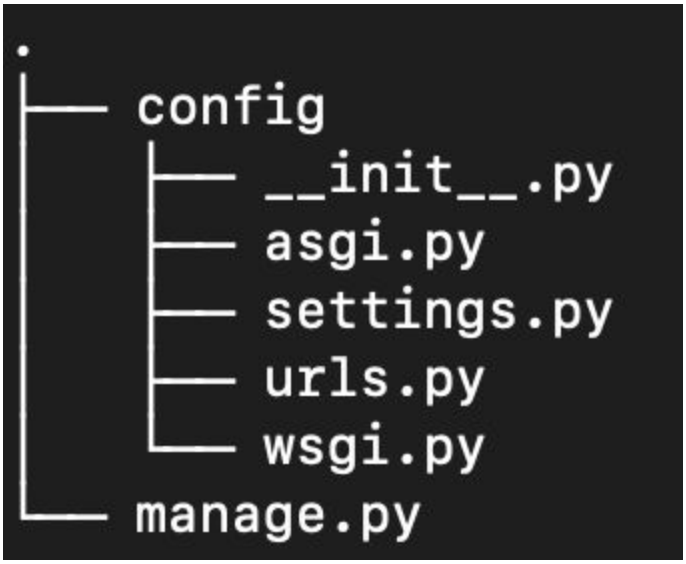
PWC: An Introduction to Django

Required/Expected setup

- Text editor you're comfortable with (I'll be using Sublime)
- Python 3.7 or greater
- A virtual environment and knowledge of how to
    - activate/deactivate that environment
    - Install packages in that environment
- Basic command line skills (we'll be using the terminal in this course)
    - Make/remove a directory
    - Create/delete files from your command line
    - Move files b/w directories
    - Be able to "find where you are" in your file structure
    - Know where files are saved in your file structure
- Having the `tree` command installed will be helpful, but not required.
- Git/Github
    - Github account (if you want to fork my code or make your code available to others)
    - Have Git installed
    - Git init
    - Git add; git push; git checkout; status

Slides for Part 1: Creation steps

1. Create a directory
2. Create a virtual env
    a. Provide detailed (how to)
    b. Try default Python option first, if you don't already have a preferred option
3. Install Django
4. Discuss diff b/w a project vs. an app
5. Create Django project
    a. `django-admin.py startproject config`
    b. Explain `` `config` `` as project name
    c. Move the "inner" config directory contents up one level: `mv config/config/* config`
    d. Move `manage.py` up one more level, to the "root" of the project
    e. Delete "inner" `config` directory

f. 

6. Run Django (rocket screen)
    a. `python manage.py runserver`
    b. Notice the “Unapplied Migrations” message
    c. Go to `http://localhost:8000/`
    d. See “rocket screen” at the above address; Congrats! You’ve got Django installed and running!
7. Git Time!
    a. Run `tree`, see the new files;
    b. Add `.gitignore`
    c. Git init; git status
    d. Git add: note the error message; Git WANTS to help you!
    e. `Git add .`; `git status`; Initial commit
    f. `Git log`: see your list of commits
8. Github (optional, but highly recommended)
    a. Create Github repo: note the “**Skip this step if you’re importing an existing repository.**” section
        i. do NOT initialize w/ a README
        ii. Do NOT add .gitignore (we already did)
        iii. Do NOT add a license (do so after this step if you feel the need)
    b. If you’re unsure, Github will provide you with the commands to “connect” your local repo to Github and push the code from local to remote..
    c. `git remote add origin git@github.com:{YOUR_USERNAME}/{YOUR_EXACT_REPO-NAME}.git`
    d. `git push -u origin master`
9. BREAK!

