Beheading Wagtail

Using Wagtail as a headless CMS

Slides available at goo.gl/V9PAXu

Michael Harrison

https://github.com/mwharrison

- Advocate for using open-source technology in education
- Backend Developer with OpenStax at Rice University, Houston, Texas.
- Formerly worked in central IT at Rice, introducing Django into their core services (search, graduate applications, conflict reporting)
- Working with Django since 1.1 (old habits die hard, like function based views...)

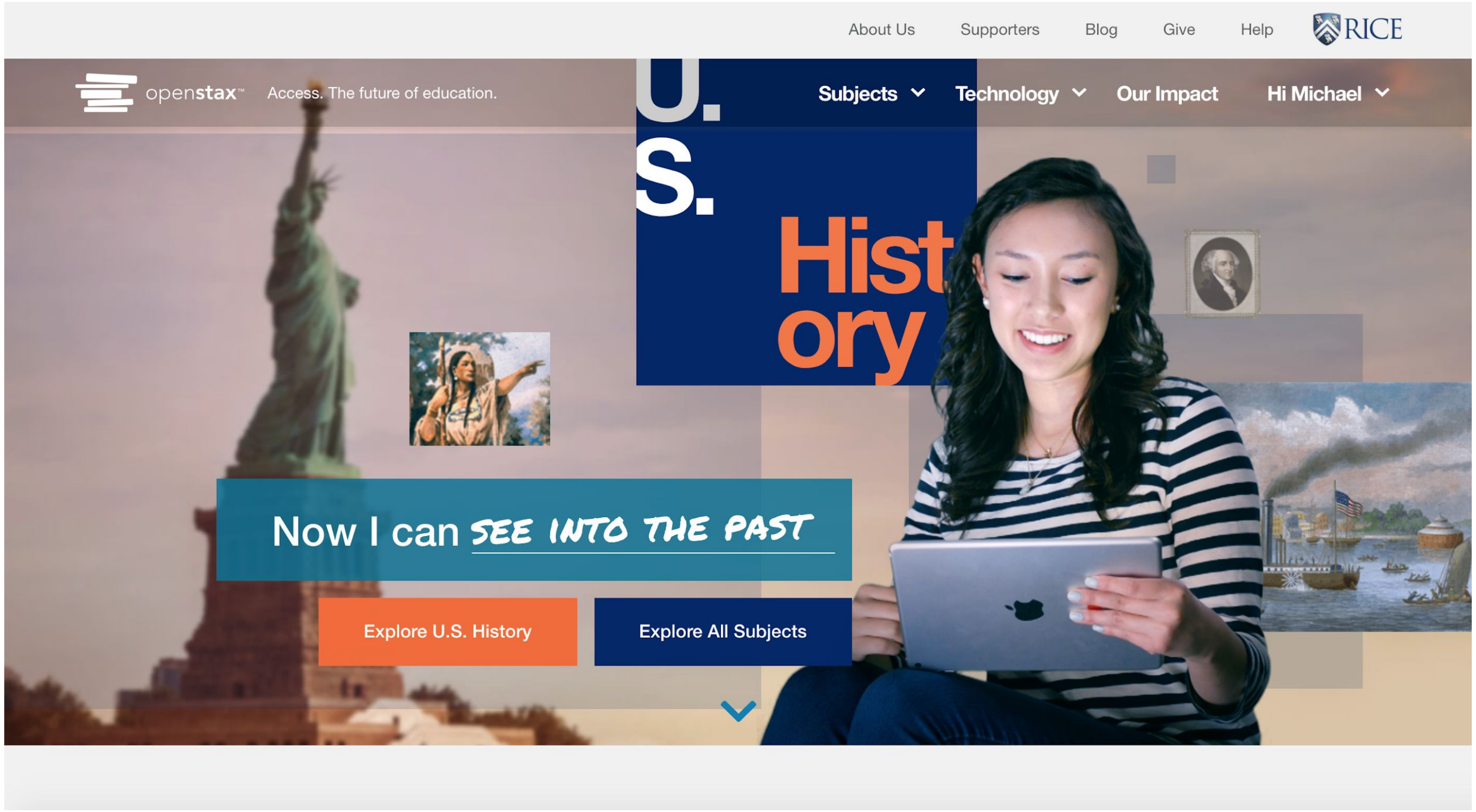
Slides available at goo.gl/V9PAXu

openstax™

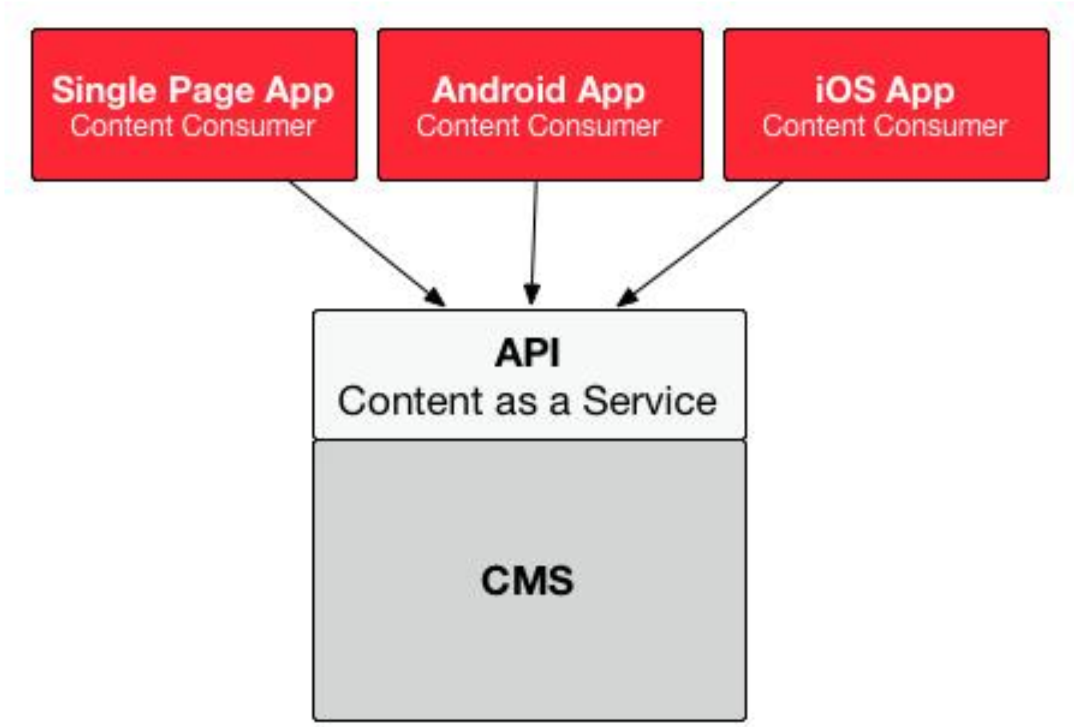
- Formally Connexions, launched in 1999 - changed to OpenStax in 2012.
- Focus on building openly-licensed textbooks, which are free in all digital formats.
- All textbooks are licensed under CC-BY - so you can use and modify our content as long as you attribute it to us.
- Focus on common-core college-level books.

https://openstax.org/

Slides available at goo.gl/V9PAXu



openstax.org- can’t even tell it’s missing its head!



What is headless?

Why did we choose Wagtail?

And why did we decide to go headless?

- Familiarity
- The editing interface
- Developer Friendly
- The ability to decouple our FE and BE
- Had a pre-build content API

openstax.org code

Backend

https://github.com/openstax/openstax-cms

- Runs Wagtail
- Using default admin, with (mostly vanilla) Wagtail API for content

