Other notes

September 20th - deployment notes

- `ssh-keygen`
- public in droplet, private in github

cb-staging-rsa

su cb

updated passphrase

https://doane-ccla.gitbook.io/docs/learning-linux/file-permissions

chmod 600 filename

-----d
deleted DO_AUTH_TOKEN

note to self: don't change django secret key after deploying
https://docs.djangoproject.com/en/3.1/howto/deployment/checklist/

August 5th - pairing with J

Clojure doesn't have types
https://github.com/fulcrologic/fulcro - discrepancy

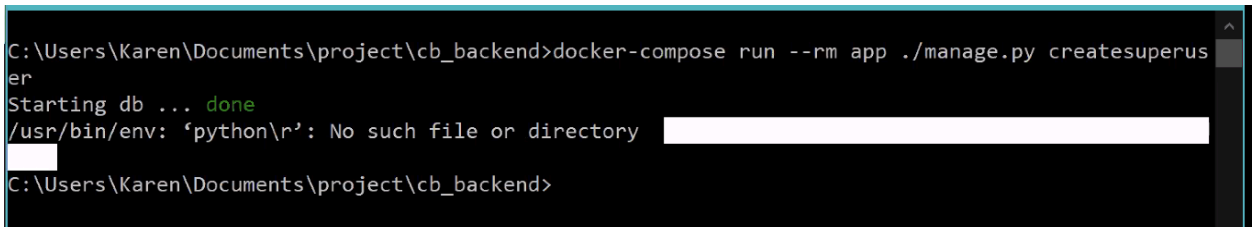
July 30th - pairing with J

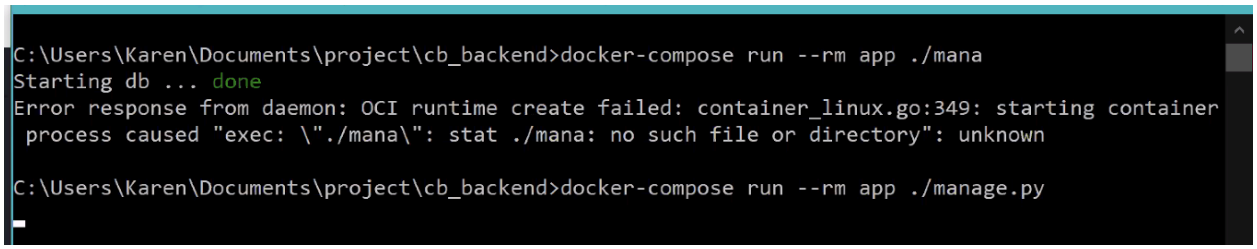
July 29th - pairing with Ben

July 28th - pairing with J

https://reactjs.org/docs/react-dom.html
https://medium.com/@etherealm/named-export-vs-default-export-in-es6-affb483a0910
https://scrimba.com/g/gintrotoes6
https://reactjs.org/docs/hooks-reference.html#usestate

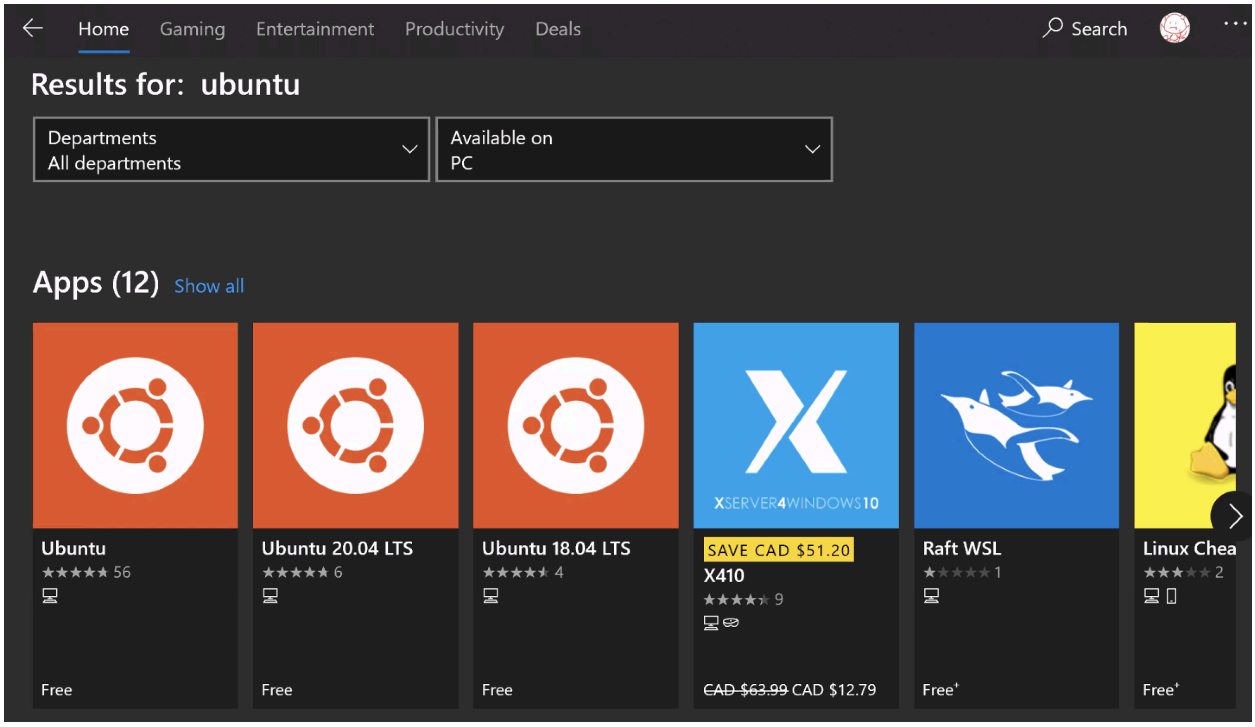
July 27th - pairing with Karen





Traditional way to run Linux alongside VM:

- WSL is running on top of Windows transparently; translating it into kernel calls. Translation layer between the low kernel-level… so there is no boot-up time. WSL can be brought up instantly. If you wanted to run `ls` in a command line shell, all you have to do is pull in command line, `wsl ls` and it will run the `ls` binary in linux. Treats all linux binaries as native
- Best way to install: Microsoft Stores -> Ubuntu



Stick to 18.04 LTS because 20.04 LTS is new

WSL and VSCode has a tight integration
In Windows, newline characters are 2 characters. In Linux, they're 1 character.

```
karen@DESKTOP-068VU91:/mnt/c/users/karen/documents/project/cb_backend$ ls
COPYING  README.md                contributing.md      nginx     project
LICENSE  community_health_file.md  docker-compose.yaml  postman   readme_win.md
karen@DESKTOP-068VU91:/mnt/c/users/karen/documents/project/cb_backend$ docker-compose up -d
Creating network "cb_backend_default" with the default driver
Creating adminer ... done
Creating mailhog ... done
Creating db      ... done
Creating app     ... done
Creating web     ... done
karen@DESKTOP-068VU91:/mnt/c/users/karen/documents/project/cb_backend$ docker-compose run --rm ap
p ./manage.py
Starting db ... done
Error response from daemon: OCI runtime create failed: container_linux.go:349: starting container
 process caused "exec: \"./manage.py\": stat ./manage.py: no such file or directory": unknown
karen@DESKTOP-068VU91:/mnt/c/users/karen/documents/project/cb_backend$
```

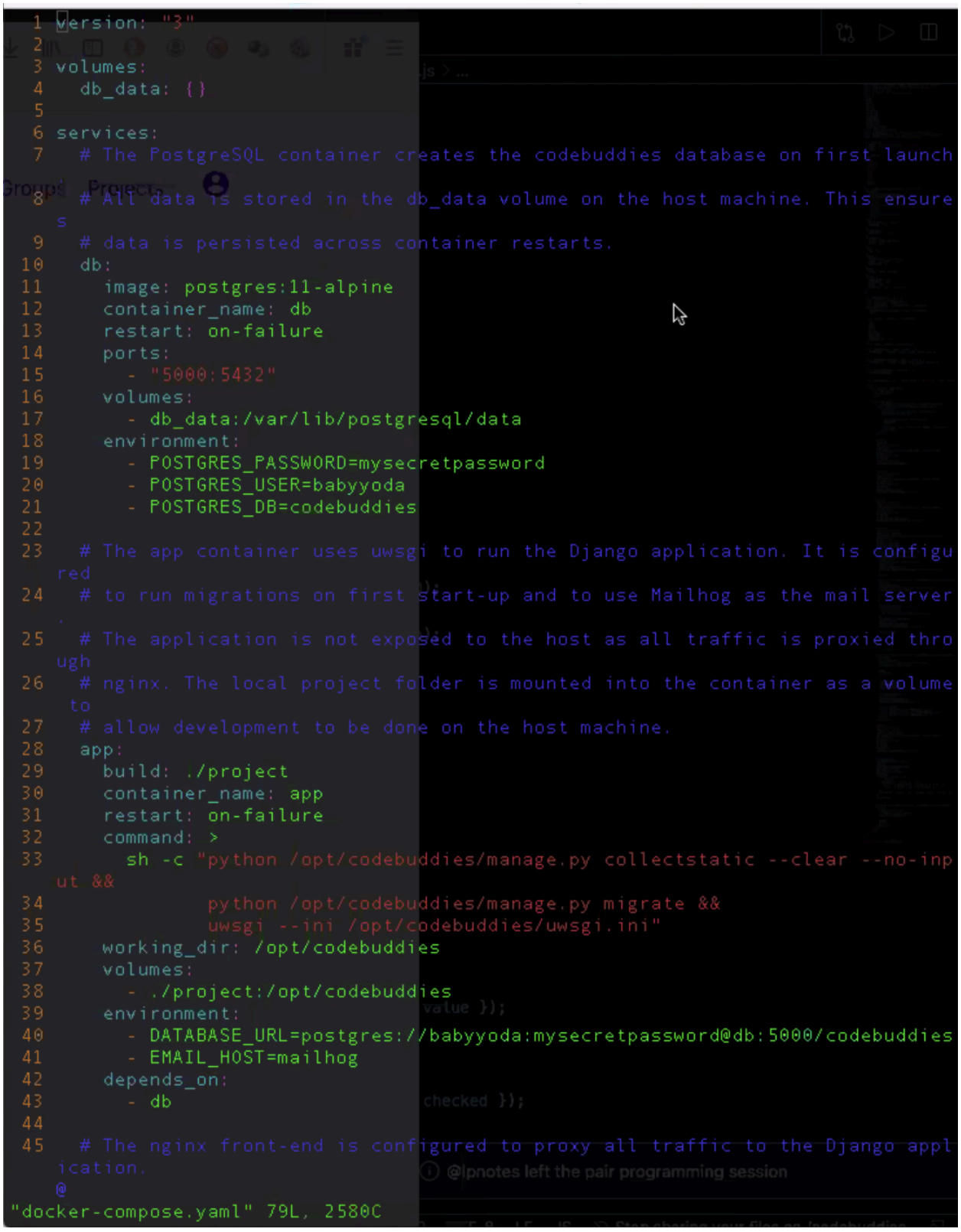
July 26th - pairing with Brent, Twitch streamed

Brent - postgres port issues

https://stackoverflow.com/questions/42416527/postgres-app-port-in-use

https://stackoverflow.com/questions/52579820/how-to-safely-stop-start-my-postgres-server-when-using-docker-compose

sudo pkill -u postgreshttps://stackoverflow.com/questions/37971961/docker-error-bind-address-already-in-usedocker rm -f $(docker ps -aq)



```
version: "3"

volumes:
  db_data: {}

services:
  # The PostgreSQL container creates the codebuddies database on first launch
  # All data is stored in the db_data volume on the host machine. This ensure
s
  # data is persisted across container restarts.
  db:
    image: postgres:11-alpine
    container_name: db
    restart: on-failure
    ports:
      - "5000:5432"
    volumes:
      - db_data:/var/lib/postgresql/data
    environment:
      - POSTGRES_PASSWORD=mysecretpassword
      - POSTGRES_USER=babyyoda
      - POSTGRES_DB=codebuddies

  # The app container uses uwsgi to run the Django application. It is configu
red
  # to run migrations on first start-up and to use Mailhog as the mail server
.
  # The application is not exposed to the host as all traffic is proxied thro
ugh
  # nginx. The local project folder is mounted into the container as a volume
 to
  # allow development to be done on the host machine.
  app:
    build: ./project
    container_name: app
    restart: on-failure
    command: >
      sh -c "python /opt/codebuddies/manage.py collectstatic --clear --no-inp
ut &&
             python /opt/codebuddies/manage.py migrate &&
             uwsgi --ini /opt/codebuddies/uwsgi.ini"
    working_dir: /opt/codebuddies
    volumes:
      - ./project:/opt/codebuddies
    environment:
      - DATABASE_URL=postgres://babyyoda:mysecretpassword@db:5000/codebuddies
      - EMAIL_HOST=mailhog
    depends_on:
      - db

  # The nginx front-end is configured to proxy all traffic to the Django appl
ication.
  @
"docker-compose.yaml" 79L, 2580C
```

What we tried:

Bill's solution: for people running into that docker issue, change docker-compose.yaml file to be port 5000:5432

1:32 **BillGloverUK**: The `ports` definition exposes the container port 5432 as 5432 on your mac.

