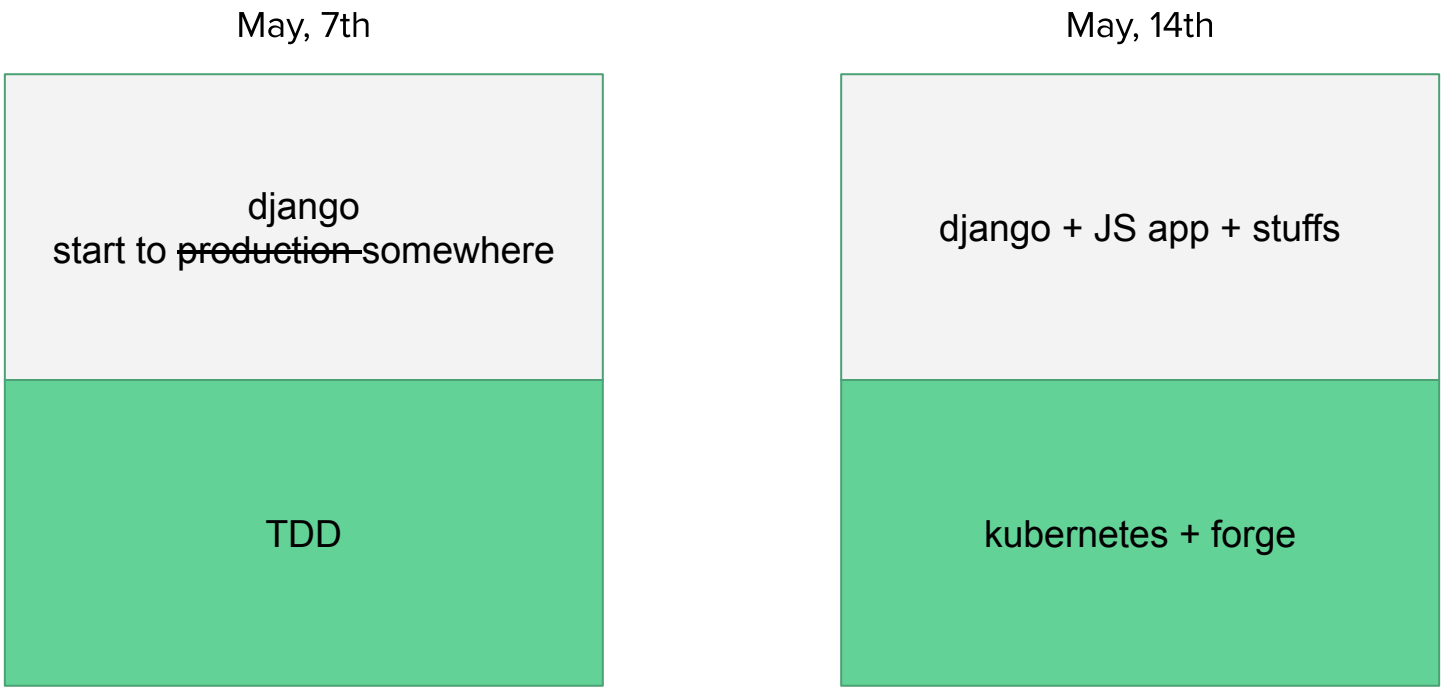
May, 7th
django
start to ~~production~~ somewhere
TDD
May, 14th
django + JS app + stuffs
kubernetes + forge

django, react, stuff

May 13th, 2019

This talk

github.com/horosin/django-cc-live

(slides, code, etc.)

The plan

1. Django - rounding up
2. User auth
3. Plotting
4. React in containers
5. Other tips

Set up homepage

```
from django.views.generic import RedirectView

urlpatterns = [
    path('', RedirectView.as_view(url='samples/list')),
    …
]
```

Authorization

Docs

https://docs.djangoproject.com/en/2.2/topics/auth/

User-related views

```
path('accounts/', include('django.contrib.auth.urls')),
```

http://localhost:8000/accounts/login/

...missing template

Create new folder:

app/templates/registration

And file:

app/templates/registration/login.html

Login template

Official docs is the source:

https://docs.djangoproject.com/en/2.2/topics/auth/default/#django.contrib.auth.views.LoginView

```
LOGIN_REDIRECT_URL = '/samples/'
LOGOUT_REDIRECT_URL = '/accounts/login'
```