Frontend

https://github.com/openstax/os-webview

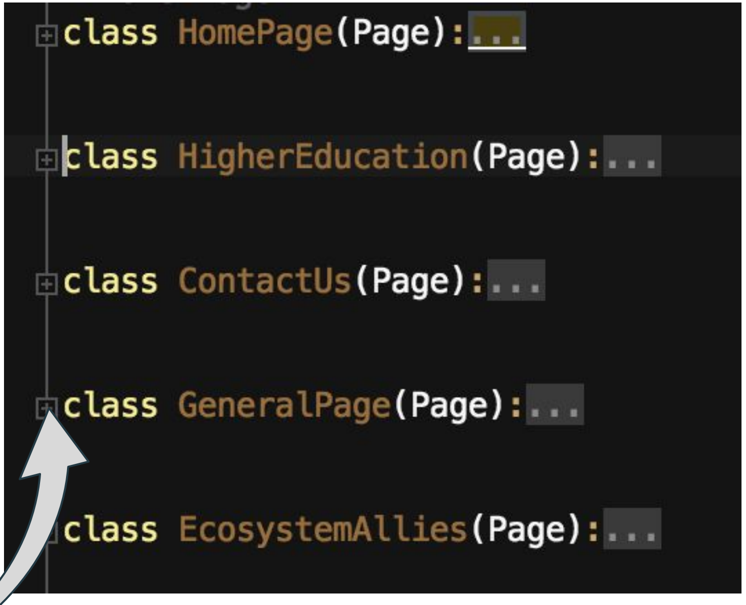
- custom javascript framework, written in-house
- based on superviews and incremental DOM

Pages, pages, everywhere

We define each page on the site as a subclass of Page.

I'm hoping to use StreamFields to solve this in the future, but for now this works for us.

/openstax
/accounts
/errata
/pages
/models.py
/books



```
class HomePage(Page):...

class HigherEducation(Page):...

class ContactUs(Page):...

class GeneralPage(Page):...

class EcosystemAllies(Page):...
```

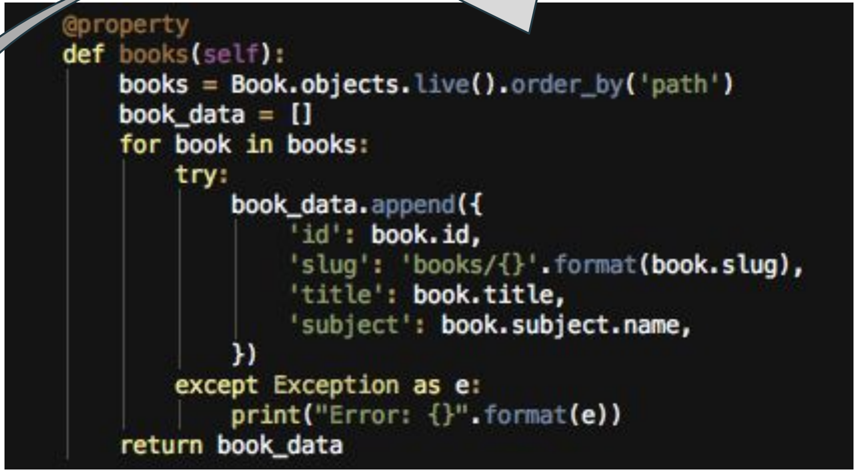
Slides available at goo.gl/V9PAXu

Returning Grouped/Sorted Pages in the API

Properties on models to return sets of data for use on other pages.

.order_by('path') allows us to maintain order from the admin resorting ability

http://openstax.org/api/books



```
@property
def books(self):
    books = Book.objects.live().order_by('path')
    book_data = []
    for book in books:
        try:
            book_data.append({
                'id': book.id,
                'slug': 'books/{}'.format(book.slug),
                'title': book.title,
                'subject': book.subject.name,
            })
        except Exception as e:
            print("Error: {}".format(e))
    return book_data
```

https://github.com/openstax/openstax-cms/blob/66e7639924d2411f275e7bf1a421eed89fbe4b00/books/models.py#L667

