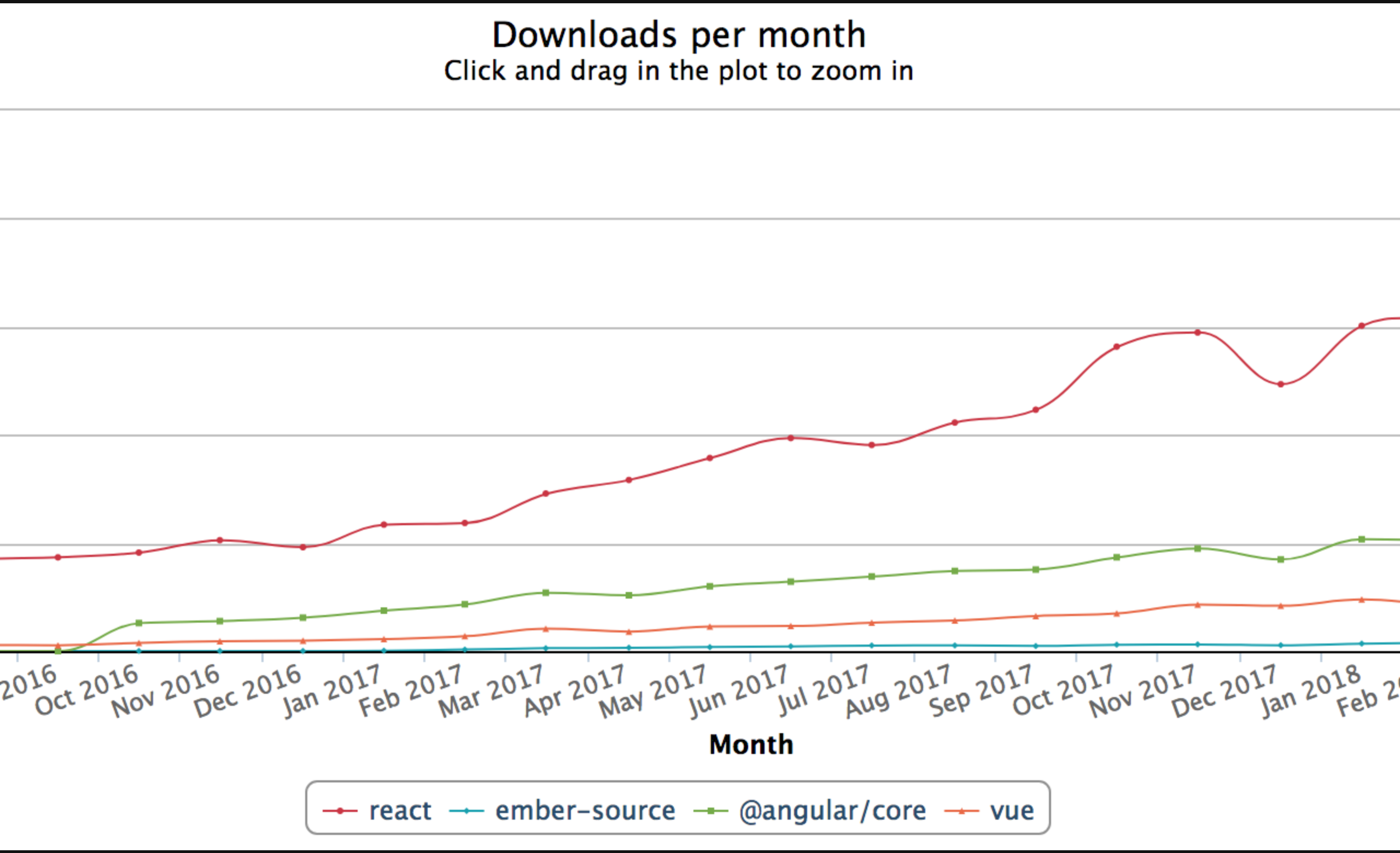
AMBITIOUS REACT APPLICATIONS

WHY REACT?