11:33 **BillGloverUK**: You don't need this line unless you want to access the DB from your mac

11:33 **BillGloverUK**: Committing the change won't fix the problem as someone may already have something running on port 5000 on their mac.

11:34 **BillGloverUK**: Most sensible "fix" would be to avoid exposing the ports to local network. And then add FAQ on how to do it.

Bill:

The documentation on Networking in Compose can be quite overwhelming but does describe the networking behaviour when running `docker-compose up`.

"By default Compose sets up a single network for your app. Each container for a service joins the default network and is both *reachable*by other containers on that network, and *discoverable* by them at a hostname identical to the container name."

This means that processes running in different containers in the same application are all connected to each other on the same network. You don't need to expose ports for them to be able to talk to each other.

If your Postgres container is called `db` then other processes running in containers that are part of your application can connect to the URL `postgres://db:5432`

Now… if you want to access the DB from your host (in most cases your laptop) you need a route from your host network to the application network. This is what the `ports:` value does. When you see a directive like:

```
db:
  image: postgres
  ports:
    - "5000:5432"
```

This is saying, expose container port 5432 as port `5000` on the host network. Or, in other words, take all requests to port `5000` on my laptop and route them to port `5432` on the `db` container on the compose network.

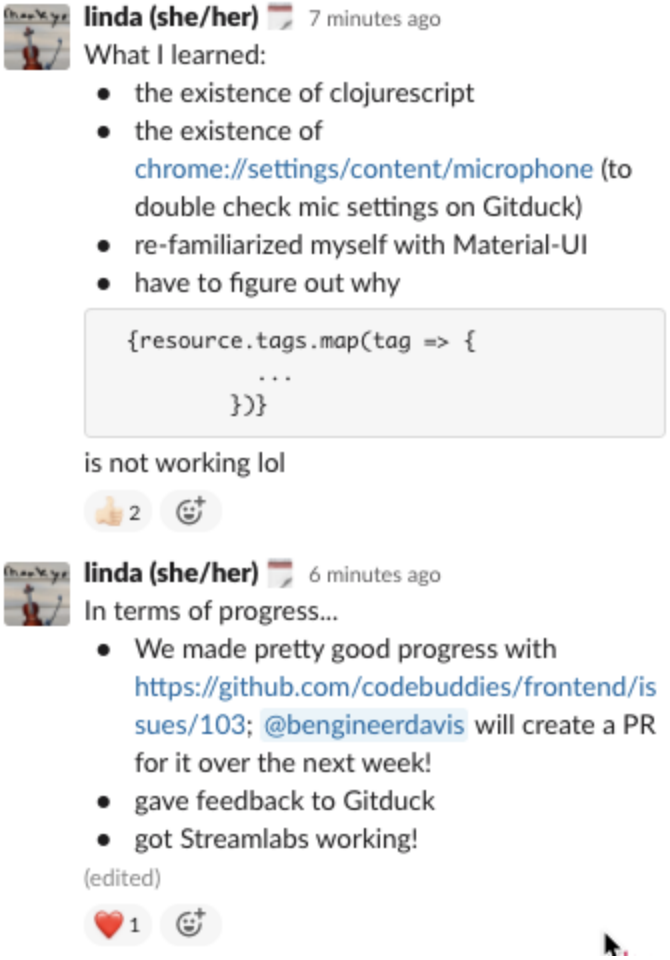
In our case, the Django application already sits on the same compose network as the database and so it is able to access the database on the container port.

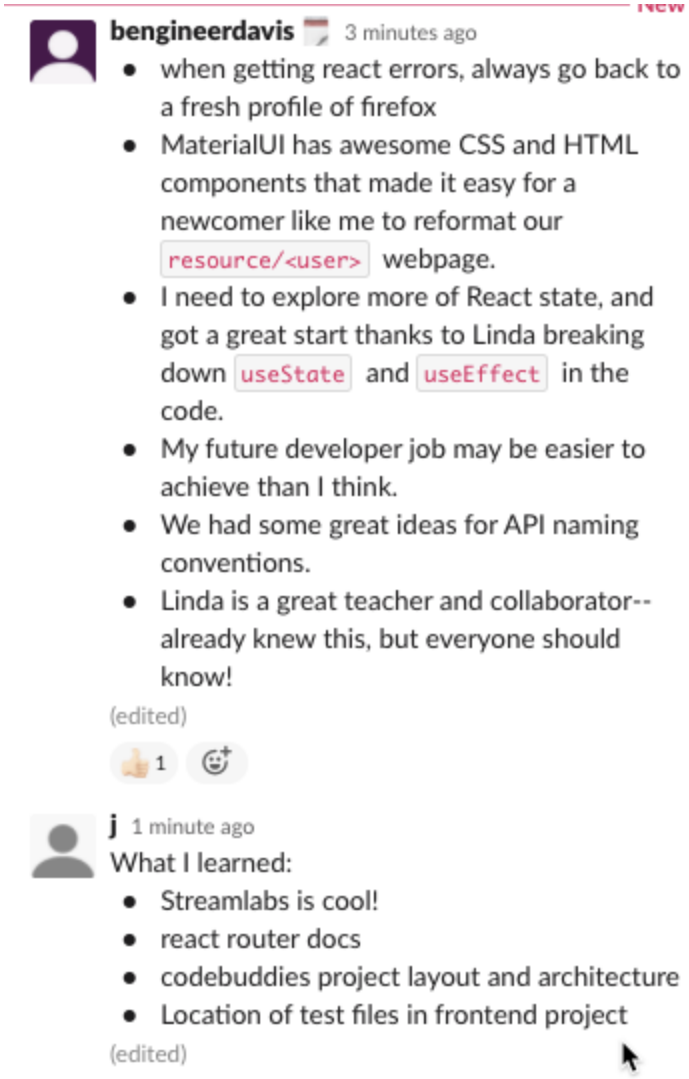
It feels like a diagram might help but I couldn't find anything that looked like it describe this with any clarity. (edited)

React todos:

// TODO : // when type, store values on change // decide what happens on successful submit (redirect to created resource, alert like "congrats you submitted resource..") // validation (follow auth implementation) (as type, post-submit) // display failed submissions to the user (if not already) // test for posting to mock up successful response // ensure these are all desired fields for forms

July 25th, 2020





May 10th, 2020 - Github Actions and React Testing Library

https://kentcdodds.com/blog/common-mistakes-with-react-testing-library
https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Roles
https://kentcdodds.com/blog/fix-the-not-wrapped-in-act-warning
https://help.github.com/en/actions/language-and-framework-guides/using-python-with-github-actions
https://github.com/bobheadxi/deployments
https://help.github.com/en/actions/reference/context-and-expression-syntax-for-github-actions#job-status-check-functions

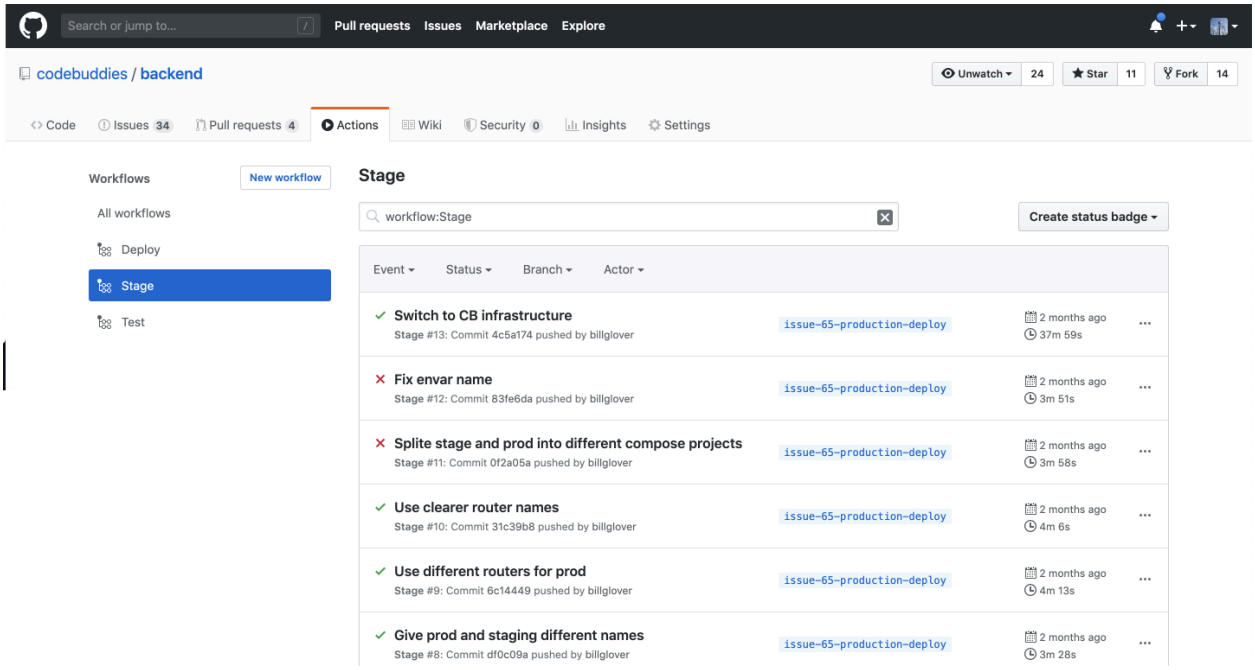
https://help.github.com/en/actions/configuring-and-managing-workflows/using-environment-variables
https://github.com/codebuddies/backend/actions/runs/46981820/workflow

April 2020 - Staging Deployment notes

1) [ ] Create a staging branch
2) [ ] Create a new workflow in Github Actions
    a) https://github.com/codebuddies/backend/actions/new
3) [ ] Create a DigitalOcean droplet
4) [ ] Create a postgres DB on DigitalOcean
5) [ ] Figure out how to deploy from that `staging` branch into a new wor Github Actions
6) [x] brew install doctl
7) [ ] Create a **docker-compose-staging.yaml** file
    a) Contents ?

Reference links

1. Github Actions syntax: https://help.github.com/en/actions/reference/context-and-expression-syntax-for-github-actions
2. Docker-compose reference: https://docs.docker.com/compose/compose-file/



April 26th, 2020

- Gaurav found that if we getByText('Sign Up') it'll select the span element and click it, but due to event bubbling reasons it's not clicking the actual button. That's why the tests don't pass.
- Chris48s explained our docker setup:
  Dockerhub is for publishing a docker container

  Docker-compose is for describing multi-container applications

  cb-backend is an application made up of 5 containers:

  - db

  - app

  - web

  - adminr

  - webhog

  with a docker-compose config to glue them all together.

  I think probably what you're doing is just trying to publish the app container to DockerHub, but that isn't enough to run the whole backend app.

  If you published all 5 containers, then the frontend user would be able to grab them all and compose them, but I think that would make things more complicated than just using the backend repo, not simpler: You're basically putting it on the frontend dev to write their own docker-compose instead of using the one that already exists.

