STATIC SITES THROUGH SPAS AND GRAPHQL

Chris Biscardi

@chrisbiscardi
Design Systems and UI Platform @Dropbox

Low Earth Orbit

superawesomelabs/leo

STATIC SITES

- Templating
- Styling
- Data Model
- Universal Apps
- Routing
- PWAs
- Low Earth Orbit

TEMPLATING

liquid, jsx, handlebars, etc

LIQUID

```
{% assign coll = site.content %}
<ul class="breadcrumbs">
  <li><a href="{{site.baseurl}}/">Home</a></li>
  {% assign crumbs = page.url | split: '/' %}
  {% for crumb in crumbs offset: 1%}
  {% capture crumb_url %}{% assign crumb_limit = forloop.index | plus: 1 %}{% for crumb in crumbs limit: crumb_limit %}{{ crumb | append: '/' }}{% endfor %}{% endcapture %}
  {% capture site_name %}{% for p in coll %}{% if p.url == crumb_url %}{{ p.title }}{% endif %}{% endeor %}{% end capture %}
  {% endunless %}
  {% unless site_name == '' %}
  <li>
  {% unless forloop.last %}
    <a href="{{ site.baseurl }}{{ crumb_url | strip_newlines }}">{{site.name}}</a>
  {% else %}
    <span>{{ site_name }}</span>
  {% endunless %}
  </li>
  {% endfor %}
</ul>
```

JSX

```
<ul className="breadcrumbs">
  <li><a href={`${baseUrl}/`}>Home</a></li>
    {url && url.split("/").map((v, i, arr) => {
      const crumbURL = _.take(arr, i + 1).join("/");
      if (i === arr.length) {
        return <li><a href={crumbURL}>...</a></li>;
      } else {
        return <li><a href={`${baseUrl}${crumbURL}`}>{site.name}</a></li>;
      }
    })}
</ul>
```

REACT

```
class Breadcrumbs extends React.Component {
  static propTypes = {
    baseUrl: React.PropTypes.string,
    url: React.PropTypes.string
  };
  render() {
    return (
      <ul className="breadcrumbs">
        <li><a href={`${baseUrl}/`}>Home</a></li>
        {this.props.url && this.props.url.split("/").map((v, i, arr) => {
            const crumbURL = _.take(arr, i + 1).join("/");
            if (i === arr.length) {
              return <li><a href={crumbURL}>...</a></li>;
            } else {
              return <li><a href={`${baseUrl}${crumbURL}`}>{site.name}</a></li>;
            }
          })}
      </ul>
    );
  }
}
```

FLOW

```
const Breadcrumbs = ({ baseUrl, url }: { baseUrl: URL, url: URL }) => (
  <ul className="breadcrumbs">
    <li><a href={`${baseUrl}/`}>Home</a></li>
    {url && url.split("/").map((v, i, arr) => {
        const crumbURL = _.take(arr, i + 1).join("/");
        if (i === arr.length) {
          return <li><a href={crumbURL}>...</a></li>;
        } else {
          return <li><a href={`${baseUrl}${crumbURL}`}>{site.name}</a></li>;
        }
      })}
  </ul>
);
```

STYLING

CSS, SASS, CSS Modules, Glamor

CSS

```
.f1 { font-size: 1rem };
.red { background: red };
.f1.red { font-size: 2rem };

<div class="f1 red">
  I'm red and 2rem!
</div>
```

PROBLEMS WITH CSS AT SCALE

- Global Namespace
- Dependencies
- Dead Code Elimination
- Minification
- Sharing Constants
- Non-deterministic Resolution
- Isolation

- React: CSS in JS by vjeux

CSS MODULES

```
.f1 { font-size: 1rem };
.red { background: red };
.fired { composes: red; font-size: 2rem };

<div class="Thing__fired_h83sn">
  I'm red and 2rem!
</div>
```

GLAMOR (REPLICATING CSS)

```
const f1 = css({ fontSize: '1rem' });
const red = css({ background: 'red' });
const f1Red = css({
  [`&.${f1}.${red}`]: { fontSize: '2rem' }
});

// ...but you still have to use all three selectors
<div class={`${f1} ${red} ${f1red}`}></div>
```

GLAMOR

```
// Or use merge to output a single css selector
const f1 = css({ fontSize: '1rem' });
const red = css({ background: 'red' });
const f1Red = css(f1, red, { fontSize: '2rem' });

<div class={rule}></div>
```

DATA MODEL

.md, .json, APIs and GraphQL

CONTENT DATABASE

```
const data = [
  Markdown,
  Yaml,
  JSON,
  Latex,
  GitHub API,
  Contentful Posts,
]
```

JSON MODEL

```
const Markdown = {
  attributes: {
    title: 'My First Post',
    slug: 'my-first-post',
    url: '/my-first-post',
  },
  file: {
    path: './data/posts/my-first-post.md',
  },
  body: '<h1>thing</h1>...',
  rawBody: '# thing\n...',
}
```

QUERYING

```
query MyStuff {
  markdown(first: 5) {
    attributes: {
      title
      slug
    },
    body
  }
}
```

QUERY RESULT

```
{
  attributes: {
    title: 'My First Post',
    slug: 'my-first-post',
  },
  body: '<h1>thing</h1>...',
}
```

CONTENT TYPES

```
type Markdown {
  attributes: MarkdownAttributes
  body: HTML
  rawBody: RawMarkdown
}
```

AUTO-GENERATED FILTERING APIS

```
query {
  allPosts(
    filter: {
      published: false
    }
  ) {
    id
    title
    published
  }
}
```

UNIVERSAL APPS

SSR vs Static

UNIVERSAL CODE

- Render on Server and Client
- Unified Routing
- Code Sharing
- Unified Data Layer

UNIVERSAL RENDERING

```
const body = ReactDOM.renderToString(<App/>);
const html = ReactDOM.renderToStaticMarkup(
  <Html
    body={body}
    assets={locals.assets}
    bundleAssets={locals.assetsPluginHash}
    data={initialState}
  />
);
```

```
ReactDOM.render(<App />, document.getElementById("content"));
```

UI FRAMEWORKS SUPPORT UNIVERSAL RENDERING

- React
- Preact
- Inferno
- Angular 2
- Ember
- Vue.js
- ...

UNIFIED ROUTING

```
const html = ReactDOMServer.renderToString(
  <StaticRouter location={req.url} context={context}>
    <App />
  </StaticRouter>
);

<BrowserRouter
  basename={optionalString}
  forceRefresh={optionalBool}
  getUserConfirmation={optionalFunc}
  keyLength={optionalNumber}
>
  <App/>
</BrowserRouter>
```

CODE SHARING

ROUTING

URL Generation and Rendering

URL GENERATION

- dirs&paths
- Content
- Post Processing

UI AS FUNCTION OF URL

LOW EARTH ORBIT

A Library for Building Static Site Generators

- Acquire Data
- Process Data
- Expose Data as GraphQL API
- Extract URLs from Data
- Webpack Bundles
- Static Bundle (URL)

- Webpack
- Plugins
- Data Processing
- GraphQL
- Pluggable Scaffolding
- Content Types and Sources
- Decoupling Editing, Rendering, and Storage

> **... Jekyll is not meant to be included in a build process. Jekyll was created to *be* the build process**
>
> *- Smashing Magazine*

LEO PLUGINS

- index
- process
- schema
- loader

GRAPHQL

PLUGGABLE SCAFFOLDING

ETC...

FIN

- @chrisbiscardi
- superawesomelabs/leo
- [webpack] 3:45; Track 2
- [universal apps] 4:35; Track 2