



Building Great Search Experiences with Search UI

—

Jason Stoltzfus, Software Engineer
Diane Tetrault, Director Product Marketing, Elastic



Jason Stoltzfus

Software Engineer



Diane Tetrault

Director Product Marketing
Elastic

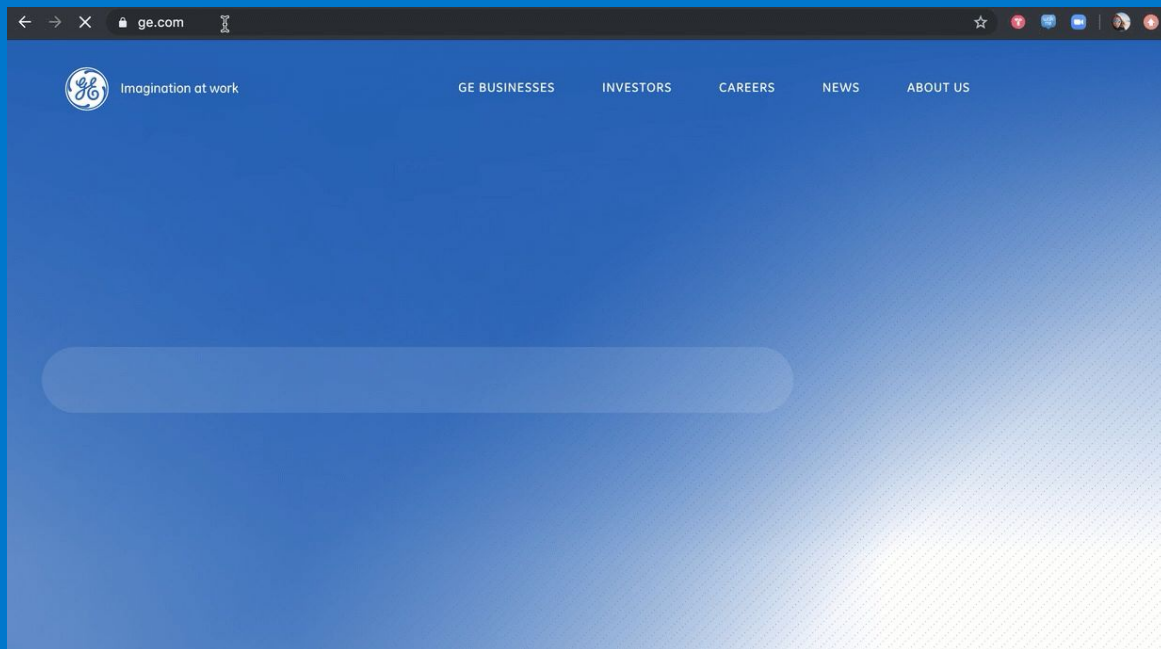
Housekeeping & Logistics

- Chat with us in the zoom chat
- **Recording** will be available after the webinar
- If you're only seeing part of screen, click "View Options" above the slides and click "Fit to Window"

Search is no longer an afterthought

Search has become the primary way for people to interact with your organization

- Online shopping
- Customer support
- Location finder
- Product information
- Partner portal
- Job search
- Internal workflow



What does good search look like?