  The advantage you might gain from publishing to dockerhub might be that you could publish stable releases of the backend so frontend doesn't have to track dev, but I'm not sure the project is at a stage where that has value yet.
- docker-compose and deployment on heroku:

  https://medium.com/tarkalabs/docker-deployments-to-heroku-5802b14df4fa

March 21st, 2020

❓ **Question -- what topics would folks like to discuss in a 30-minute chat about CBV3 goals (https://github.com/orgs/codebuddies/projects/1) possibly sometime this weekend?**

Anticipated agenda:

- I'd be interested in finding out a bit more about:
    - what you want to get done next and
        - Bethany: thoughts on logins, migrating the database. On the backend ,we have one piece not completely done (resources?)
        - Pick big three things we need for the working MVP
        - Great that we're having a conversation about the whole website, but let's get a roundtrip working and a beta working. Let's try to balance those things.
        - Linda: MVP is **user login/authentication, ability to post, edit, delete a resource**
        - Frontend is currently using mocked resources (db.json)
    - what you most need help with on the backend (chris48s)
        - Bethany: things to do left on backend
            - more robust testing
            - email confirmation /
            - Milestone for resources: https://github.com/codebuddies/backend/issues?q=is%3Aopen+is%3Aissue+milestone%3Aresources
            - Change email field to be required
- I'd be interested in agreeing how we make the decisions that result in answers to the questions above. (Bill)
    - MVP -> simpler choice
    - Angelo: don't make the same mistakes we made with V2
    - Backend decision log: https://github.com/codebuddies/backend/wiki/Decision-log
    - Frontend decision log: https://github.com/codebuddies/frontend/wiki/Technical-decision-log
    - Sunu: too much flux in decision making. We should come up with a fixed API definition.
        - How do we balance necessary changes?
        - How do we make changes discoverable?
- I'd be interested in figuring out what the minimum is that we can start to "ship". (Bill)

- I’d be interested in exploring the idea of core contributors who would be willing to help keep PR reviews moving. (Bill)
    - **frontend** and **backend**
        - email notifications not working
    - Angelo: PRs aren't reviewed fast enough
    - Have more discussions in the front end to discuss features and establish context and decisions
    - Should we have a call out to reviewers?
        - Yes, but let’s not have drive-by reviewers; it’s difficult to have insightful feedback if you don’t have context to how things are going with the project at a higher level?
- Which features we're going to implement next? (Bethany)
    - Roundtrip user registration + resources
    - Tests
    - Have community discussions around implementations that would be suggested by other contributors
        - Where should we have our discussions regarding these kinds of topics?
            - Decisions reached should be recorded in GitHub
            - Okay to have informal discussions in Slack, but once we start having a more serious decision, we should start recording this in GitHub.
- Also wanting to know about things like tests, shipping schedule, and how front end design decisions will impact backend implementations.
    - Sunu: API schema
    - Linda: discussion API, hangouts -> warrants API discussion
        - If it appears in figma, is it on the Roadmap?
        - Pull out features from figma and do sizing
        - Angelo: maybe we should have a kickoff meeting for each milestone
- Would like to discuss general roadmap for both backend and frontend projects (Angelo)

Attendance: Linda, Angelo, Bethany, Sunu, Chris, Emmanuel
Recorded Meeting

TODOS - MVP

- Add milestone for user authentication - backend (Bethany)
- Require email confirmation on the backend after user signs up

    - Ticketed
    - Assigned
- Make email required on backend
    - One-line change (plus a migration +1)
    - Ticketed
    - Easy ticket possibly for anyone? (but should be done quickly)
- Tests on backend
    - To be discussed (Chris and Bethany)
- Authentication tests on React (Linda)
- Users model
    - Ticket
- Write up GitHub Actions + React Testing Library documentation (Angelo)

NEXT FEATURES

- Hangouts

DECISIONS

- Have a kickoff meeting for each new milestone, mapping figma design -> sized tickets for FE and BE
- Strive to put as many decisions/discussions in GitHub as possible (although OK to ask questions/start informal discussions in #codebuddies-meta on Slack)

March 8th, 2020

https://blog.hlab.tech/part-ii-how-to-sign-up-user-and-send-confirmation-email-in-django-2-1-and-python-3-6/

February 20th, 2020

Docker w/ Bill

docker-compose.yaml — backend

docker-compose.yaml

```
      - POSTGRES_PASSWORD=mysecretpassword
      - POSTGRES_USER=babyyoda
      - POSTGRES_DB=codebuddies

  # The app container uses uwsgi to run the Django application. It is configured
  # to run migrations on first start-up and to use Mailhog as the mail server.
  # The application is not exposed to the host as all traffic is proxied through
  # nginx. The local project folder is mounted into the container as a volume to
  # allow development to be done on the host machine.
  app:
    build: ./cbv3_django_prototype
    container_name: app
    restart: on-failure
    command: >
      sh -c "python /opt/cbv3_django_prototype/manage.py migrate &&
             uwsgi --ini /opt/cbv3_django_prototype/uwsgi.ini"
    volumes:
      - ./cbv3_django_prototype:/opt/cbv3_django_prototype
    environment:
      - DATABASE_URL=postgres://babyyoda:mysecretpassword@db:5432/codebuddies
      - EMAIL_HOST=mailhog
    depends_on:
      - db
```

Dockerfile:

- tells Docker how to biuld a container for your application

Dockerfile — backend

Dockerfile

```
FROM python:3.7

RUN apt-get update && \
    apt-get install -y && \
    pip3 install uwsgi

COPY ./requirements/ /opt/cbv3_django_prototype/requirements/

RUN python3 -m pip install --upgrade pip
RUN pip3 install -r /opt/cbv3_django_prototype/requirements/local.txt

RUN groupadd -r uwsgi && useradd -r -g uwsgi uwsgi

ENV DJANGO_ENV=prod
ENV PYTHONUNBUFFERED 1
ENV PYTHONPATH /opt/cbv3_django_prototype/

COPY ./ /opt/cbv3_django_prototype/

EXPOSE 8000

CMD ["uwsgi", "--ini", "/opt/cbv3_django_prototype/uwsgi.ini"]
```

python3.7-alpine

Every time docker runs it saves a version of the image.

If the Python image changes, it rebuilds everything.

If the requirements changes, everything requirements and below runs.

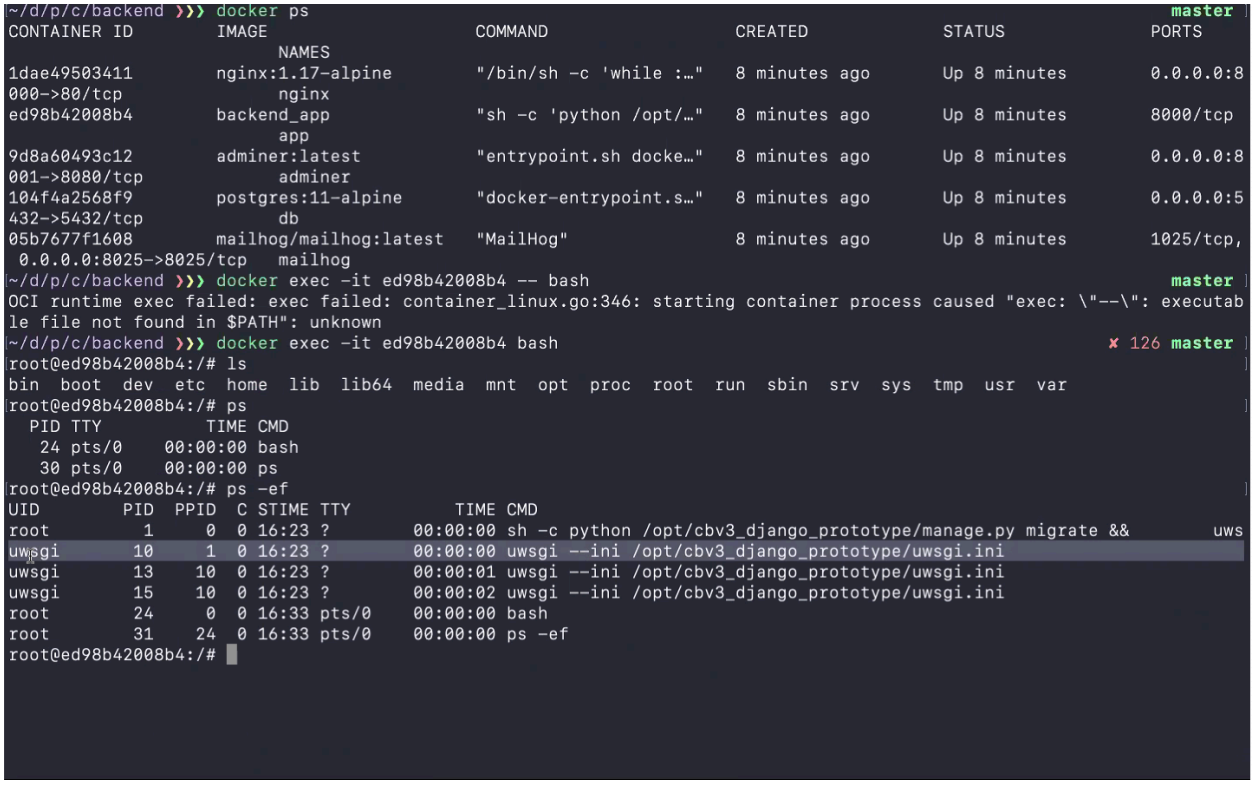
The step required to build the application depends on which line changed.

Locally, the version of the app inside the docker container is available locally.

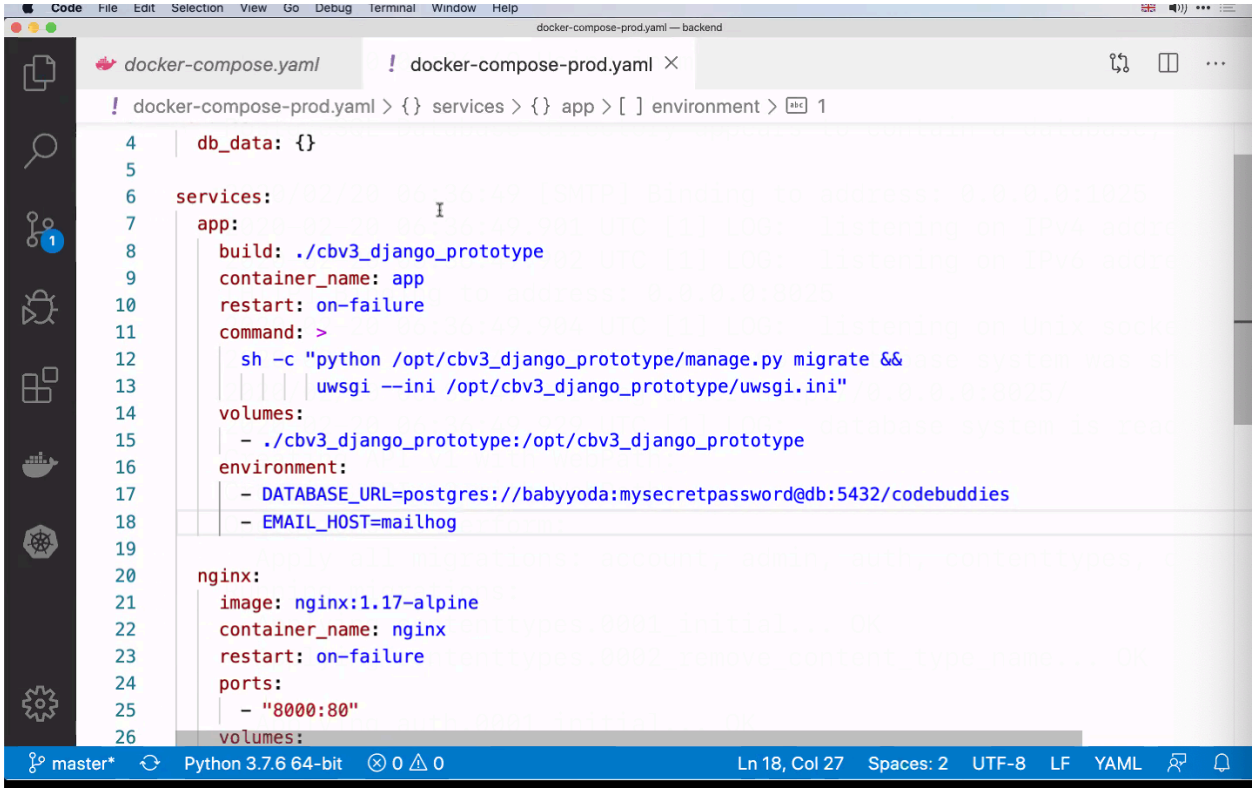
Not rebuilding the image associated with the Python file.

Makefile: if requirements change, run build. If not, then…

Makemigrations is a step who the developer who changed the models runs.



Deployment to production:



Next steps: Create a GitHub action.