Downloads per month
Click and drag in the plot to zoom in
Oct 2016
Nov 2016
Dec 2016
Jan 2017
Feb 2017
Mar 2017
Apr 2017
May 2017
Jun 2017
Jul 2017
Aug 2017
Sep 2017
Oct 2017
Nov 2017
Dec 2017
Jan 2018
Month
react
ember-source
@angular/core
vue

Framework Spectrum

"View layer"
Focussed
Flexible
Lightweight
Innovative
Fatigue

"All the things"
Opinionated
Productive
Bloated
Stable
Stagnation

PHILOSOPHY

EMBER

Convention > Configuration

REACT

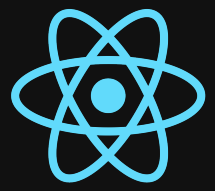
```
ui = f(s)
```

TOOLING

EMBER CLI

```
ember new my-new-app
cd my-new-app
ember test
ember serve
ember generate route items
```

REACT



[create-react-app](#)

[Next.js](#)

[Gatsby](#)

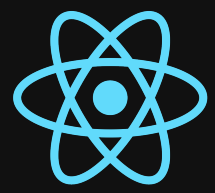
🚀 Like Ember CLI, each:

- installs dependencies & tooling
- insulates devs from low-level tooling
- comes with (a few) opinions (folders, etc)
- has commands to run & build app

📖 Emphasize docs/examples for the rest:

- styling components (CSS in JS, Sass, etc)
- installing & using libraries
- writing & running tests
- deployment

CREATE-REACT-APP



Key Feature: official support/docs

Best Uses: learning react, simple apps

Owner: Facebook

NEXT.JS

Key Feature: SSR, route-based data fetching & code-splitting

Best Uses: data-centric apps

Owner: Zeit (Now)