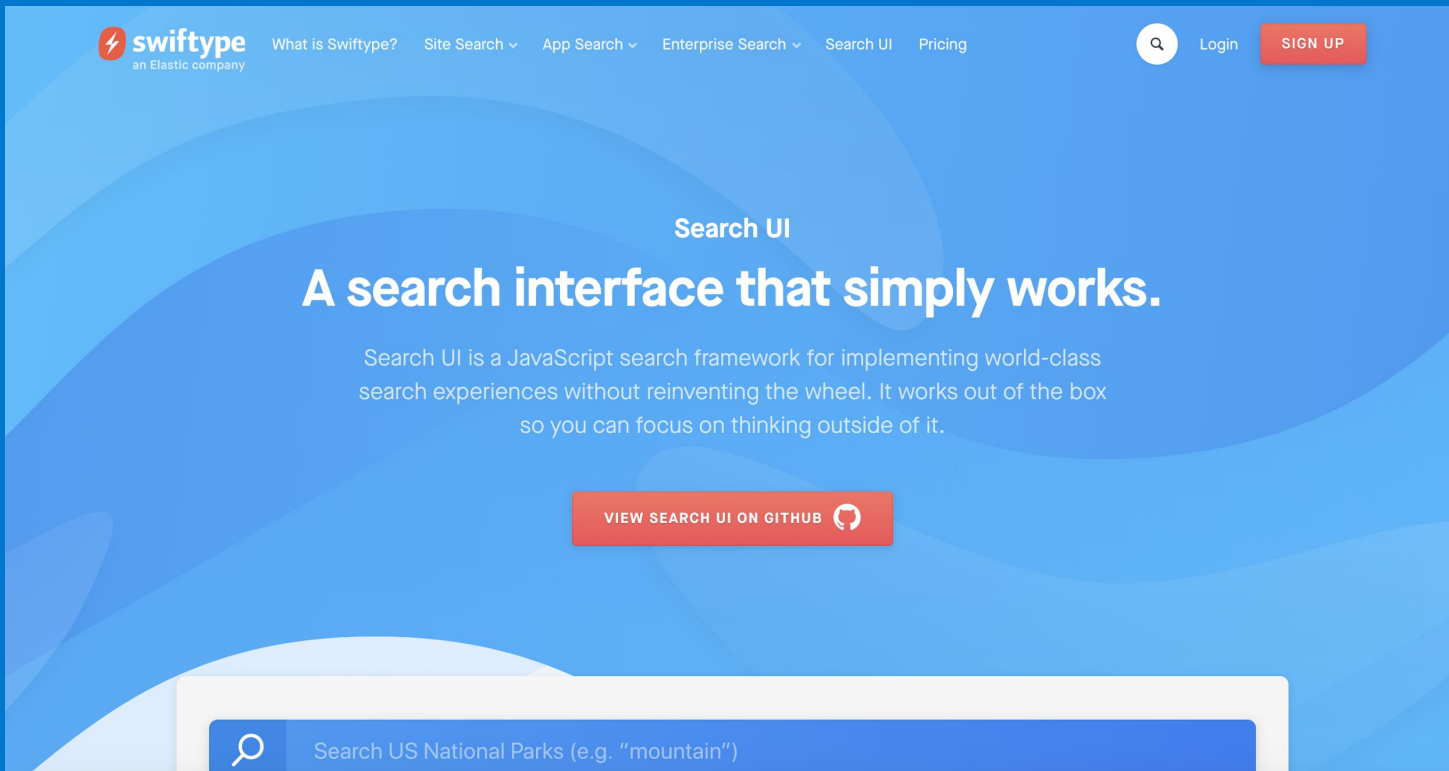
How to deliver an excellent user experience

Users expect search that's **easy to use**, that returns the **most relevant results**, and makes it **simple to filter, preview and act on results**.

But that can be **surprisingly hard to build**



Make it simpler with Search UI



Site Search

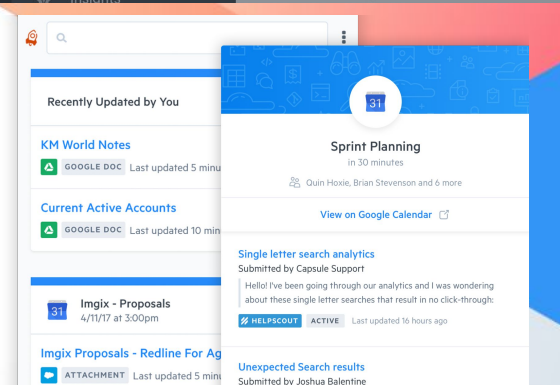
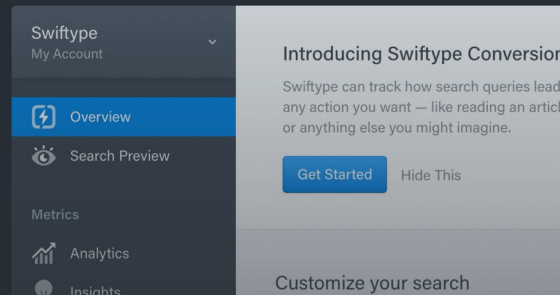
Create and manage a tailored search experience for your website with world-class relevance, intuitive customization, and rich analytics.