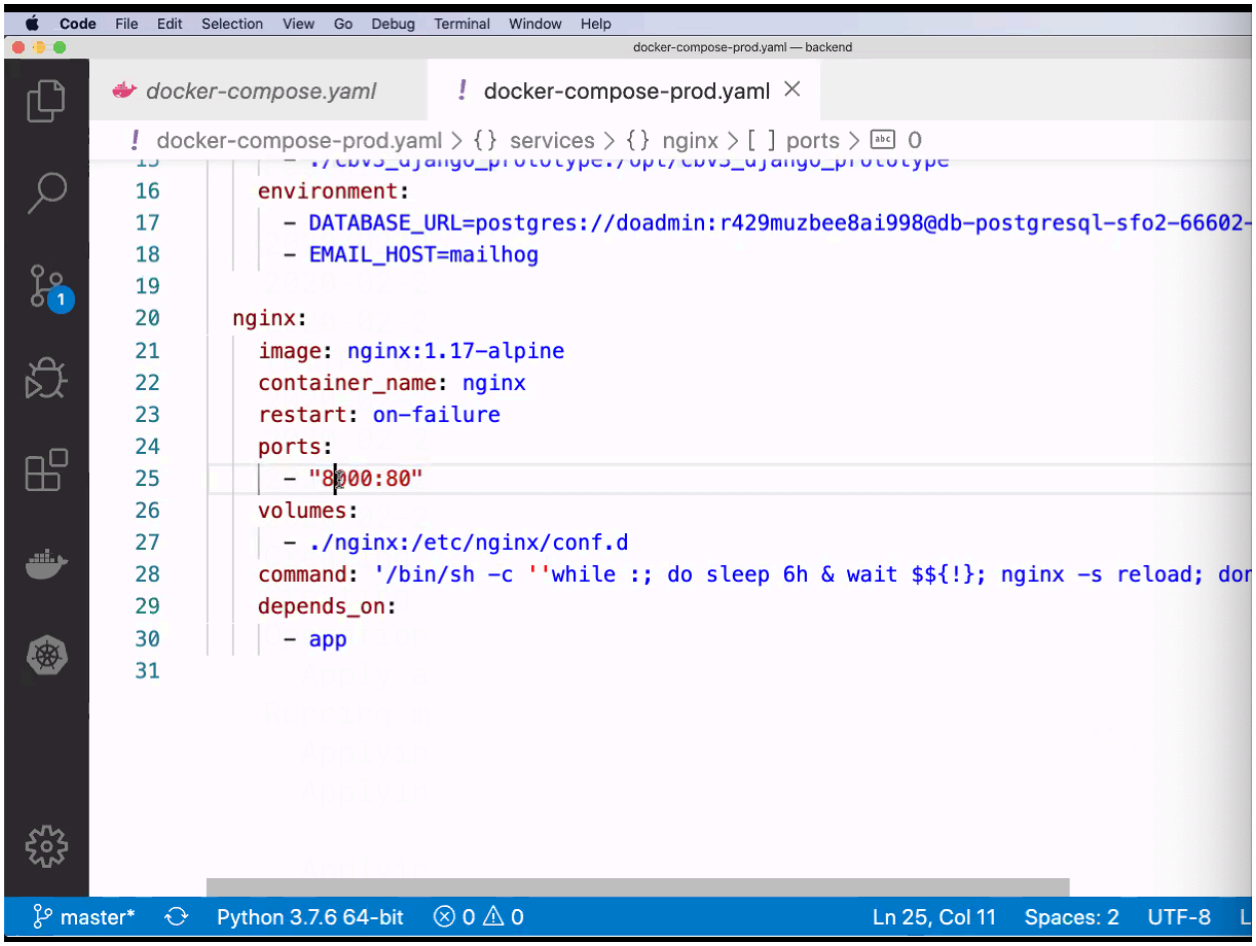
In DigitalOcean

docker build -t billglover

- we build the image
- push the image to dockerhub
- when we tag a version in the repo in GitHub, the pipeline triggers a script on the DigitalOcean box that says "deploy a new version using docker-compose up"
- docker push billglover/cb-backend:0.0.1

Questions:

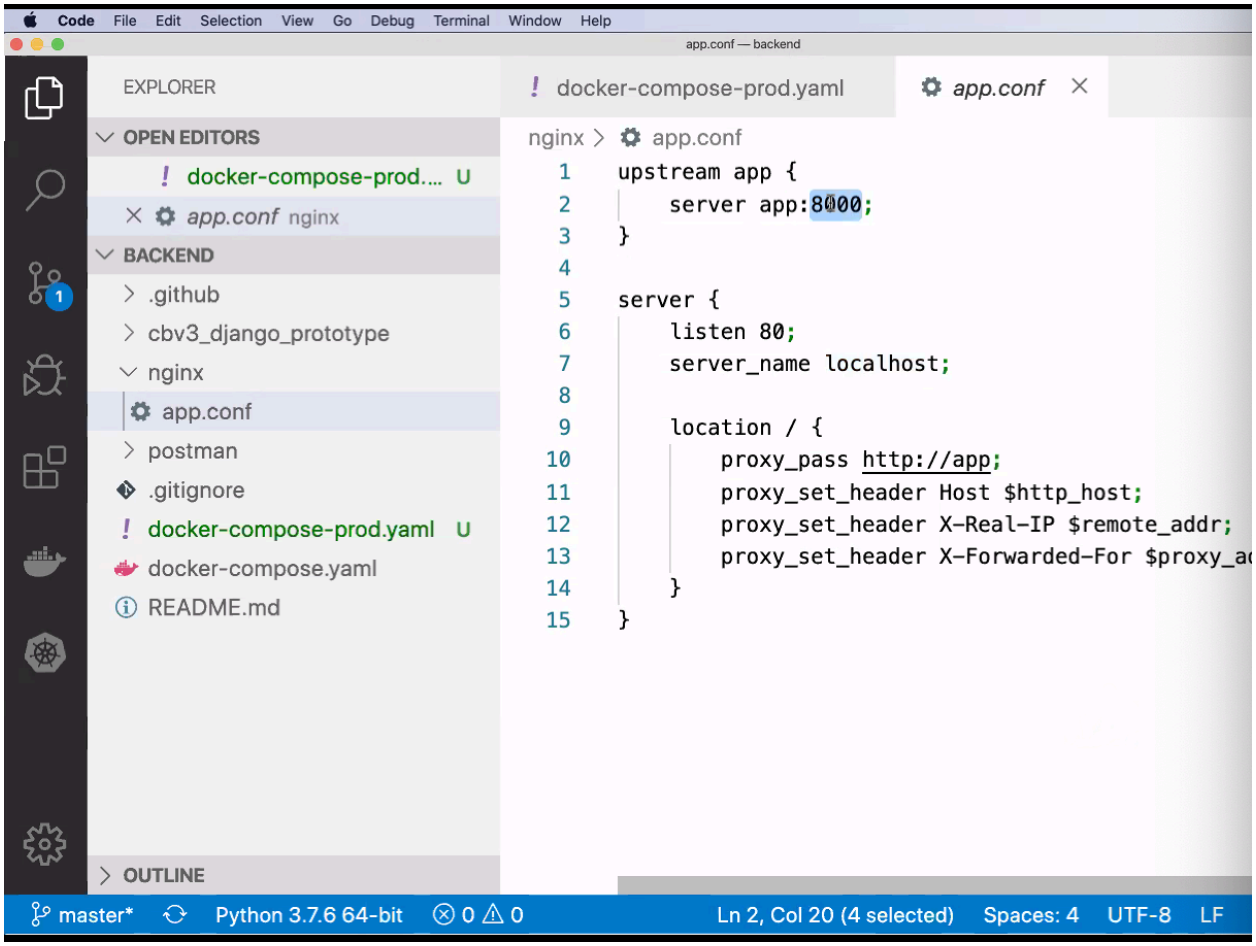
- complexity with deploying without taking something offline?
- when do we release?
    - FE is easy
    - If we change the API, the API needs some notion of "this is an API change that doesn't change the contract"
        - minor changes get deployed
        - major changes -> coordinate with frontend



In production, 443

Why is nginx not exposing 8000:80?

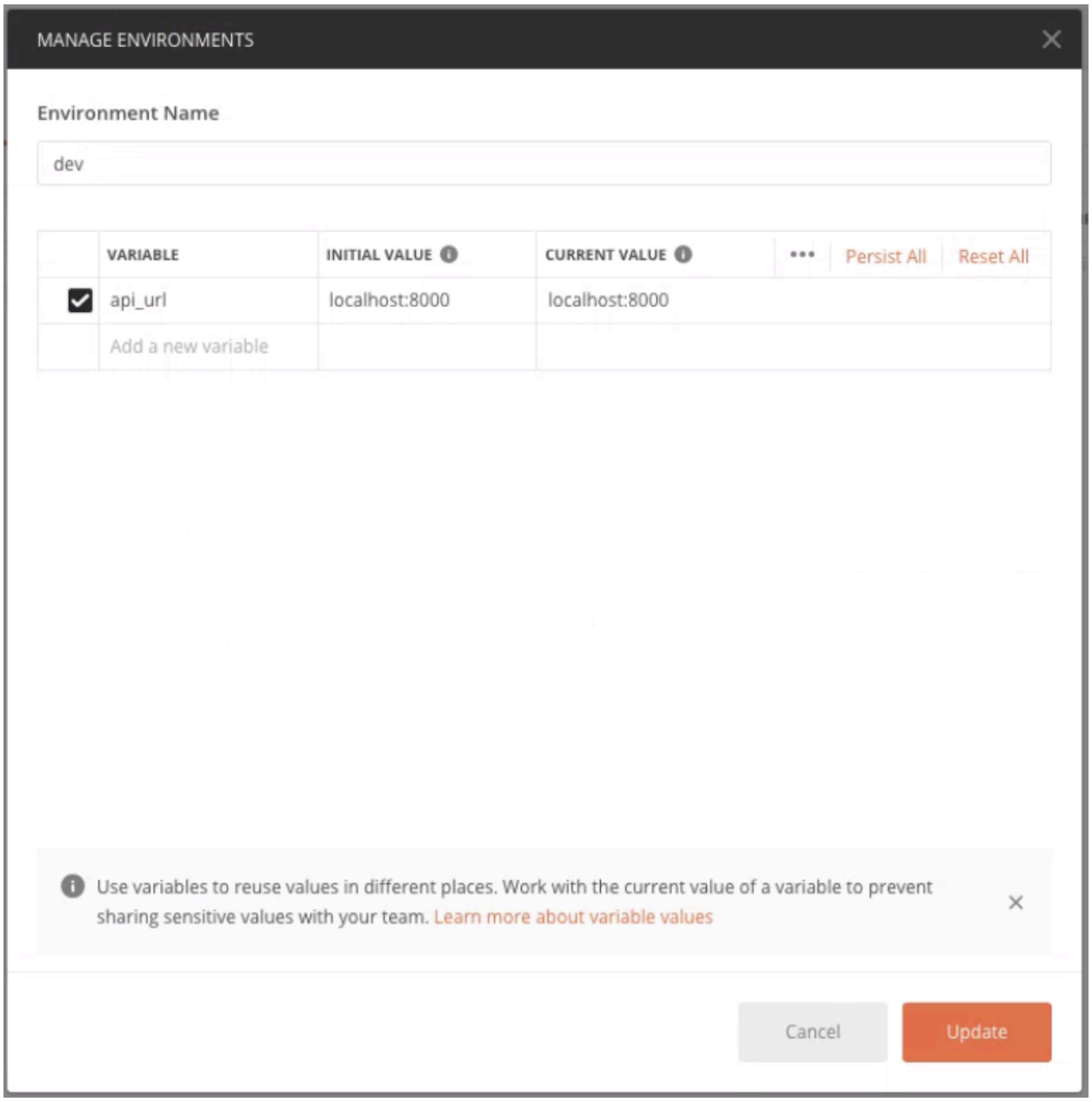
- app is listening port 8000
- nginx inside docker is listening to 80
- when nginx gets a request to /api, it forwards it to 8000 inside the container
- all of that is isolated
- the exposed command in the docker-compose-prod.yaml file "8000:80" says "take 8000 outside the container (in your laptop) and connect it to 80 inside the container."
- inside app.conf:



Somewhere in app.conf, we need to set the config so that it listens?