Slides for Part 2: Building our Apps

10. Apps focus on what information you want to keep track of (`Model`) and how you want to "work with" that information(`View`) and sharing the results of that work with the user (`Template`). You also have to let the user know WHERE to go to use the app (`URL`)
11. Migrate: `python manage.py migrate`
12. Create Super user: `python manage.py createsuperuser`
13. Log in at /admin, then go to Users

PAUSE TO DISCUSS WHAT WE'RE SEEING ON THE ADMIN

- Hover and see "Models in the Authentication and Authorization application"
- 1 User: We created a superuser
- No Groups
- Note: no changes to code
- "Tk8/Thrz" app provided FOR you
    - Mention reusable apps
- We'll focus on building YOUR app

CREATE DJANGO APP: Talk

1. `python manage.py startapp talks`
2. New directory: `talks`; Let's see what's inside
    a. Focus on "models" and "views"
3. Create Talk model
4. Register talks app
    a. "Our" approach vs. "Default"
5. Migrations
    a. Python manage.py makemigrations
    b.
```
(PWC) kojoidrissa@KOJOs-MacBook-Pro pwc_2020_django-intro % python manage.py makemigrations
Migrations for 'talks':
  talks/migrations/0001_initial.py
    - Create model Talk
```
        i. Let's look at that new file
    c. Python manage.py migrate
    d.
```
(PWC) kojoidrissa@KOJOs-MacBook-Pro pwc_2020_django-intro % python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions, talks
Running migrations:
  Applying talks.0001_initial... OK
```
6. Still nothing on the app?!? Register talks with the admin
    a. talks/admin.py

LET THE WORLD SEE!!

So, now it works for US, but that's not how web apps work!

1. Views, URLs, Templates

    a. Create a View
    b. Create URL to FIND that View (no more homepage!)
        i. http://127.0.0.1:8000/talks/ returns “TemplateDoesNotExist” error
    c. Create associated template
        i. Create `templates` directory in project root, then `AppName` directory inside it
        ii. mkdir templates; mkdir templates/talks; touch templates/talks/talk_list.html
    d. Tell Settings you made a Templates directory: https://docs.djangoproject.com/en/3.0/ref/settings/#dirs
2. Template Language: https://docs.djangoproject.com/en/3.0/topics/templates/#the-django-template-language
    a. HTML + Python + ...Template Magic?
    b. Variables, Tags, Filters, Comments
3. Talk Detail: View, URL, Template: Part 1
    a. What IS the URL for an individual talk?
    b. This view will need the ID (primary key) of the talk
    c. Path Converters
    d. Talk Detail URL and Template work. But no one will USE the site like that.
    e. What about the 404s?
4. Talk Detail: View, URL, Template: Part 2
    a. Resolve Unfound Talk errors
        i. Get_object_or_404
        ii. 404.html template
        iii. Update settings.py
            1. DEBUG = False
            2. ALLOWED_HOSTS = ["127.0.0.1"]
    b. Link Talk List entries to Talk Details
        i. Each Talk instance needs to know it’s own URL. That means a change to the model, using a named URL pattern.

Things I WON’T cover: HOMEWORK!

- Class Based Views: https://ccbv.co.uk/
    - There’s a lot of “magic” here, which I don’t think is great for people trying to LEARN Django
- Users creating talks
    - https://docs.djangoproject.com/en/3.0/ref/class-based-views/generic-editing/
    - CreateView, UpdateView, DeleteView
- Users/authentication: Will Vincent’s articles
    - 3 part intro:
        - https://learndjango.com/tutorials/django-login-and-logout-tutorial
        - https://learndjango.com/tutorials/django-signup-tutorial

        - https://learndjango.com/tutorials/django-password-reset-tutorial
    - User Model Details
        - https://learndjango.com/tutorials/django-custom-user-model
        - https://learndjango.com/tutorials/django-best-practices-user-permissions
        - https://learndjango.com/tutorials/django-best-practices-referencing-user-model
- Docker
    - Lacey's talk: https://www.youtube.com/watch?v=qsEfVSTZO9Q
- Testing
    - Use pytest & https://pytest-django.readthedocs.io/en/latest/
- Styling/CSS