Python Fun

Things to ask when designing a restful api

- Domains / Resources
- What kind of actions (verbs) should be applied to which domain
- URI Paths for resources that combines the domain with action
- What is the input for every domain action
- What is the output for every domain action
- It is a function by the end of the day
- Always think in MVC to make your life easier to visualize ur app

Things to do after

- Create project layout
- Start mocking the API
- Add API doc
- Create endpoint with empty implementation under controllers one function at a time
- Create models for the request and the response if needed
- Create repository for the controller one function at a time
- Connect the controller to the repo
- Test
- Repeat until you are done

TODO restful API Design

Notes

- APIDoc
- Swagger

Flask

- What is flask
- Why to use flask
- When to use flask
- Who to use flask (kidding)
- Let us do some python by combining what we didn’t learn

Falsk Basics

- Micro framework
- WSGI (Web Server Gateway Interface)
- PEP 3333

Why Flask

- Tool an lib agnostic you can use whatever lib alongside the web app
- The possibility to use flask extensions (ORM, Auth, Validation...etc)
- Development server and debugger
- Integrated support for unit testing
- Restful request dispatching
- Jinja
- Support for secure cookies (client side sessions)
- Amazing Doc

First app

```
from flask import Flask

app = Flask(__name__)

app.run(debug=True)
```

Routes

```python
from flask import Flask

app = Flask(__name__)


@app.route('/')
def hello():
    return "hello there"


app.run(debug=True)
```

Some More

- Decorators @
- URL Variables */post/<int:post_id>*
- Templates return render_template('template', **kwargs)
- Form and Request Object

Additional docs

- https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world
- https://tedboy.github.io/flask/index.html
- https://flask.palletsprojects.com/en/2.0.x/quickstart/#a-minimal-application
-