

# Static Sites In Python

Kat Passen

# Before we begin

- Taxonomy - classification scheme
- Tag - metadata used to classify content
- SSG - static site generator
- CMS - content management system

"what is a static site"

"what is a static site"

=>

"what is JAMstack"

# JAMstack

- JavaScript
- APIs
- Markup



# What is **not** JAMstack

- Traditional server setup (Django)
- CMS-based site
- Single Page Application

# Why JAMstack

- Pages generated at deploy time, not during request
- All files (including HTML!) can live on CDN
- Can use a CMS alongside a JAMstack site: consolidate your stacks
- All server-side processes are micro services



# What is a static site

- HTML page with prepopulated content
- <https://www.staticgen.com/>

# In this talk:

- Lektor
- Statik
- Pelican
- Not things like Sphinx

# Lektor

<https://www.getlektor.com/>

# Lektor - Intro

- Partway between SSG and CMS
- Aimed more towards CMS-style content, but without the CMS server
- WYSIWYG editing with HTML output

# Lektor - Setup

```
$ curl -sf https://www.getlektor.com/  
install.sh | sh
```



```
/usr/bin/ruby -e "$(curl -fsSL https://  
raw.githubusercontent.com/Homebrew/  
install/master/install)"
```

```
$ pip install pelican
```

```
$ brew install hugo
```

```
$ gem install bundler jekyll
```

```
$ npm install --save-dev @11ty/eleventy
```

```
...
```

```
$ curl -sf https://www.getlektor.com/  
install.sh | sh
```



# Statik

<https://getstatik.com/>

# Statik - Intro

- Non-blog oriented SSG
- "what if Hugo but custom data models" (SQLAlchemy)
- Jinja2 templates

# Statik - Setup

```
$ python3 -m venv path/to/stuff/; source  
venv/bin/activate
```

```
$ statik --quickstart
```

(make some views)

```
$ statik --watch
```

# Statik - Detail

- Quickstart just kind of leaves you hanging
- Models and views are defined as YAML, data is defined as either YAML or Markdown
  - In your data, you can refer to model instances by primary key to autofill info
- Dynamic variables in views are just database queries
- Themes are separate: <https://github.com/thanethomson/statik-themes>

# Statik - Detail

Sample model: `models/Post.yml`

```
title: String
slug: String
author: Author      # This will be auto-detected as a foreign key reference
summary: String
published: DateTime
content: Content
```

Sample model: `models/Author.yml`

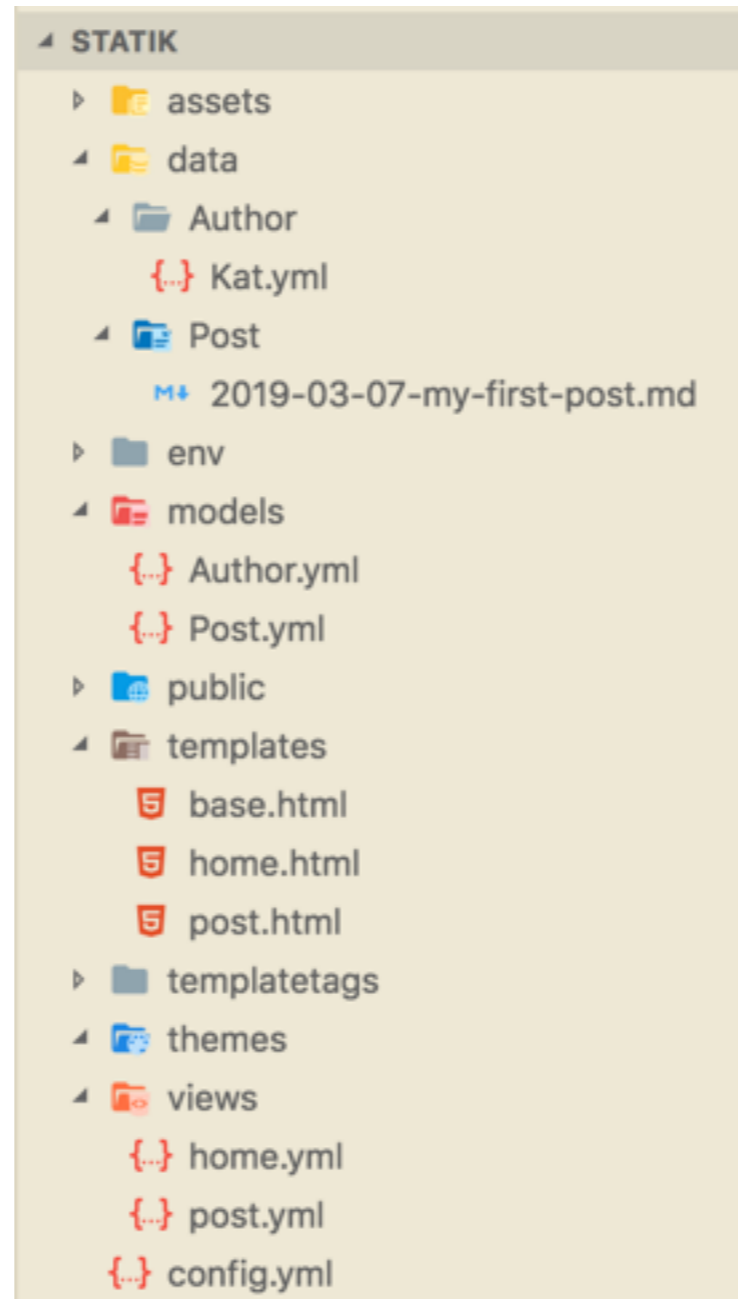
```
first-name: String
last-name: String
email: String
```

# Statik - Detail

Sample view: `views/post.yml`

```
path:  
  template: /{{ post.published|date("%Y/%m/%d") }}/{{ post.slug }}  
  for-each:  
    post: session.query(Post).filter(Post.published != None).all()  
template: post
```

# Statik - Results



# Pelican

<https://blog.getpelican.com/>



# Pelican - Intro

- Extremely blog-oriented - category taxonomy, chronological content
- "Pages" - non-content pages (about, contact)
- Content is reStructuredText or Markdown

# Pelican - Intro

- i18n support (thru plugins)
- Jinja2 templates
- Integrates with various third-party services (thru plugins)

# Pelican - Demo