Nothing For Us, Without Us; Breaking Unconcious Bias in Building Products

seek feedback for your products from folks different from the community developing it.

Opening Remarks

how to get involved - organize or volunteer - hello@djangocon.us

Today is deep dive day - only one track

Keynote: Testing Modern Web Apps Like a Champion

development and testing are 2 sides of the same coin

developed Bulldoggy app to remember all the things to remember w/ their dog - tutorial app

https://github.com/AutomationPanda/bulldoggy-reminders-app

PyTexas keynote - talks about the architecture. This talk focuses on test.

Testing challenges
- tests are slow
- brittle
- flaky
- don't make sense
- don't make money
- requires changing context

see testing pyramid slide

Speaker talks about how this is what we do, but it's not necesarrily the right thing, especially with Web Apps

every type of test provides some value - shouldn't shirk whole classes of tests

Modern Testing Goals

1. focus on fast feedback loops instead of certain types of tests
2. Make test development fast and painless as possible
3. Choose test tooling that complements dev workflows

use arrange-act-assert test patterns

can't we just use pytest or unittest?

pytest and unittest by themselves cannot test web apps.

What about Django's testing support?

the main limitation is that it can't test like a user

Tools for blackbox testing:

- UI - Selenium, Cypress, Playwrite
- API - Requests, Cypress, Playwrite
- Component - Storybook, Cypress, Playwrite

How to do Component testing?

Storybook is one of the better apps for this

API testing?

Playwrite does this well.

(slide shows you how you write a successful api login test)

Contract testing is API test as a unit test

What about Selenium?

intro/review to Selenium

make sure you always quit at the end or you end up with a bunch of zombie browsers that will crash your CI

Selenium pain points
1 - no auto waiting so tests are susceptible to flakiness

2 - slow performance because of browser bloat

3- only for browser automation; not a full test framework; does not support API or component testing

So, what about Cypress?

(see the slide about its benefits)

Front-end web devs have loved to use it for a long time. Bad thing - bindings for javascript only

Cypress pain points

1- Javascript only
2 - trapped in the browser

What about Playwright?

Modern test framework from MS

(see slide with what it works with)

Has bindings for Python!

the tests are a bit simpler than with Selenium. You get a page fixture to use in Pytest

(see comparison slide)

Test Automation University - best place ot learn about test automation and it's free
---

Django migrations, friend or foe? Optimize your Django migrations for faster testing

starts off talking about the standard migration process

Inside Out: My Journey of Understanding Inclusion

lawncare and hardware store stories about bias

Cultures and Communication Context

- High-context (indirect, contextual)
- Low-context (direct, explicit)

What's good in one can be negative in the other

eg patronizing or not trustworthy

using seasons can exclude half of the world - Summer of Code for Google is Winter in southern hemisphere

---

Django's Data Science Makeover: Integrating D3.js and Bokeh for Data Visualization

it can be as easy as adding a script tag to use D3.js in Django

Bokeh cna be published in a Jupyter Notebook; it's also interactive

Django Channels allow for real-time data streaming

How to Schedule Tasks with Celery and Django

Celery - distributed message proc system - focus on real-time processing - it's like cron for a large bunch of servers

some other things celery used for - generating asssets after upload, notifying users when an event happens, keeping search index up to date

celery beat requires state and there can only be ont beat at a time

long-running tasks are harder to debug, but extremely short tasks involve non-trivial messaging overhead. So you need to find the sweet spot.

One database table, one model, many behaviours: Proxy model

Model Inheritance helps to reduce complexity

Types of inheritance: abstract, mutl-table inheritance, and a third one speaker will get to later

abstrcat - you have some fields you want to use in multiple places. Uses a meta class with abstract=True

you can also creat proxy models.

Back to the Future of Hypermedia in Django

Hypermedia -> HTTP -> REST

HATEOAS - Hypermedia as the Engine of Application State

IN 2023.... NOT HATEOS - JSON as the Engine of Application State?!

HTMX is a Hypermedia library
    - allows for full AJAX set of actions

Patterns

django + htmx patterns (a readme on github)

- HTMX request headers
- rendering partial content

tools to help w/ these patterns

django-htmx - uses middleware to provide convenience tools for using htmx headers

template fragments - django-render-block helps with this

also django-template-partials package

another tool - hx-requests

forge packages

Django’s accessibility track record

the talk based on someone reporting to Wagtail that the admin interface was hard to use because user was blind

one thing you can do is turn off the colors (make the website grey scale) and see if this makes it harder to used

---

Panel Discussion: Who put me in charge? Moving beyond day-to-day coding in Django