- Wrap it up in

February 16th, 2020



Make sure you select "dev" as the environment

The token returned from running the "create user" endpoint actually gets saved in globals so we don't need to explicitly add it in the parent settings

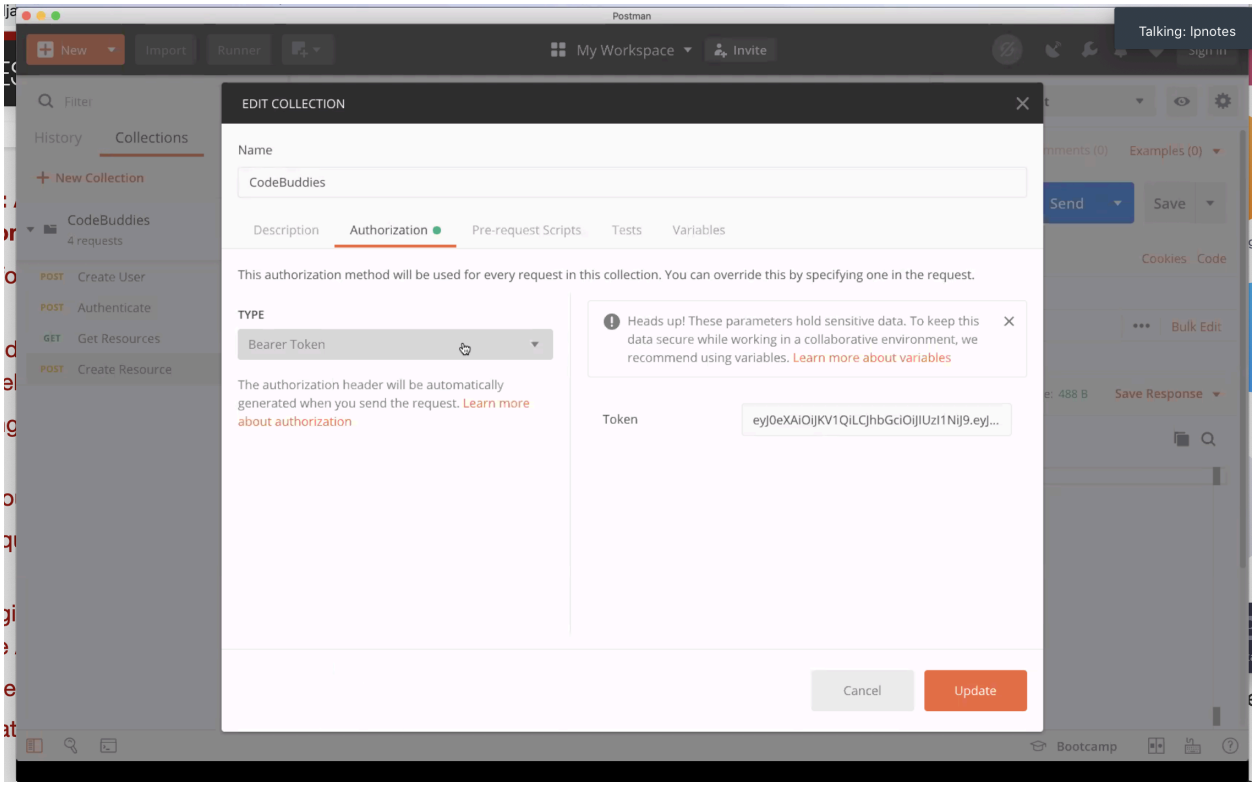
```
sudo pkill -u postgres
```

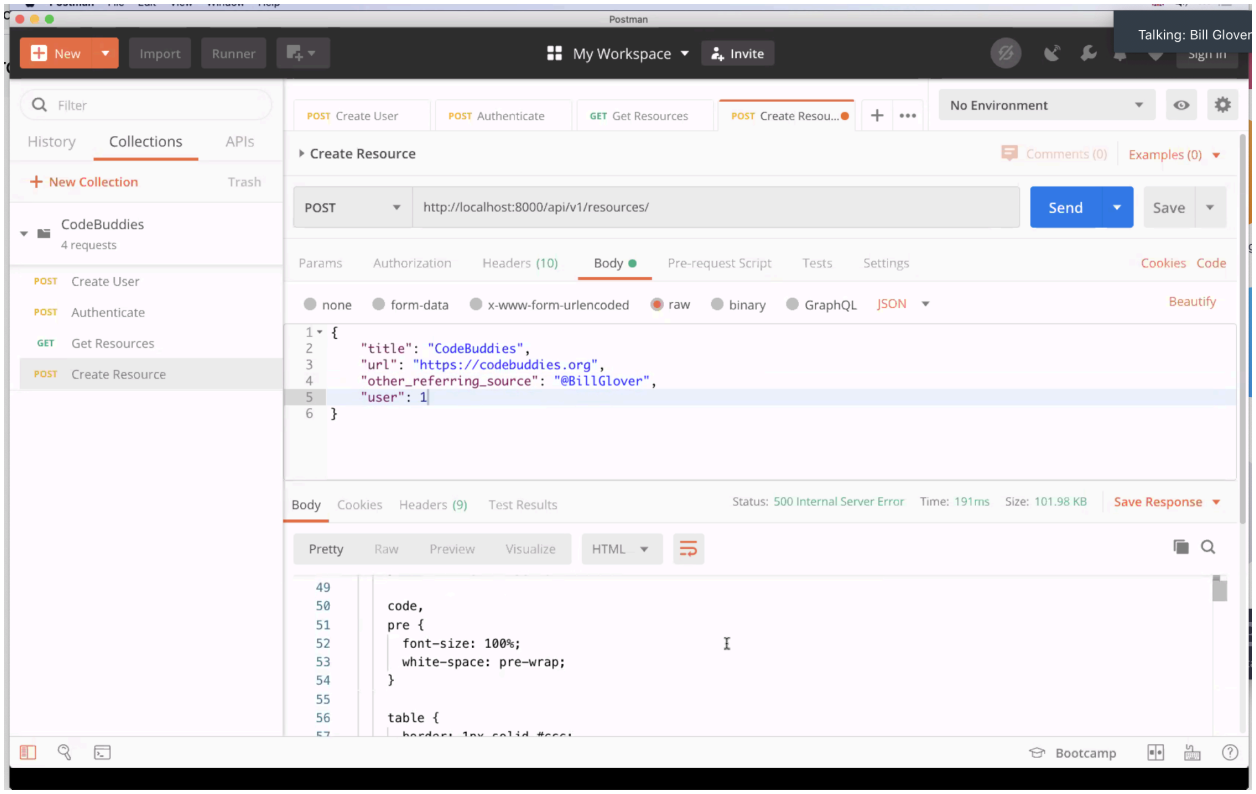
February 12th, 2020

- username is a required field
- one user can create resources based on another
- When you create a resource, how do you set the value of the field to be the logged-in user
- `owner = models.ForeignKey('auth.User' vs. settings.authUserModel`

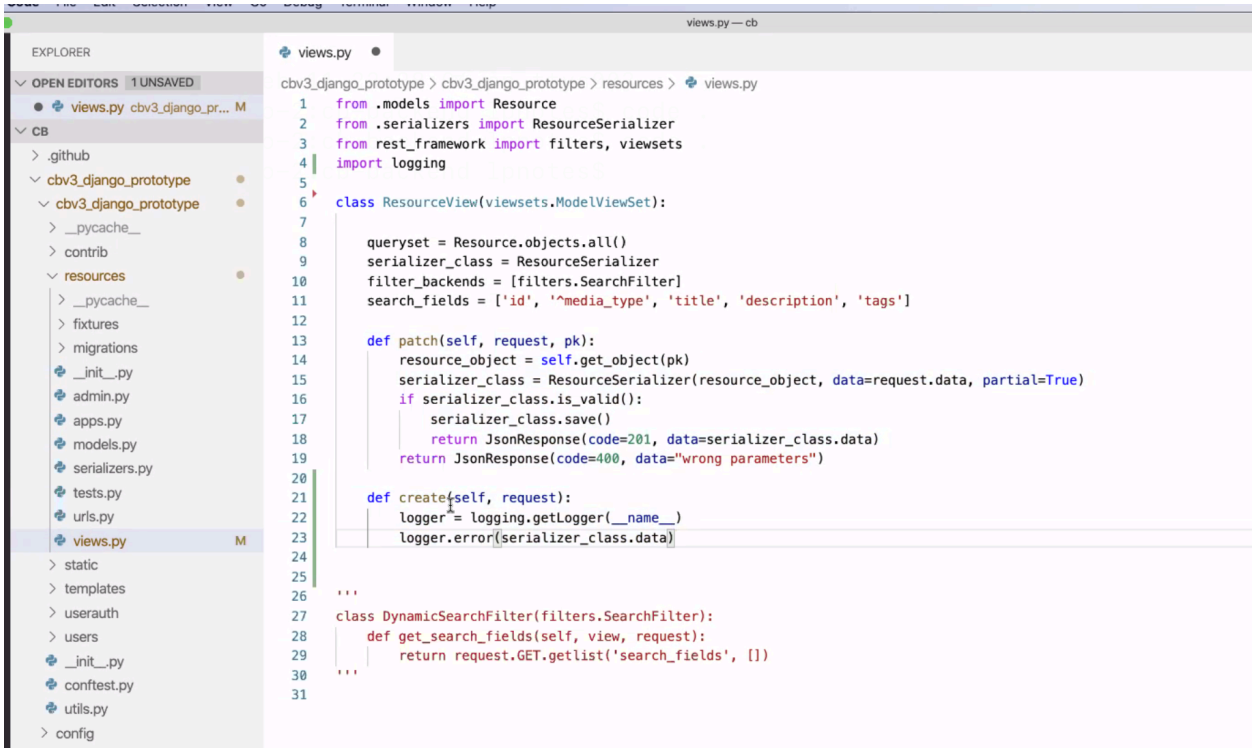
  `(https://docs.djangoproject.com/en/3.0/topics/auth/customizing/)`

- `composer-compose -d (detaches do you don't see the logs)`
- `docker stop [id]`
- `docker rm [id]`
- `docker-compose down`
- `docker-compose logs`



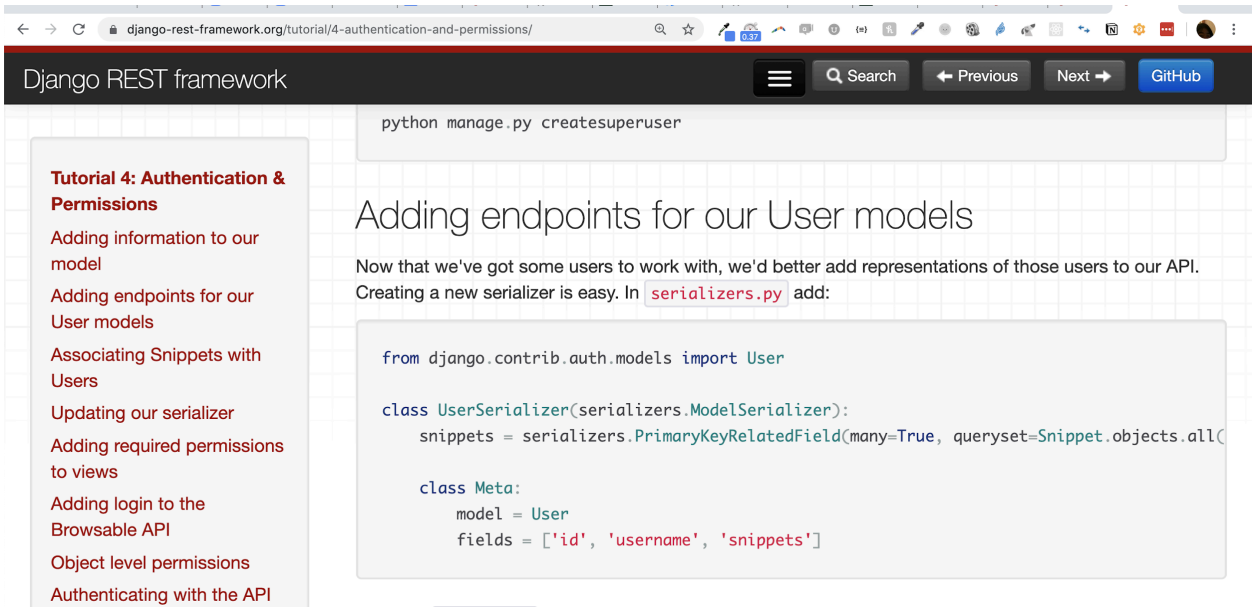


Request Method: POST
Request URL: http://localhost:8000/api/v1/resources/
Django Version: 2.2.4
Exception Type: TypeError
Exception Value: `Object of type _TaggableManager is not JSON serializable`
Exception Location: /usr/local/lib/python3.7/json/encoder.py in default, line 179
Python Executable: /usr/local/bin/uwsgi
Python Version: 3.7.6



Questions

- Why are we using viewsets.ModelViewSet in views.py? What is the difference between that and using `serializers.ModelSerializer` in the tutorial at https://www.django-rest-framework.org/tutorial/4-authentication-and-permissions/?



- Why is serializer_class.data not accessible within def create() but within def patch()?

February 9th, 2020

These are some examples of using CURL to create a user, request a token and query the API. I initially tripped myself up when using the token because I was prefixing my token with ‘Token’ or ‘JWT’ as this is what I found in many examples online. Line 327 in base.py appears to configure our auth middleware to use a prefix of ‘Bearer’.