Key Feature: static site generation, ecosystem

Best Uses: content-centric apps like blogs

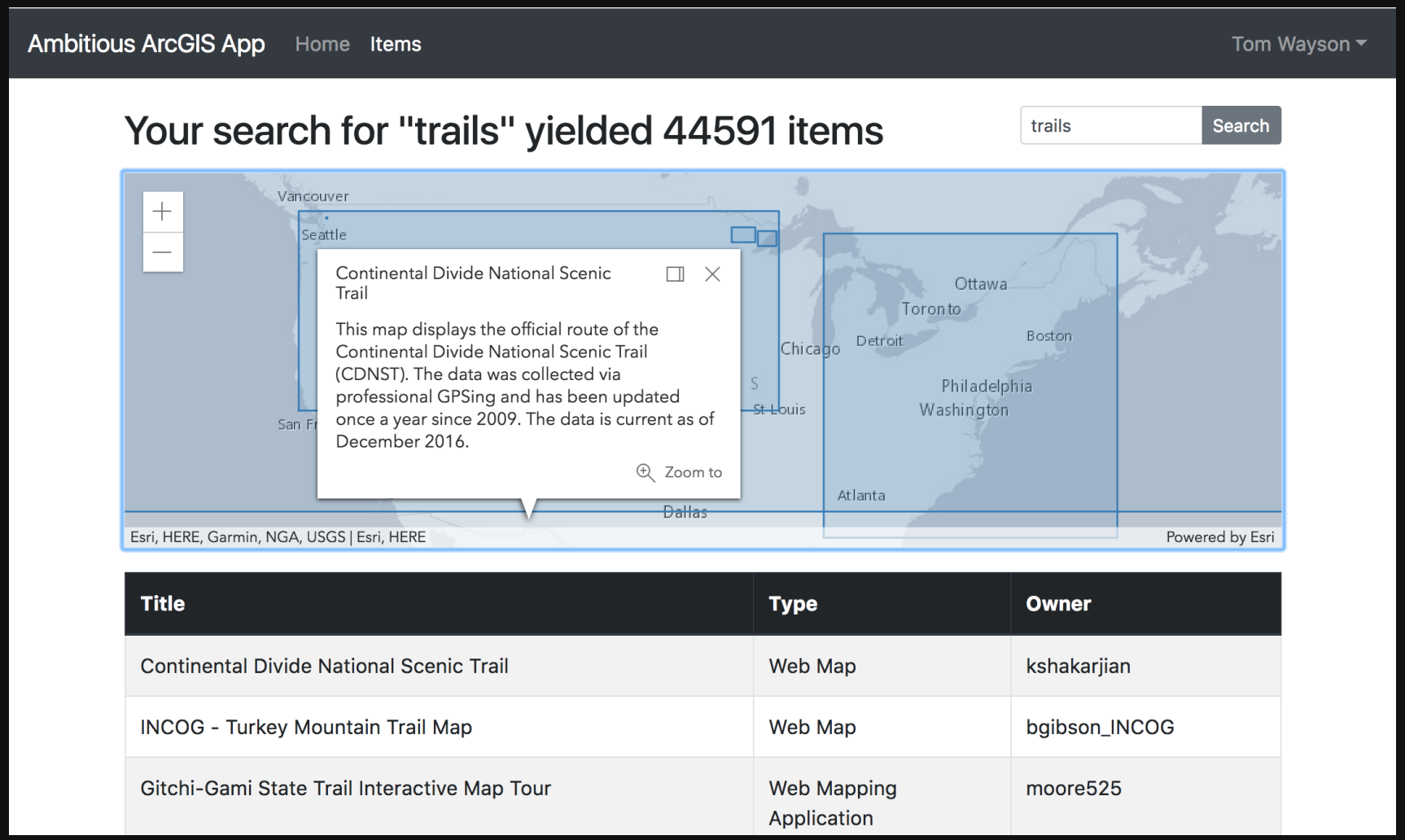
Owner: Gatsby.com

WHICH DID I CHOOSE?