Add middleware to secure all views

https://stackoverflow.com/a/46976284

```
LOGIN_REQUIRED_URLS = (
    r'(.*)',
)
LOGIN_REQUIRED_URLS_EXCEPTIONS = (
    r'/admin(.*)$',
    r'/accounts(.*)$',
)
```

User-dependent links

```
{% if user.is_superuser %}
<li class="nav-item">
  <a class="nav-link" href="{% url 'admin:index' %}">Admin</a>
</li>
{% endif %}


<ul class="nav navbar-nav ml-auto">
  {% if user.is_authenticated %}
  <li class="nav-item mr-1">Witaj, {{ user.get_username }} </li>
  <li class="nav-item"><a href="{% url 'logout'%}?next={{request.path}}">wyloguj</a></li>
  {% else %}
  <li class="nav-item"><a href="{% url 'login'%}?next={{request.path}}">zaloguj</a></li>
  {% endif %}
</ul>
```

Automatically set user

https://docs.djangoproject.com/en/2.2/topics/class-based-views/generic-editing/#models-and-request-user

```
def form_valid(self, form):
    form.instance.created_by = self.request.user
    return super().form_valid(form)
```

Managing permissions

Create View

Class

```
class SampleCreateView(generic.CreateView):
   model = Sample
   fields = '__all__'
   success_url = reverse_lazy('index')
```

FORM

```
   <form method="post">{% csrf_token %}
       {{ form.as_p }}
       <input type="submit" value="Save">
   </form>
```

Create form - bootstrap

```
pip install django-crispy-forms

INSTALLED_APPS = [
    # [...]
    'crispy_forms'
]

CRISPY_TEMPLATE_PACK = 'bootstrap4'

{% load crispy_forms_tags %}

{{ form|crispy }}
```

Documentation

https://docs.djangoproject.com/en/2.2/topics/auth/default/#permissions-and-authorization

Django default permissions

See in admin > users

foo - app name
bar - model name

- add: 'foo.add_bar'
- change: 'foo.change_bar'
- delete: 'foo.delete_bar'
- view: 'foo.view_bar'

Checking permissions

Function views

from django.contrib.auth.decorators import permission_required

@permission_required('samples.add_sample)

Class Based Views

from django.contrib.auth.mixins import PermissionRequiredMixin

permission_required = 'samples.view_sample'

Plotting

Let’s plot test

https://plot.ly/javascript/getting-started/

```
<script src="https://cdn.plot.ly/plotly-latest.min.js"></script>


document.addEventListener("DOMContentLoaded", function(event) {
     PLOTLY SAMPLE
})
```

Transfering the data

1. (BEST) HTTP request to API from the page
2. (DIRTY) JSON embedded in django template

DIRTY:

```
import json

json.dumps(var)
```

Simple API endpoints

```
from django.http import JsonResponse

return JsonResponse({'foo': 'bar'})
```

https://docs.djangoproject.com/en/2.2/ref/request-response/#jsonresponse-objects

Making requests from the browser

```
fetch('http://localhost:8000/samples/plotdata')
  .then(function(response) {
    return response.json();
  })
  .then(function(myJson) {
    // plot the data
  });
```

React and docker

Dockerfile - start app

```
FROM node:10-alpine

WORKDIR /usr/src/app

RUN npm install -g create-react-app

CMD tail -f /dev/null
```

Compose

```
services:
 ...
 client:
   build:
     context: client
   volumes:
     - './client:/usr/src/app'
     - '/usr/src/app/node_modules'
   ports:
     - '3000:3000'
   environment:
     - NODE_ENV=development
```

Start app

```
docker-compose exec client sh
```

```
create-react-app .
```

Dockerfile - for development

```
FROM node:10-alpine

WORKDIR /usr/src/app

# install and cache app dependencies
COPY package.json .
RUN npm install --silent

# copy app files
COPY . .

# start app
CMD ["npm", "start"]
```

Other tips

Django Rest Framework

1. Standard for making REST APIs in django
2. A lot of features out of the box - different formats of responses, serializing models, permissions, api documentation, SWAGGER, lots more
3. https://www.django-rest-framework.org/

Generating pdfs

Great client side library (JS)

jsPDF

https://github.com/MrRio/jsPDF

Loading excel files etc.

1. Send file via normal form or django form
2. Parse using your favourite python excel library
3. Insert rows into DB using django model API

Q&A