✅ Creating a user with CURL:

```
curl -F "username=demo" -F "password=pass" -XPOST
http://localhost:8000/auth/users/
```


```
{"username":"demo","token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZS
I6ImRlbW8iLCJpYXQiOjE1ODEyODU3NjQsImV4cCI6MTU4MTI4OTM2NCwidXNlcl9pZCI6Mywib3JpZ
19pYXQiOjE1ODEyODU3NjR9.92hi1sK7iMk4K6fHVdkNzCUc6g915u5wPWD9r1gtrbM","first_nam
e":"","last_name":"","email":""}
```


✅ Requesting a token with CURL:

```
curl -F "username=demo" -F "password=pass" -XPOST
http://localhost:8000/auth/obtain_token/
```


```
{"token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6ImRlbW8iLCJpYXQi
OjE1ODEyODU4NTksImV4cCI6MTU4MTI4OTQ1OSwidXNlcl9pZCI6Mywib3JpZ19pYXQiOjE1ODEyODU
4NTl9.5MUtf3iRk4GIaFG9LYSkdTnufGPNfi7tgj-9R-GhpFo","username":"demo"}
```


✅ Using the API with a JWT token:

```
curl -H 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6ImRlbW8iLCJpYXQiOjE1ODEyOD
U4NTksImV4cCI6MTU4MTI4OTQ1OSwidXNlcl9pZCI6Mywib3JpZ19pYXQiOjE1ODEyODU4NTl9.5MUt
f3iRk4GIaFG9LYSkdTnufGPNfi7tgj-9R-GhpFo' -XGET
http://localhost:8000/api/v1/resources/
```


```
{"count":0,"next":null,"previous":null,"results":[]}
```


- Questions
    - /auth/users endpoint not working - ask @bethany maybe
    - How do we store tokens as cookies in React?
    -
- Decrypting a token:
    - eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6ImFkbWluMiIsIml
      hdCI6MTU4MTI3NjM4NCwiZXhwIjoxNTgxMjc5OTg0LCJ1c2VyX2lkIjoyLCJvcmln

X2lhdCI6MTU4MTI3NjM4NH0.u-hRyIbt87neyECIcrgey-aC2SutWvgFlHoAeSI6tTk
    - JSON.parse(atob('eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6ImFkbWluMiIsImlhdCI6MTU4MTI3NjM4NCwiZXhwIjoxNTgxMjc5OTg0LCJ1c2VyX2lkIjoyLCJvcmlnX2lhdCI6MTU4MTI3NjM4NH0.u-hRyIbt87neyECIcrgey-aC2SutWvgFlHoAeSI6tTk'.split('.')[1]));
    - {username: "admin2", iat: 1581276384, exp: 1581279984, user_id: 2, orig_iat: 1581276384}
- Curl requests
    - Succeeds in logging in a user
        - curl -d '{"username": "admin2", "password": "Helloworld12!"}' -H "Content-Type: application/json" -X POST http://localhost:8000/auth/obtain_token/
            - {"token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6ImFkbWluMiIsImlhdCI6MTU4MTI3NjM4NCwiZXhwIjoxNTgxMjc5OTg0LCJ1c2VyX2lkIjoyLCJvcmlnX2lhdCI6MTU4MTI3NjM4NH0.u-hRyIbt87neyECIcrgey-aC2SutWvgFlHoAeSI6tTk","username":"admin2"}
    - Does not succeed because wrong password entered
        - curl -d '{"username": "admin2", "password": "Helloworld1"}' -H "Content-Type: application/json" -X POST http://localhost:8000/auth/obtain_token/
            - {"non_field_errors":["Unable to log in with provided credentials."]}
    - Succeeds in creating a new user
        - curl -d '{"username": "admin3", "password": "Helloworld12!"}' -H "Content-Type: application/json" -X POST http://localhost:8000/auth/users

January 7th, 2020

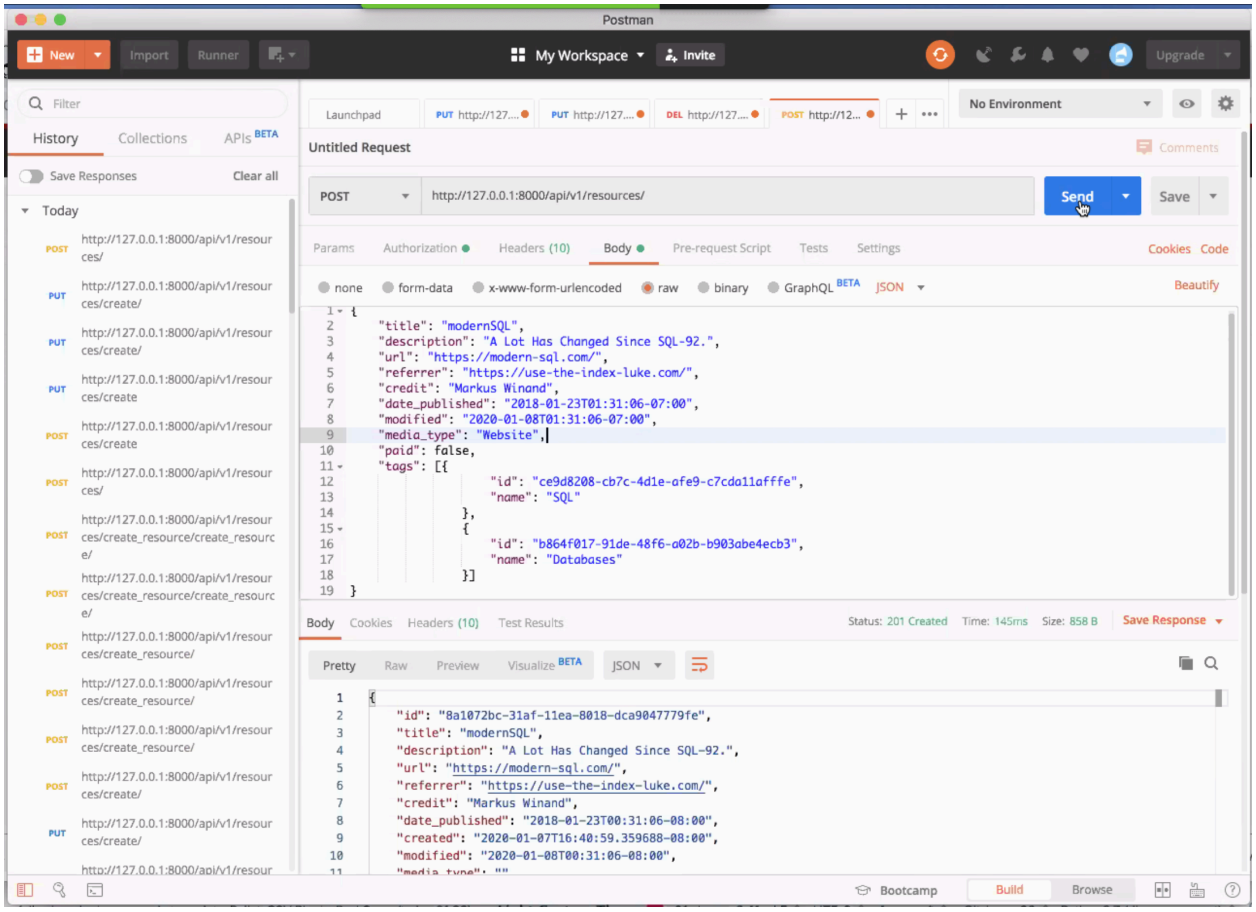
Next steps:

- [] move these notes into a Github wiki (Linda)
- [] Write up conventions for contributing -- filing Github issue convention, branch naming conventions, Django - follow official guidelines/link to them, React/prettier setup, config file for styling
- **[] Document setup notes on Linux (Ben)**
- **[] Document setup notes for Mac**
- **[] Document setup notes for Windows**
- **[] Write tests for issue-24 and create a PR for it (Bethany)**
- **[] Deploy Django app so that people can give feedback on the API (Linda)**
- [] Work more on UI designs (Linda)
- [] Ask for feedback from others
- [] Start working on more Django API endpoints

- [] Form validation pairing w/ peoray - React (Linda)
- [] Dynamic filter backend (Bethany)

Progress made:

- This commit (issue 24) that changes the fields: https://github.com/codebuddies/django-concept/commit/9c322cb253d50cc8ae61f5cadafec201b2f8a41e
- Bethany did a 30-minute walkthrough/demo of DRF and work so far

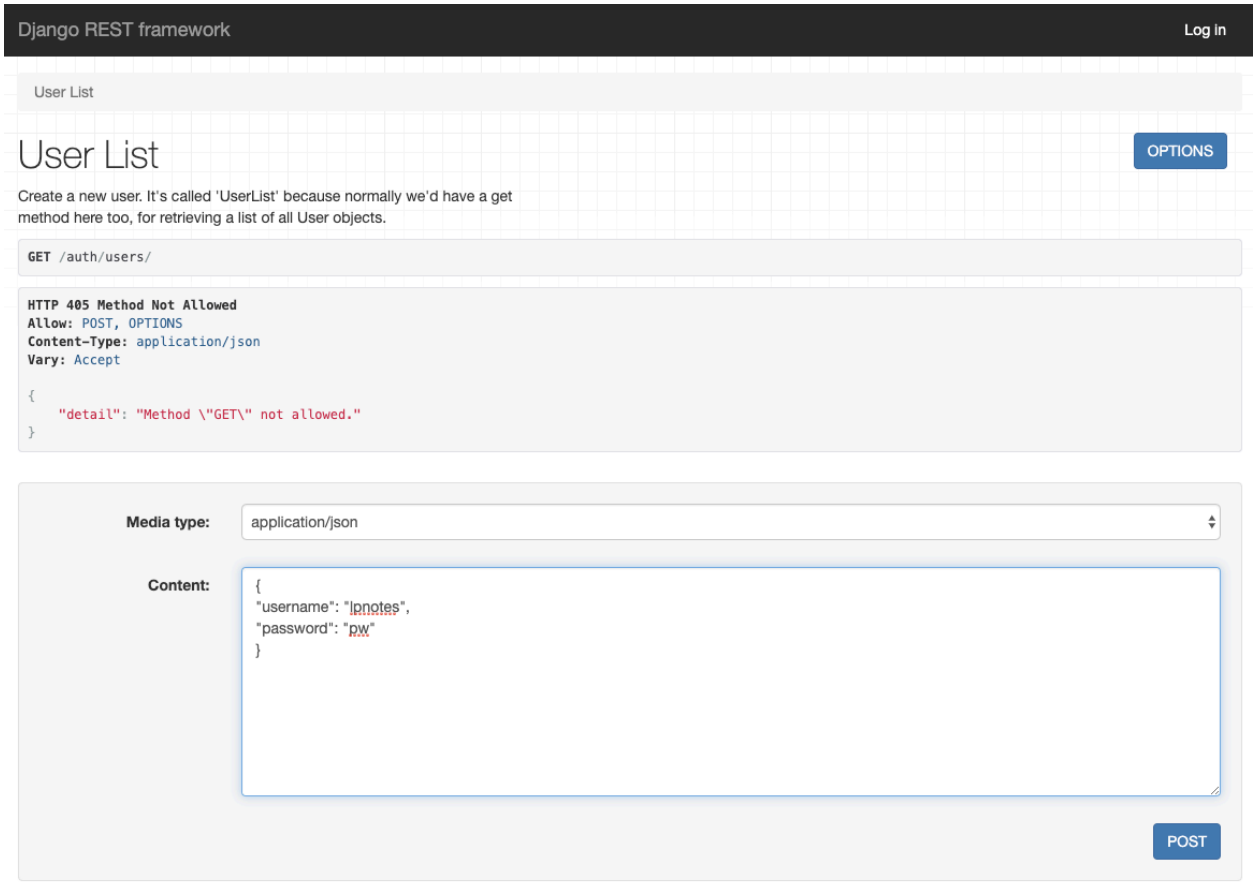


Checking out bethany fork

- **bethany/issue-20**
- **http://127.0.0.1:8000/api/v1/**

Create a user:

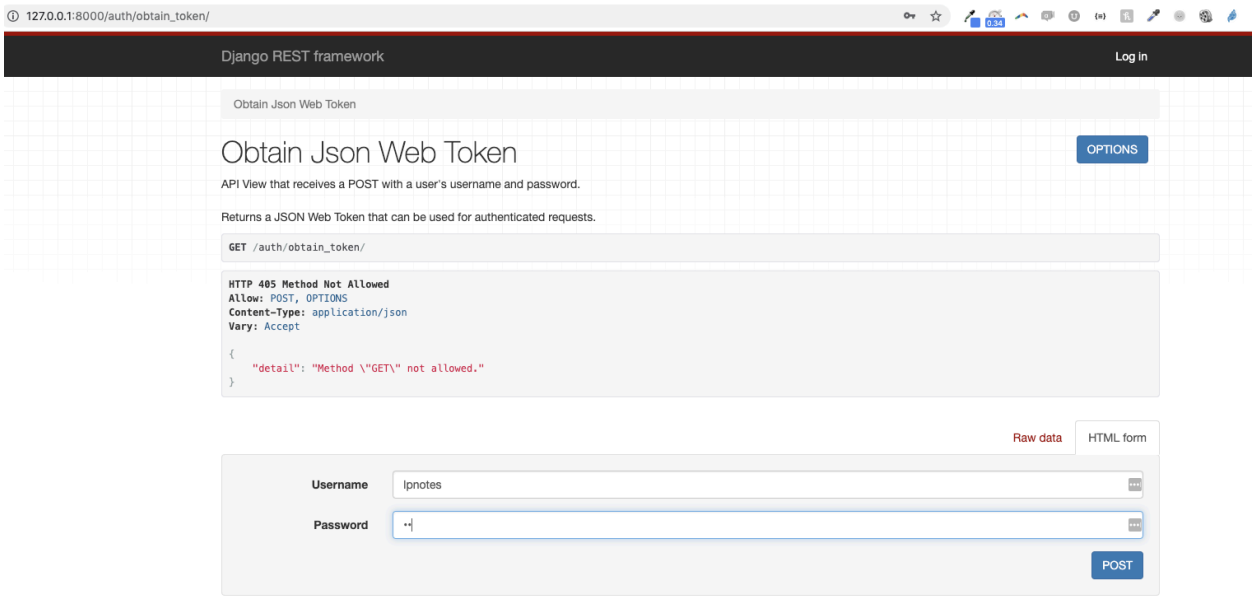
- **http://127.0.0.1:8000/auth/users/**

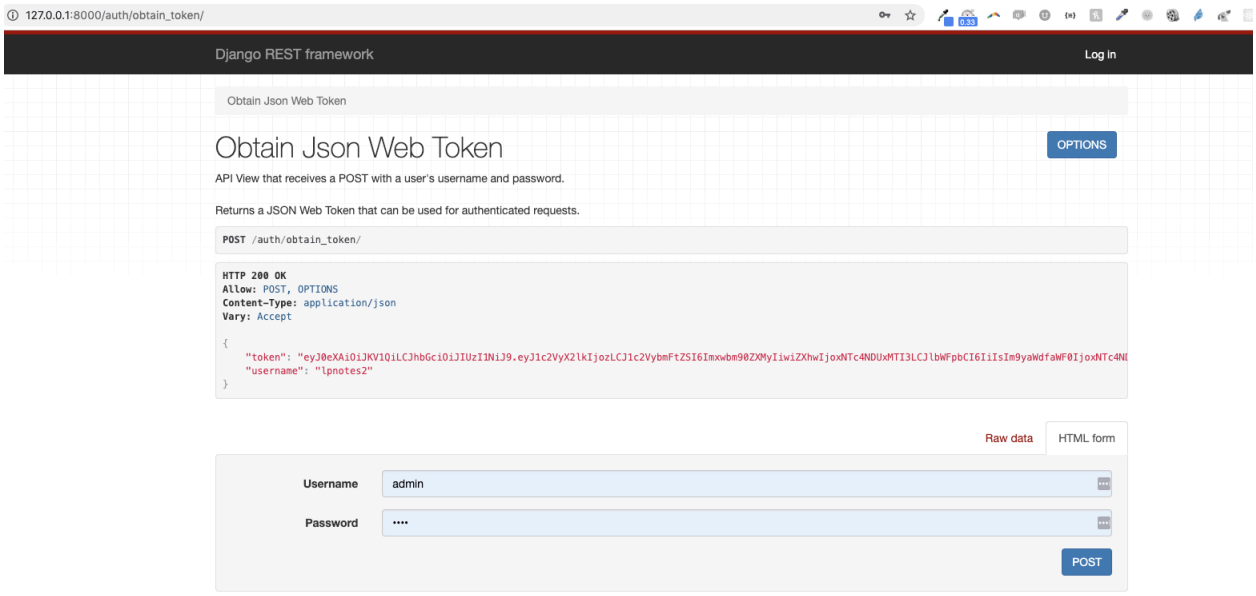


-

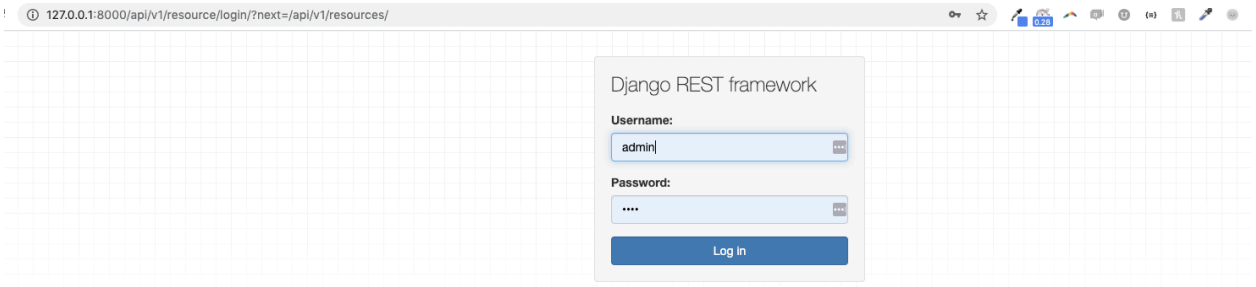
Get token:

- http://127.0.0.1:8000/auth/obtain_token/





Login on the DRF page with the creds from before:



Go back to the /resources page:

http://127.0.0.1:8000/api/v1/resources/

Walkthrough example

- Add field to models.py
- python manage.py makemigrations
- python manage.py migrate
- Foreign key example:

https://docs.djangoproject.com/en/3.0/ref/models/fields/#django.db.models.ForeignKey.on_delete

```
 2) Quit, and let me add a default in models.py
Select an option: 1
Please enter the default value now, as valid Python
The datetime and django.utils.timezone modules are available, so you can do e.g. timezone.now
Type 'exit' to exit this prompt
>>> exit
(cbv3) Lindas-MacBook-Pro-2:cbv3_django_prototype lpnotes$ python manage.py makemigrations
Did you rename resource.referrer to resource.referring_url (a URLField)? [y/N] y
Did you rename resource.credit to resource.referring_user (a CharField)? [y/N] y
You are trying to add a non-nullable field 'user' to resource without a default; we can't do that (th
Please select a fix:
 1) Provide a one-off default now (will be set on all existing rows with a null value for this column
 2) Quit, and let me add a default in models.py
Select an option: 1
Please enter the default value now, as valid Python
The datetime and django.utils.timezone modules are available, so you can do e.g. timezone.now
Type 'exit' to exit this prompt
>>> lpnotes2
Invalid input: name 'lpnotes2' is not defined
>>> 'lpnotes2'
Migrations for 'resources':
  cbv3_django_prototype/resources/migrations/0005_auto_20200107_1818.py
    - Rename field referrer on resource to referring_url
    - Rename field credit on resource to referring_user
    - Add field user to resource
(cbv3) Lindas-MacBook-Pro-2:cbv3_django_prototype lpnotes$ python manage.py makemigrations
No changes detected
(cbv3) Lindas-MacBook-Pro-2:cbv3_django_prototype lpnotes$ python manage.py migrate
```

How to back out of a migration

python manage.py showmigrations

734

You can revert by migrating to the previous migration.

For example, if your last two migrations are:

- `0010_previous_migration`
- `0011_migration_to_revert`

Then you would do:

```
./manage.py migrate my_app 0010_previous_migration
```

You can then delete migration `0011_migration_to_revert`.

If you're using Django 1.8+, you can show the names of all the migrations with

```
./manage.py showmigrations my_app
```

To reverse all migrations for an app, you can run:

```
./manage.py migrate my_app zero
```

share edit flag

edited Jan 31 '17 at 16:05

nsg
9,718 ● 4 ● 17 ● 27

answered Aug 20 '15 at 17:00

Alasdair
210k ● 30 ● 369 ● 362

python manage.py resources previous_migration

Delete the file in Desktop/django-concept/cbv3_django_prototype/cbv3_django_prototype/resources/migrations

rm -rf 0005_auto_20200107_1828.py

Go back up three levels.

Re-run python manage.py makemigrations

git pull upstream issue-24

```
 [X] 0010_auto_20190429_0326
 [X] 0011_auto_20190508_0153
resources
 [X] 0001_initial
 [X] 0002_auto_20190924_0428
 [X] 0003_auto_20190924_2157
 [X] 0004_resource_author
 [ ] 0005_auto_20200107_1818
sessions
 [X] 0001_initial
sites
 [X] 0001_initial
 [X] 0002_alter_domain_unique
 [X] 0003_set_site_domain_and_name
socialaccount
 [X] 0001_initial
 [X] 0002_token_max_lengths
 [X] 0003_extra_data_default_dict
userauth
 (no migrations)
users
 [X] 0001_initial
(cbv3) Lindas-MacBook-Pro-2:cbv3 django_prototype
```

January 6th, 2020

Follow up work on docker commands:
[ ] Create a docker command that brings run locally a postgres image in a container and let's me log into that instance from a local termina with postgres commands.
[ ] Create issue

Work session:

1. From Postgres Docker _hub:
    a. Copied: `docker run --name some-postgres -e POSTGRES_PASSWORD=mysecretpassword -d postgres`
        i. Terminal result: successful container after docker ps was run.
        ii. Running from hackernoon:

```
1. psql -h localhost -U postgres -d postgres
2. Result:
      a. psql: /usr/pgsql-11/lib/libpq.so.5: no version
         information available (required by psql)
      b. psql: /usr/pgsql-11/lib/libpq.so.5: no version
         information available (required by psql)
      c. psql: could not connect to server: Connection refused
      d.        Is the server running on host "localhost"
         (::1) and accepting
      e.        TCP/IP connections on port 5432?
      f. could not connect to server: Connection refused
      g.        Is the server running on host "localhost"
         (127.0.0.1) and accepting
      h.        TCP/IP connections on port 5432?
      i.
      j.
```

b. Changing to:
    i.

Current issue:
`docker run --rm  --name pg-docker -e POSTGRES_PASSWORD=docker -d -p 5432:5432 -v $HOME/docker/volumes/postgres:/var/lib/postgresql/data  postgres` creates a hash that seems to confirm that I ran a container, but `docker ps` does not show any container present. I need to break down the command and then build it back up to both run a c®ontainer and then log into it remotely with a pg command.

Resources:
https://hub.docker.com/_/postgres # official docs on postgres images
https://hackernoon.com/dont-install-postgres-docker-pull-postgres-bee20e200198 # the blog post Linda and Bethany are using to bring up postgres, dockerized.

History:
From Codebuddies slack #codebuddies-meta:

> **linda (she/her)** 10:04 PM
> Pairing with @bengineerdavis on Django Rest Framework now — open to anyone to observe/hang out!
>
> https://codebuddies.org/hangout/LMvComuYtL3aTyGLR
>
> **CodeBuddies**
>
> **Goals: CB V3 RESTful api endpoint fields and resources**
>
> Main: Add 1 field to API endpoints. Stretch Goal(s) 1. 'Add new endpoint for POSTing a resource' 2. Consider reviewing the Django backend for

documenting purposes 3. 'Make sure instructions are clear for setting up the Django repo'(515 kB)

h

ttps://codebuddies.org/images/jitsi-example2.jpg

1 dj 1

linda (she/her) 11:39 PM

Finally solved those `DATABASE_URL` not found errors by doing `export DATABASE_URL=postgres://lpnotes:postgres@127.0.0.1:5432/postgres` (I was missing that `export` last time I tried to solve it

2

Saturday, January 4th

linda (she/her) 12:42 AM
Notes from this session:
https://docs.google.com/document/d/1YuVO-v0n73ogoFIwpwJgI1Bkso8sP2mg5zqbX9FB3lU/edit#heading=h.xggkalubufo3

linda (she/her) 12:43 AM
Thanks @bengineerdavis and @Phil aka tgrrr (he/him) for participating!

1

1 reply
3 days ago
View thread

Yesterday

bengineerdavis 7:03 PM
What do you folks need from me in so I can join the CodeBuddies org and contribute PRs? (edited)

5 replies
Last reply today at 10:51 PM
View thread

bengineerdavis 7:40 PM
Linda, when you get the chance, I also want to know if you need me to make an issue around standardizing/confirming the Dockerized Postgres container for dev environment.

3 replies
Last reply 24 hours ago
View thread

bethanyg 8:13 PM
@bengineerdavis - You should be able to submit a PR without having to be a member of the org.
If you've forked the repo, these directions should work: https://help.github.com/en/github/collaborating-with-issues-and-pull-requests/creating-a-pull-request-from-a-fork.
Please also open an issue in the corresponding codebuddies repo with a description of what you are working on, and assign that issue to yourself so that everyone knows what you're up to. Making a branch named after the issue is also a good strategy to help clarify what is in your PR when you submit it.

