling\_face:

U63VCB8AH : I'm searching for something that can handle async methods since I'm going to fetch data from external HTTP APIs

U63VCB8AH : Can I do async I/O in Flask? Would like to use async/await since I'm already familiar to that concept

U5NMSURQAQ : I have no experience with async :confused:But Flask can play nicely with WebSockets for example

U63VCB8AH : Hmm

U63VCB8AH : "However Flask is just not designed for large applications or asynchronous servers. Flask wants to make it quick and easy to write a traditional web application."

U1UFZTD4J : No you can't really do asyncio with flask.

U63VCB8AH : What happens if I try? :grin:

U1BP42MRS : There's some gevent based options for flask that makes it "async", but generally it's considered best to have a background worker do asnc things (celery or rq)

U63VCB8AH : so I would be better off to use aiohttp?

U1UFZTD4J : You can combine sync with async by using `loop.run\_until\_complete`. it will work locally, but once you run in thread mode (prod) you won't have an event loop.

U63VCB8AH : don't want to use any queuing

U1UFZTD4J : <@U63VCB8AH> if making lots of remote calls in parallel is required for returning the response, i would use an asyncio framework. Sanic is flasklike

U1UFZTD4J : Aiohttp is also a good option.

U63VCB8AH : Seems nice

U63VCB8AH : Then I just need to figure out how to get JWT auth working :smile:

U5NMSURQAQ : jwt-extended!

U63VCB8AH : that may work with the Sanic API

U13L8J76J : <@U63VCB8AH> better have `to\_dict` method, so you can serialize it into more than just json

U13L8J76J : Say, yaml

U2X3YC1U6 : I'm considering how to make this part python 2 friendly

U2X3YC1U6 : it works, but I have to do it in a couple of places in my script

U2X3YC1U6 : what is the proper way to make this code work in both python 2 and python 3?

U1BP42MRS : The quick and easy way is to just make a function that gets the basename from your string.

U6BM4GY0G : Hi! What TUI library would you guys recommend learning? I want to make GUIs for the terminal but I dont want any new windows opening

U5NMSURQAQ : easy or hard? :stuck\_out\_tongue:

U6BM4GY0G : Whatever really, as long as it gives me the oppertunity to develop some kind of gui for an application for the terminal :slightly\_smiling\_face:

U5NMSURQAQ : <<https://github.com/pfalcon/picotui>>

U5NMSURQAQ : There was another project like that but I can't find it!

U5NMSURQAQ : Very similar, with windows and common elements like checkboxes...

U5NMSURQAQ : If you want something more barebones, then good old `curses`, or `bearlibterminal` for a modern approach, though it's aimed at games

U6BM4GY0G : Thanks, will check them out!

U3UR8LD18 : ```Hi, I have a file with following contents

> 1234 alphabet /vag/one/arun

> 1454 bigdata /home/two/ogra

> 5684 apple /vinay/three/dire

but i want the output to be like

> 1234 alphabet one

> 1454 bigdata two

> 5684 apple three

```

U5NMSURQAQ : tee hee :slightly\_smiling\_face:

U5NMSURQAQ : `sed -r 's|(.+)/[^\|]+|([^\|]+)/.+|\1\2|g`

U6BM4GY0G : Even though I dont have anything to do with this question, could you explain the logic behind the answer? The formatting sentence seem so random

U5NMSURQAQ : `sed -r` is an extended mode, so that `+` is enabled (matches one or more characters, unlike `\*` that matches zero or more)

U5NMSURQAQ : `s///g` or `s|||g` or any symbol instead of `|` is how a basic replacing expression is constructed. The first field is what to match, the second is what to replace it with.

U5NMSURQAQ : `(.+)/[^\|]+|([^\|]+)/.+` matches anything from the start until the first `^` and puts found characters in the first group (`\1`)

U5NMSURQAQ : `[^\|]+/` matches anything that is not a slash, and then a slash (`vag/` or `home/`)

U5NMSURQAQ : `([^\|]+)/` matches the same thing, but puts the stuff found in-between slashes in the second group `\2`

U5NMSURQAQ : and then `.+` matches whatever comes next to the end of line