Slides available at goo.gl/V9PAXu

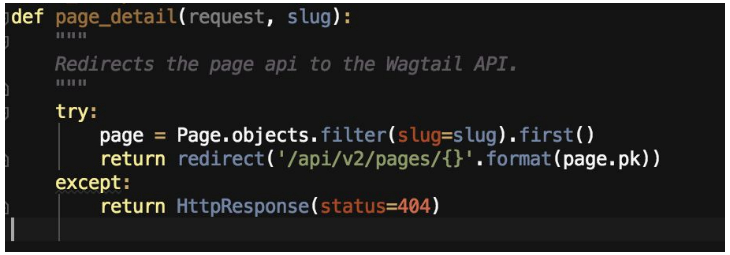
Returning Page by Slug

Allows us to redirect our FE requests to the proper API endpoint by slug.
This reduces HTTP calls - because the default way is to search by slug - the follow the details link.

https://openstax.org/api/pages/openstax-homepage/ 301-> https://openstax.org/api/v2/pages/29/

VS

https://openstax.org/api/v2/pages/?slug=openstax-homepage FE request-> `"detail_url": "https://openstax.org/api/v2/pages/29/"`



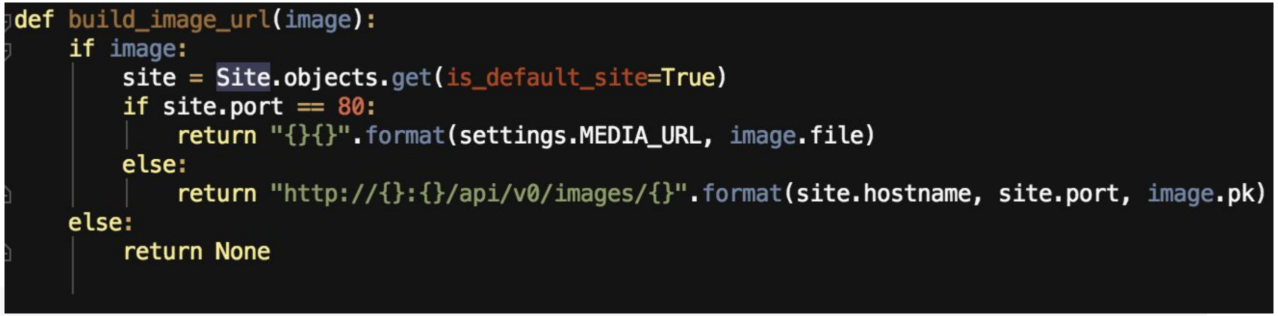
```python
def page_detail(request, slug):
    """
    Redirects the page api to the Wagtail API.
    """
    try:
        page = Page.objects.filter(slug=slug).first()
        return redirect('/api/v2/pages/{}'.format(page.pk))
    except:
        return HttpResponse(status=404)
```