1 1

8:15 PM

For postgres, the following image was pulled from dockerhub: https://github.com/docker-library/postgres/blob/0d0485cb02e526f5a240b7740b46c35404aaf13f/11/Dockerfile (**11.3 (Debian 11.3-1.pgdg90+1)**) so any standardization work should begin with that image. (*And yes - the README is out of date and needs to be re-written*) The current version of postgres we're developing against is `11.3` (see the version above)- but the `11.6` that is in the repo should be fine. I would not use `12` since postgres doesn't guarantee backward compatibility with major versions. (edited)

✔️ 1

linda (she/her) 10:07 PM

CBV3 current status:

[ ] @bengineerdavis is setting up the github.com/codebuddies/django-concept repo and writing up a list of issues encountered re: postgres installation on Linux, and might help with documenting setup instructions in general

[ ] @bethanyg and I are figuring out why the register new user endpoint is not working for Django and JWT (based on https://medium.com/@dakota.lillie/django-react-jwt-authentication-5015ee00ef9a). Our code so far: https://github.com/BethanyG/django-concept/commit/4003222dc2a1fc441ac6bf21b03f1bad6a51ae23

[ ] @peoray is taking a look at form validation for https://github.com/codebuddies/react-concept/issues/55. (If you want to pair on thiis, @peoray, let me know!)

[ ] There's still a lot of user experience / design prototyping work left to do on https://github.com/codebuddies/react-concept, which should be fun

[ ] At the same time, we would slowly get interested people onboarded in terms of adding to the Django Rest Framework API that the React project would be making API requests to (edited)

👍 2

2 replies

Last reply today at 9:31 PM

View thread

bengineerdavis 11:04 PM

@bethanyg thank you for the head's up and directions, I'll get on it. Also thank you @linda (she/her) for the follow up summary. (edited)

---

https://github.com/codebuddies/django-concept/

January 3rd, 2020

Participants in the hangout:

- linda
- ben (bengineerdavis)
- phil (tgrrr)

https://cookiecutter-django.readthedocs.io/en/latest/developing-locally.html

export DATABASE_URL=postgres://postgres:docker@127.0.0.1:5432/postgres

PR (draft) for setting up docker:
https://github.com/codebuddies/django-concept/pull/15/files

psql -h localhost -U lpnotes -d postgres

docker ps
//see which docker images are running

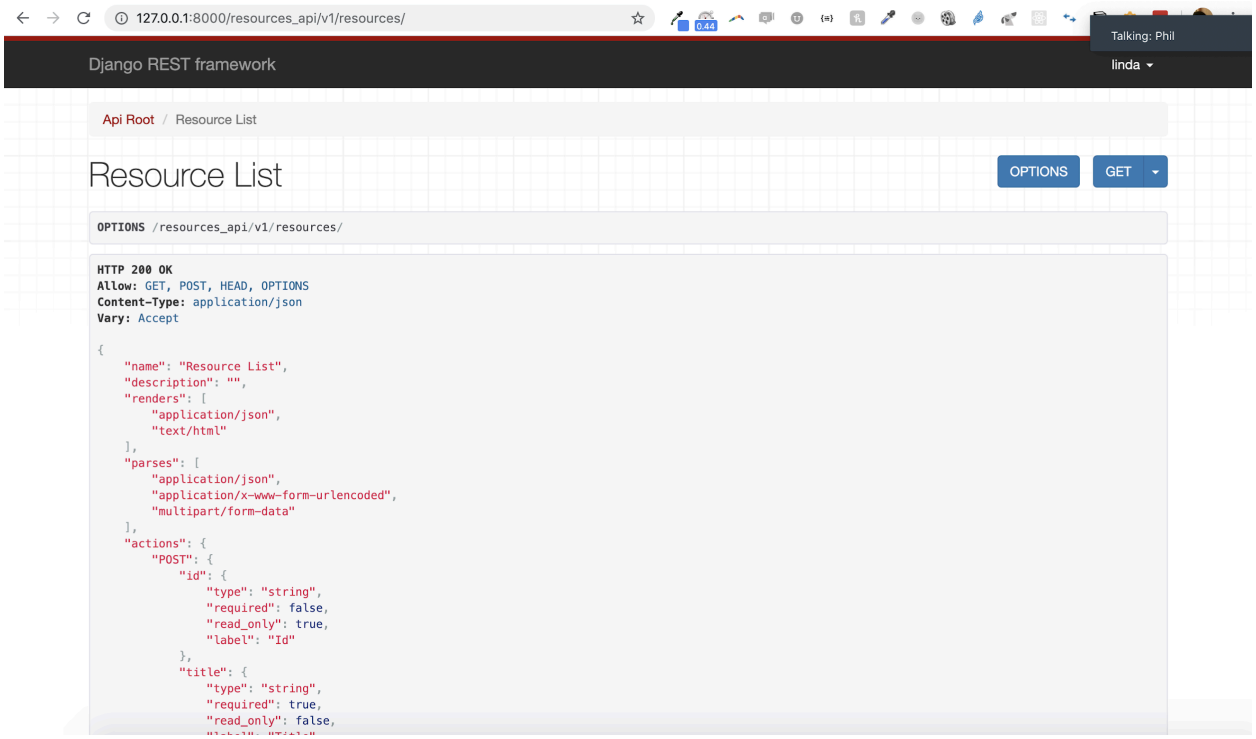
```
docker run --rm   --name pg-docker -e POSTGRES_PASSWORD=docker -d -p 5432:5432 -v $HOME/docker/volumes/postgres:/var/lib/postgresql/data  postgres
```

Publish or expose port (-p, --expose)

```
$ docker run -p 127.0.0.1:80:8080/tcp ubuntu bash
```

This binds port `8080` of the container to TCP port `80` on `127.0.0.1` of the host machine. You can also specify `udp` and `sctp` ports. The Docker User Guide explains in detail how to manipulate ports in Docker.

Localhost:
http://127.0.0.1:8000/resources_api/v1/resources/



December 19th, 2019 8:30pm EST

Hangout link: https://codebuddies.org/hangout/MDtdEDa6FgBuA742L#
Participants in the hangout:

- linda
- bethany
- kev (foodogx300)

Goal:
Figure out auth using Django and how to return it for the React front-end to use

Reference:
https://medium.com/@dakota.lillie/django-react-jwt-authentication-5015ee00ef9a

Technical Decisions

- We're going to use token authentication instead of session authentication because session authentication is appropriate for AJAX clients that are running in the same session context as your website.
- Do we want the backend or the front-end to be issuing tokens?
    - Decision: the backend should issue it.
        - One endpoint /auth-token would verify tokens POSTed to it

PRs done

- https://github.com/codebuddies/react-concept/pull/65

TODO

- Django
    - Ask a new contributor to set up https://github.com/codebuddies/django-concept/tree/bethany/cbv3_django_prototype, and where they get stuck
    - Update the contributing.md notes from that
    - Potentially two setup approaches:
    - docker
        - more manual instructions
    - Open questions:
        - Is the **master** branch on django-concept broken for anyone else?
        - Any Django folks who'd be willing to review a PR?
- React
    - Finish up https://github.com/codebuddies/react-concept/pull/65

Specs

- we want varying levels of permissions
- https://www.django-rest-framework.org/api-guide/authentication/

How to start up the app

- $ source cbv3/bin/activate
- docker run --rm --name pg-docker -e POSTGRES_PASSWORD=docker -d -p 5432:5432 -v $HOME/docker/volumes/postgres:/var/lib/postgresql/data postgres
  - https://hackernoon.com/dont-install-postgres-docker-pull-postgres-bee20e200198
  - OR
    - `docker run --name some-postgres -v /my/own/datadir:/var/lib/postgresql/data -d postgres:tag`
    - https://hub.docker.com/_/postgres ("**Where to Store Data**")

```
(cbv3) Lindas-MacBook-Pro-2:cbv3_django_prototype lpnotes$ psql -h localhost -p 5432 -d postgres
psql (11.5)
Type "help" for help.

postgres=# show tables;
ERROR:  unrecognized configuration parameter "tables"
postgres=# \dt
Did not find any relations.
postgres=# \l
                                  List of databases
         Name          |  Owner   | Encoding | Collate | Ctype |  Access privileges
-----------------------+----------+----------+---------+-------+---------------------
 cbv3                  | lpnotes | UTF8     | C       | C     |
 cbv3_django_prototype | lpnotes | UTF8     | C       | C     |
 postgres              | lpnotes | UTF8     | C       | C     |
 template0             | lpnotes | UTF8     | C       | C     | =c/lpnotes         +
                       |          |          |         |       | lpnotes=CTc/lpnotes
 template1             | lpnotes | UTF8     | C       | C     | =c/lpnotes         +
                       |          |          |         |       | lpnotes=CTc/lpnotes
(5 rows)
```

```
postgres=# \q
(cbv3) Lindas-MacBook-Pro-2:cbv3_django_prototype lpnotes$ psql -h localhost -p 5432 -d cbv3_django_prototype
psql (11.5)
Type "help" for help.

cbv3_django_prototype=# \dt
                    List of relations
 Schema |                Name                 | Type  |  Owner
--------+-------------------------------------+-------+---------
 public | account_emailaddress                | table | lpnotes
 public | account_emailconfirmation           | table | lpnotes
 public | auth_group                          | table | lpnotes
 public | auth_group_permissions              | table | lpnotes
 public | auth_permission                     | table | lpnotes
 public | django_admin_log                    | table | lpnotes
 public | django_celery_beat_clockedschedule  | table | lpnotes
 public | django_celery_beat_crontabschedule  | table | lpnotes
 public | django_celery_beat_intervalschedule | table | lpnotes
 public | django_celery_beat_periodictask     | table | lpnotes
 public | django_celery_beat_periodictasks    | table | lpnotes
 public | django_celery_beat_solarschedule    | table | lpnotes
 public | django_content_type                 | table | lpnotes
 public | django_migrations                   | table | lpnotes
 public | django_session                      | table | lpnotes
 public | django_site                         | table | lpnotes
 public | resources_resource                  | table | lpnotes
```

psql -h localhost -p 5432 -d postgres
psql -h localhost -p 5432 -d cbv3_django_prototype

```
(cbv3) Lindas-MacBook-Pro-2:cbv3_django_prototype lpnotes$ DATABASE_URL="postgres://postgres:docker@localhost:5432/cbv3_django_prototype
"
```

In the terminal, put:
DATABASE_URL="postgres://postgres:docker@localhost:5432/cbv3_django_prototype"

Errors:

```
        modified:   ../cbv3/lib/python3.7/site-packages/importlib_metadata/__pycache__/__init__.cpython-37.pyc
        modified:   ../cbv3/lib/python3.7/site-packages/importlib_metadata/__pycache__/_compat.cpython-37.pyc

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        ../.vscode/
        ../cbv3/bin/pipenv
        ../cbv3/bin/pipenv-resolver
        ../cbv3/bin/virtualenv
        ../cbv3/bin/virtualenv-clone
        ../cbv3/lib/python3.7/site-packages/__pycache__/clonevirtualenv.cpython-37.pyc
        ../cbv3/lib/python3.7/site-packages/__pycache__/virtualenv.cpython-37.pyc
        ../cbv3/lib/python3.7/site-packages/clonevirtualenv.py
        ../cbv3/lib/python3.7/site-packages/importlib_metadata-0.20.dist-info/
        ../cbv3/lib/python3.7/site-packages/ipython-7.8.0.dist-info/
        ../cbv3/lib/python3.7/site-packages/pipenv-2018.11.26.dist-info/
        ../cbv3/lib/python3.7/site-packages/pipenv/
        ../cbv3/lib/python3.7/site-packages/snowballstemmer-1.9.1-py3.7.egg-info/
        ../cbv3/lib/python3.7/site-packages/snowballstemmer/basque_stemmer.py
        ../cbv3/lib/python3.7/site-packages/snowballstemmer/catalan_stemmer.py
        ../cbv3/lib/python3.7/site-packages/snowballstemmer/hindi_stemmer.py
        ../cbv3/lib/python3.7/site-packages/virtualenv-16.7.5.dist-info/
        ../cbv3/lib/python3.7/site-packages/virtualenv.py
        ../cbv3/lib/python3.7/site-packages/virtualenv_clone-0.5.3.dist-info/
        ../cbv3/lib/python3.7/site-packages/virtualenv_support/
```

Postgres 11 on a Docker container

What is the difference between token and session auth?
https://dev.to/vasilevskialeks/token-vs-session-authentication-56ed
https://developer.okta.com/blog/2017/08/17/why-jwts-suck-as-session-tokens

- session cookies are small; web tokens are bigger