create-react-app

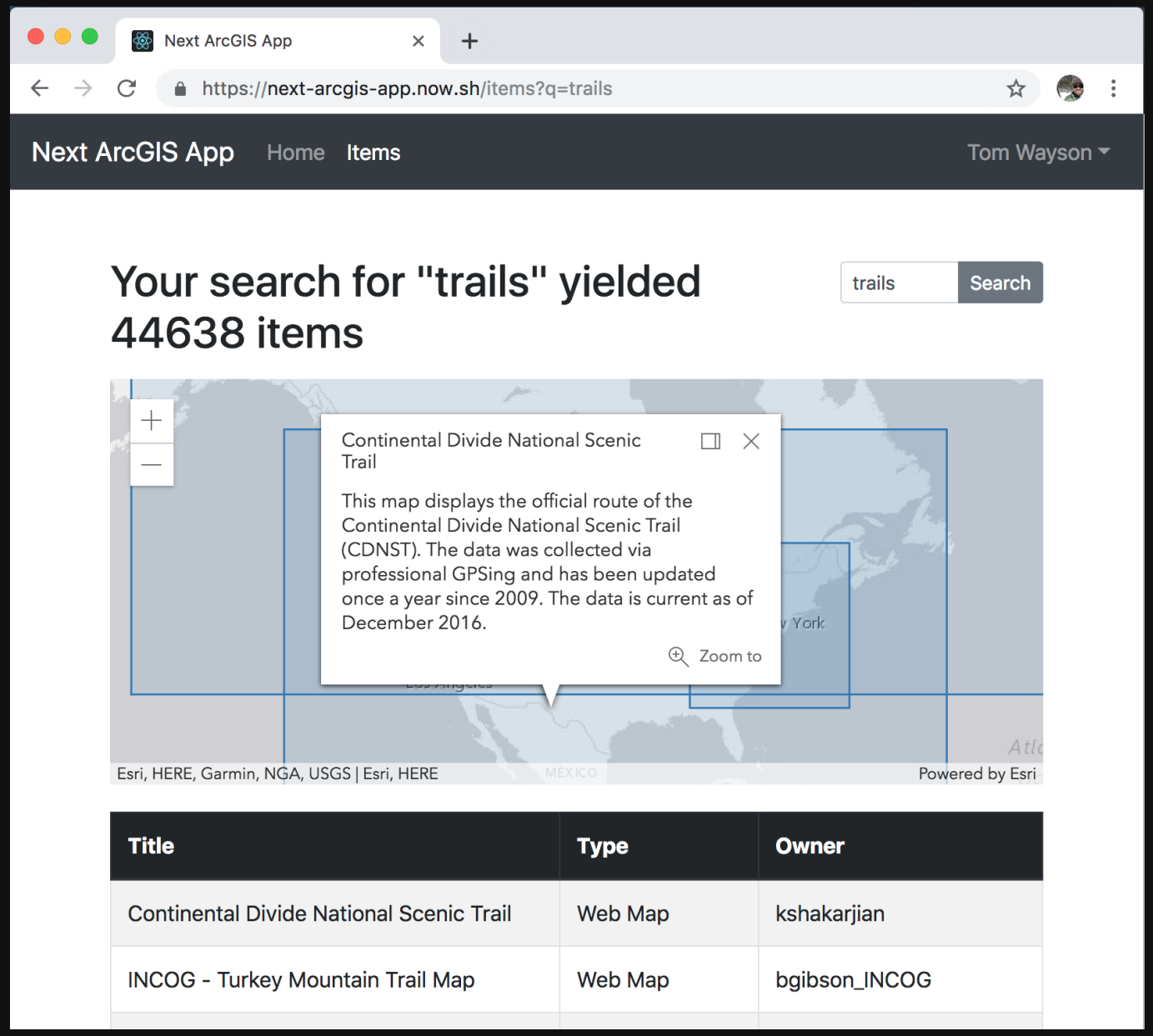
... and Next.js

CREATE ARCGIS APP



[create-arcgis-app]

NEXT ARCGIS APP



[next-arcgis-app]

ARCHITECTURE

EMBER

Routes drive the application

Plan application as a tree of nested **routes**

```
/application
  route
  template
  /index
    controller
    template
  /items
    route
    controller
    template
```

For cross-cutting concerns use **services** (i.e. `session`)

REACT

(almost) Everything is a component

Ember	**React**
Router	Component
Route	Component
Controller	Component (JS)
Template	Component (JSX)
Component	Component
Helper	Component (SFC)
Service	Context or Redux
Util	**Util**
Ember Data	GraphQL
Initializer	index.js

Plan application as a tree of **components**

```
index.js
<Router>
  <App>
    <AppNav>
    <Route match="/">
      <Home>
    <Route match="/items">
      <Items>
        <ItemsLayout>
```

For cross-cutting concerns use Context

LIFECYCLE METHODS

Ember	React
`init()`	`constructor()`
`didReceiveAttrs()`	`getDerivedStateFromProps()`
`willRender()`	`render()`
`didInsertElemenet()`	`componentDidMount()`
`didUpdateAttrs()`	`componentDidUpdate()`
`willDestroyElement()`	`componentWillUnmount()`

LIBRARIES

REACTSTRAP

~~ember install ember-bootstrap~~

```
npm install --save bootstrap
npm install --save reactstrap
```

```
import 'bootstrap/dist/css/bootstrap.css';
```

ARCGIS UTILS FOR *ANY* FRAMEWORK

ArcGIS REST JS Guides API Reference ☰ Menu ArcGIS REST API JavaScript Client

arcgis-rest-js

~~ember install torii-provider-arcgis~~
~~ember install ember-arcgis-portal-services~~

```
npm install --save @esri/arcgis-rest-items
npm install --save @esri/arcgis-rest-auth
npm install --save @esri/arcgis-rest-request
```

AUTH

Sign In (utils/session.js)

```
import { UserSession } from '@esri/arcgis-rest-auth';
import env from '../config/environment';
import * as Cookies from 'js-cookie';

const SESSION_COOKIE_NAME = `${env.cookiePrefix}_session`;

/**
 * sign in using OAuth pop up
 */
export function signIn() {
  const { clientId, portal } = env;
  return UserSession.beginOAuth2({
    clientId,
    portal,
    popup: true,
    redirectUri: `${window.location.origin}/redirect.html`
  }).then(session => {
    // save session for next time the user loads the app
```

Sign In (public/redirect.html)

```
<!DOCTYPE html>
```

Sign In (utils/session.js)

```
const SESSION_COOKIE_NAME = `${env.cookiePrefix}_session`;

/**
 * sign in using OAuth pop up
 */
export function signIn() {
  const { clientId, portal } = env;
  return UserSession.beginOAuth2({
    clientId,
    portal,
    popup: true,
    redirectUri: `${window.location.origin}/redirect.html`
  }).then(session => {
    // save session for next time the user loads the app
    saveSession(session);
    return session;
  });
```

Sign In/Out (App.jsx)

```
class App extends React.Component {
  constructor(props) {
    super(props);
    // set application state
    // NOTE: when storing objects like session or user in state
    // React uses the Object.is() comparison algorithm to detect changes
    // and changes to child properties won't trigger a re-render
    // which is fine for this application, b/c we only ever set session/user
    this.state = {
      session: props.previousSession
    };
  }
  signIn = () => {
    signIn().then(session => {
      // make session available to the app
      this.setState({ session });
      // get user info
```