https://github.com/openstax/openstax-cms/blob/master/pages/views.py

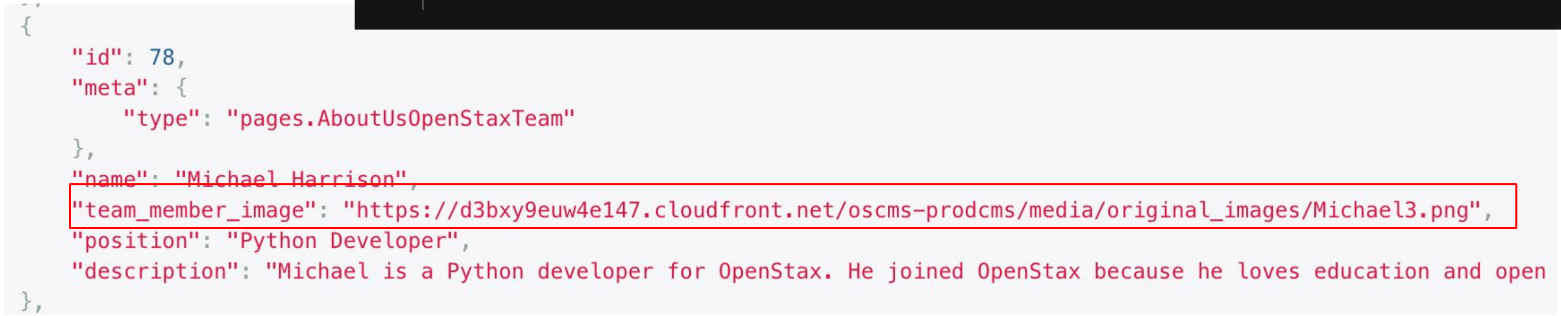
Slides available at goo.gl/V9PAXu

AWS with Wagtail API

Returning the media URL when the resource is on AWS



```python
def build_image_url(image):
    if image:
        site = Site.objects.get(is_default_site=True)
        if site.port == 80:
            return "{}{}".format(settings.MEDIA_URL, image.file)
        else:
            return "http://{}:{}/api/v0/images/{}".format(site.hostname, site.port, image.pk)
    else:
        return None
```



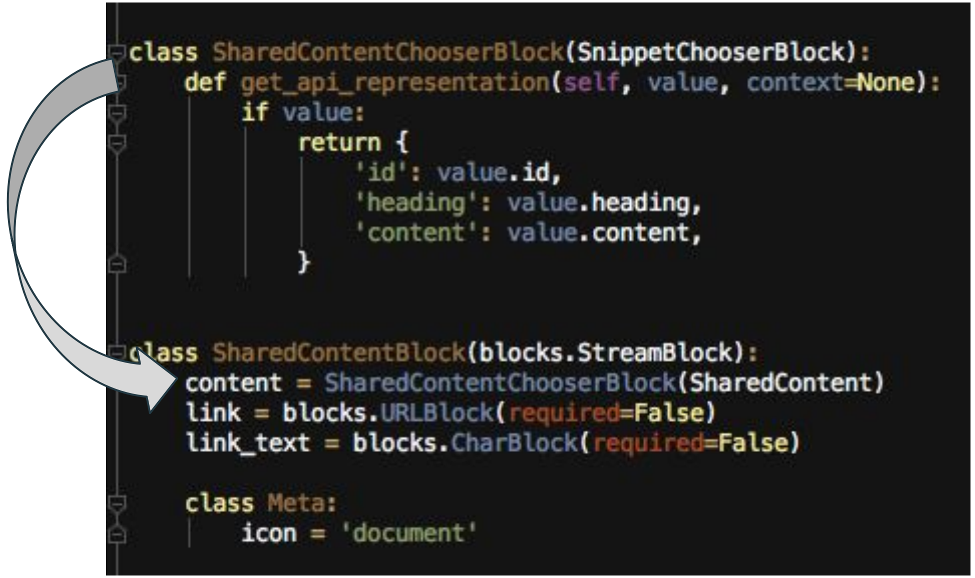
```json
{
    "id": 78,
    "meta": {
        "type": "pages.AboutUsOpenStaxTeam"
    },
    "name": "Michael Harrison",
    "team_member_image": "https://d3bxy9euw4e147.cloudfront.net/oscms-prodcms/media/original_images/Michael3.png",
    "position": "Python Developer",
    "description": "Michael is a Python developer for OpenStax. He joined OpenStax because he loves education and open
},
```

https://github.com/openstax/openstax-cms/blob/master/openstax/functions.py

Slides available at goo.gl/V9PAXu

Overriding API for Snippets

Sharing content across resources and customizing the API output.



