Opening Remarks Day 1

Lemurs because of Duke Lemur research

logistics

youtube.com/DjangoConUS

Keynote: Finding Purpose in Open Source through Community Building

Got into open source via Django Girls

Speaker was able to mentor someone by slowly giving them Pyladies tasking.

Gives more examples of mentoring where she shows that she tailored her mentoring to the person's needs and strengths. Sometimes she holds their hand more and other times gives more independance. Also shows how folks don't have to be engineers to contribute to Django/open source.

Speaker also talks about how each person brings unique and useful skills to the efforts.

Challenges
- need to find opportunities for junior devs
- it can be key to pay open source contributors
- finding resources for the Django Girls/PyLadies Ghana
- travel and visas - the paperwork can make it hard for folks to contribute and attend conferences if they are coming from around the world.

Supercharging your Django Dev Team: Introducing the Best Framework

they wanted to do a quick advertisement for tomorrow's talk

Online Bonus Talk - Automate Your City Data with Python

We do not have a lot of data about how our city councils work. And it doesn't help that they are PDFs.

Goal - get the civic data into a normalized, searchable, quueryable form

Uses tool called Datasette

want to make sure each step can be double-checked

HTML-ivating your Django web app's experience with HTMX, AlpineJS, and streaming HTML.

Usually you're going to a SPA - Single Page Application

There is a concern about the overuse of the SPA pattern

SPAs are increasing complexity - perhaps without analysis

5 Components to make a great experience
- remove whole-page refreshes from every interactoin
- use small payloads from server to update interface
- update HTML as result of changes in data
- empower rich on-page interactoins
- be fast

django-unicorn

HTMX plus AlpineJS - this gives the experience of a SPA Application

You can use HTML attributes to control the behavior which gives you the benefit of locality

also gives much faster rendering time compared to SPA application

SPA applications have much longer times to boot up - and percieved speed is important to users/ customers

Dango 4.2 allows for StreamingHttpResponses

Need to split template into pieces and yield each part- speeds up rendering and also gives small areas for replacing - rather than the whole page

---

Vue + Django: Combining Django Templates and Vue Single File Components without compromise

they wanted the power of interactivity from a JS framework, but the rapid development and tight coupling of templates

Speaker able to combine Django Templates and Vue.JS

see slide "Rewards Demo App"

article to read: Django+Vue+Vite: REST not Required

Lightning Talks

Apr 8 2024 - Total Solar Eclipse
--
Innovating with AI to build Chat-like applications

data has quirks - may not be text. How do you make it readable for an llm?

--

Black Python Devs

important for python and django conerences to be in diverse areas. or if not, to make sure to have events

--

Durham Django Girls

--

django.social

they are talkinga out meetup fatigue and some meetups having started since COVID

--

I'm A progrmamer I can solve this

Using database triggers to reliably track model history

django-pghistory is used to do the stuff in this talk

First problem - where are the changes stored?

- JSON isn't a good choice because it's not easy to quarry it
- a structured history model? It diesn't good because you end up with a bunch of bespoke history models

Second problem - how to reliably detect changes?

- Signals? Bulk changes don't fire signals. Also there are race conditions
- you can override the save/create methods or use decorators, but you need to make sure this is done everywhere

pghistory works as a decorator
@pghistory.track() on your model classes

It evolves to match your models during migrations

The triggers live in the database, not django. So it doesn't have the signals issue mentioned above.

you can add free-form metadata to the events

this allows some different design patterns

you can track logged-in users

Beyond the Basics of Migrations

explains how migrations work

may need to tweak dependencies in the migration if two devs changed model before each other had done a migration

efficiency matters on large tables (especially with CI/CD)

Missed, but wanted to watch: Building Powerful APIs with Django, Django Rest Framework, and OpenAPI

Working with Neo4j with Django neomodel library

Neo4J is a native graph database - it is better when you hvave relationsihps (than a relational database)

for example - see slide "A Neo4j graph"

good for queries about hierarchical data

also good with path finding

Cypher is a querying language that works well with this type of database

Paradise Papers analyzed this way

Modern editing experience for your Django models with Wagtail

Wagtail is a CMS written on top of Django

A Page is both a model and a view

Talk is essentially about adding Wagtail into your Django project and then using the Wagtail UI to edit your models. (Vs using Wagtail as a CMS - Wordpress replacement)

Also allows for tracking Revisions to models

The evolution of a Django Website into a radio automation back-end

project has evolved over the last decade