ITEMS

Items Route (Ember)

```
import Route from '@ember/routing/route';
```

Items Component (React)

```
import React from 'react';
import { Alert } from 'reactstrap';
import { searchItems } from '@esri/arcgis-rest-items';
import { parseSearch, didSearchParamsChange } from '../utils/search';
import ItemsLayout from '../components/ItemsLayout';

class Items extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      error: undefined,
      results: undefined,
      total: 0
    };
  }
  onParamsChange = (q, start) => {
    // update the route query params after the user either
    // submits the inline search form or links to a new page
```

MAPS

ESRI-LOADER

- ✅ create-react-app
- ✅ Next.js
- ✅ Gatsby

@ARCGIS/WEBPACK-PLUGIN

- ❌ create-react-app
- ❌ Next.js
- ❌ Gatsby

@ARCGIS/WEBPACK-PLUGIN

Roll your own

```
git init
npm init
npm i -S react react-dom
npm i -D webpack webpack-cli webpack-dev-server
npm i -D @arcgis/webpack-plugin
npm i -D @babel/core @babel/preset-env @babel/preset-react babel-loader babel-
plugin-transform-es2015-modules-amd
```

ESRI-LOADER

~~ember install ember-esri-loader~~

```
npm install --save esri-loader
```

MAP UTIL CODE

🙌 just copy + paste 🎉

Lifecycle Methods (Ember)

```
    const { symbol, popupTemplate } = config.APP.map.itemExtents;
    showItemsOnMap(this._view, this.items, symbol, popupTemplate);
  },

  // wait until after the component is added to the DOM before creating the map
  didInsertElement () {
    this._super(...arguments);
    // create a map at this element's DOM node
    loadMap(this.elementId, config.APP.map.options)
    .then(view => {
      // hold onto a reference to the map view
      this._view = view;
      this.showItems();
    });
  },

  // whenever items change, update the map
  didUpdateAttrs () {
```

Lifecycle Methods (React)

```
showItems() {
  const { symbol, popupTemplate } = config.itemExtents;
  showItemsOnMap(this._view, this.props.items, symbol, popupTemplate);
}

// react lifecycle methods
// wait until after the component is added to the DOM before creating the map
componentDidMount() {
  // create a map at this element's DOM node
  loadMap(this.mapNode.current, config.options).then(view => {
    // hold onto a reference to the map view
    this._view = view;
    // show the initial items on the map
    this.showItems();
  });
}
```

DOM Node (Ember)

```
export default Component.extend({
  classNames: ['extents-map'],

  // show items on the map w/ the symbol and popupTemplate from the config
  showItems () {
    const { symbol, popupTemplate } = config.APP.map.itemExtents;
    showItemsOnMap(this._view, this.items, symbol, popupTemplate);
  },

  // wait until after the component is added to the DOM before creating the map
  didInsertElement () {
    this._super(...arguments);
    // create a map at this element's DOM node
    loadMap(this.elementId, config.APP.map.options)
    .then(view => {
      // hold onto a reference to the map view
```

DOM Node (React)

```
// react lifecycle methods
// wait until after the component is added to the DOM before creating the map
componentDidMount() {
  // create a map at this element's DOM node
  loadMap(this.mapNode.current, config.options).then(view => {
    // hold onto a reference to the map view
    this._view = view;
    // show the initial items on the map
    this.showItems();
  });
}
componentDidUpdate(prevProps) {
  if (prevProps.items !== this.props.items && this._view) {
    this.showItems();
  }
```

TESTS

Feature	Ember	React
Helpers	ember-test-helpers	react-testing-library
Mocks	Sinon	Jest
Assertions	QUnit	Jest
Framework	QUnit	Jest
Runner	Testem	Jest

Jest - react-testing-library

DEPLOYMENT

~~`ember install ember-cli-surge`~~

Just follow [create-react-app deployment guide]

```
  "predeploy": "yarn run build && mv build/index.html build/200.html",
  "deploy": "surge ./build create-arcgis-app.surge.sh",
```

CONCLUSION

ember

QUESTIONS?