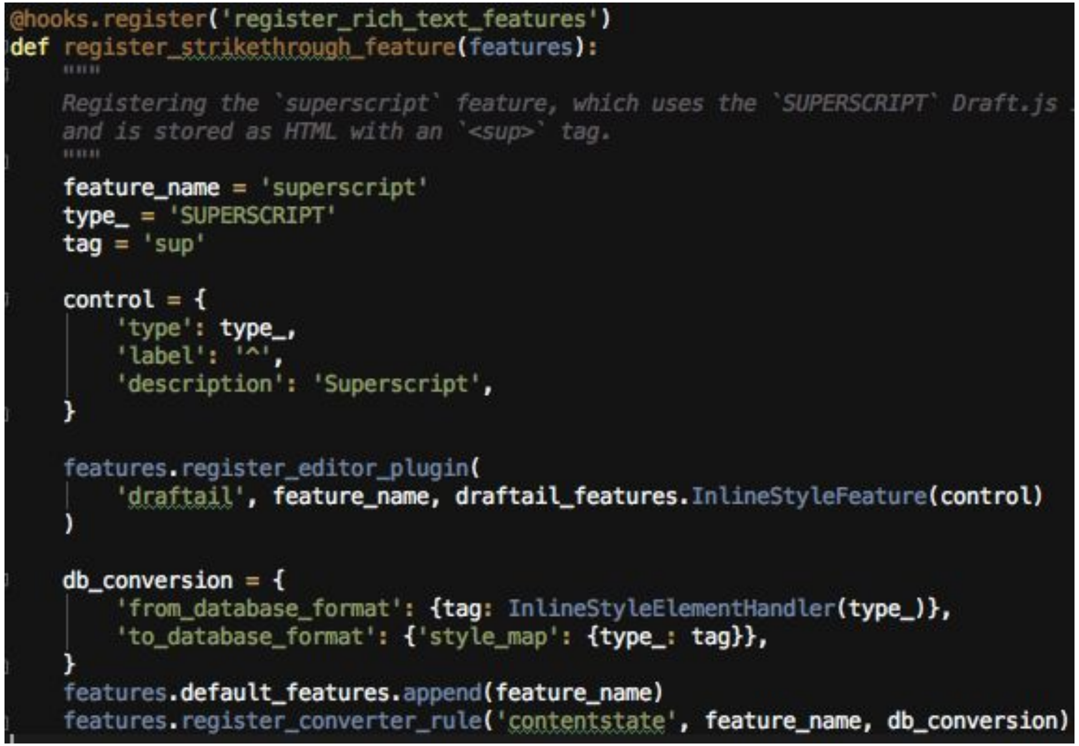
```
class SharedContentChooserBlock(SnippetChooserBlock):
    def get_api_representation(self, value, context=None):
        if value:
            return {
                'id': value.id,
                'heading': value.heading,
                'content': value.content,
            }


class SharedContentBlock(blocks.StreamBlock):
    content = SharedContentChooserBlock(SharedContent)
    link = blocks.URLBlock(required=False)
    link_text = blocks.CharBlock(required=False)

    class Meta:
        icon = 'document'
```

https://github.com/openstax/openstax-cms/blob/66e7639924d2411f275e7bf1a421eed89fbe4b00/books/models.py#L238

Slides available at goo.gl/V9PAXu

Adding Draftail Features

Custom HTML tag for superscript text.



```
@hooks.register('register_rich_text_features')
def register_strikethrough_feature(features):
    """
    Registering the `superscript` feature, which uses the `SUPERSCRIPT` Draft.js
    and is stored as HTML with an `<sup>` tag.
    """
    feature_name = 'superscript'
    type_ = 'SUPERSCRIPT'
    tag = 'sup'

    control = {
        'type': type_,
        'label': '^',
        'description': 'Superscript',
    }

    features.register_editor_plugin(
        'draftail', feature_name, draftail_features.InlineStyleFeature(control)
    )

    db_conversion = {
        'from_database_format': {tag: InlineStyleElementHandler(type_)},
        'to_database_format': {'style_map': {type_: tag}},
    }
    features.default_features.append(feature_name)
    features.register_converter_rule('contentstate', feature_name, db_conversion)
```

https://stackoverflow.com/questions/49264519/adding-superscript-to-draftail-in-wagtail-2-0/49266182#49266182
https://github.com/openstax/openstax-cms/blob/66e7639924d2411f275e7bf1a421eed89fbe4b00/global_settings/wagtail_hooks.py#L6

What’s next?

aka. What we want to improve

- StreamFields to replace our messy page models and RTFs everywhere
- Breaking components out more to make releasing easier.
- Not specific to headless CMS - but content migration between environments is a big pain point for us.

Questions?

Slides available at goo.gl/V9PAXu

mwharrison

mwharrison@rice.edu