Enterprise Search

Instantly search across all the tools you use at work, including Dropbox, Salesforce, G Suite, Office 365, Zendesk, and more.

App Search

A powerful set of APIs and developer tools designed for developers building rich, user-facing search applications.



Search Solutions are a vertical slice through the Elastic Stack

Simple to get started with turn-key features

Channeling the Elastic Stack's power and flexibility in a laser-focused way

Focuses on both out-of-the-box functionality and customizability

Adding value by abstracting away optimizations and use case-specific work



How do you deliver great search?

It's not just what goes on behind the scenes

Relevance, analytics, ingestion and more

The screenshot shows the 'Relevance Tuning' interface for an 'ecommerce' index. The left sidebar contains navigation links: Overview, Analytics, Query Tester, Reference UI, MANAGE (Documents, Schema, API Logs), SEARCH SETTINGS (Synonyms, Curations, Relevance Tuning, Result Settings), and ACCESS. The main area is titled 'Relevance Tuning' with the subtitle 'Set field weights and boosts.' and buttons for 'Restore Defaults' and 'Save'. Below this is the 'Manage Fields' section with a search bar 'Filter 15 fields...'. It lists fields like 'category' (TEXT, 1), 'colors' (TEXT, 1), and 'created_at' (DATE, 0). A 'Preview' section shows a search for 'watch' with two results. The first result has a score of 15.36 and lists fields: name (J.Crew Home Black Wate...), colors (black watch home b geen), department (home), details (Viscose/nylon/mercerize...), and category (pillows). The second result has a score of 12.39 and a description: 'Your go-to short for hang...'. Buttons for 'Restore Defaults' and 'Save' are at the bottom right.

User interface, faceting, results and more

The screenshot shows a search results page for 'US National Parks'. The search bar at the top contains the text 'Search US National Parks (e.g. "mountain")'. Below the search bar, there are two faceted sections: 'SORT BY' and 'STATES'. The 'SORT BY' section lists: Visitors: Low to High, Visitors: High to Low, Acres: Low to High, and Acres: High to Low. The 'STATES' section lists: California (9), Alaska (8), Utah (5), Colorado (4), and Arizona (3). The main results area displays six cards, each with a landscape image and a park name: Zion, Yosemite, Yellowstone, Wrangell-St. Elias, Wind Cave, and Voyageurs. At the bottom right, there is a pagination control showing '1 2 3 4 ... 10'.

Which is where Search UI comes in

Tools for building search experiences

- **Site Search JavaScript snippet**
 - Just drop the snippet on your page and we inject a search experience
- **REST APIs**
 - The engine that powers custom search experiences
- **Language Clients**
 - Libraries that make using REST API easier in various programming languages
- **App Search Reference UI**
 - An out of the box search experience that you can generate and download
- **Search UI**
 - A JavaScript library for building search experiences

sample-engine
ENGLISH

Overview
Analytics
Query Tester
Reference UI

MANAGE
Documents
Schema
API Logs

SEARCH SETTINGS
Synonyms
Curations
Relevance Tuning
Result Settings

ACCESS
Credentials

Reset Demo

Query Tester

Create a New Reference UI

Preview search or kickstart your next search experience.

The Reference UI is an open source, minimally styled search boilerplate written in React. Fill in the fields below to generate an interactive preview, then download the .zip and build upon it however you'd like.

[Explore the source code](#) | [Read the Reference UI guide](#)

Title field (Optional)

Used as the top-level visual identifier for every rendered result

Filter fields (Optional)

Click to select

Faceted values rendered as filters and available as query refinement

Sort fields (Optional)

Click to select

Used to display result sorting options, ascending and descending

URL field (Optional)

Used as a result's link target, if applicable

Generate a Preview

athletic shoe

SORT BY

Newest


FILTER BY


☒ Men


☒ Athletic

☒ Women

☒ Athletic

 Nike Revolution 4
<https://shoes.com/nike-revolution-4>

 Asics Gel Venture
<https://shoes.com/asics-gel-venture>

 Adidas Speed Trainer
<https://shoes.com/adidas-speed-trainer>

<

>

What is Search UI?

A React JavaScript library that allows you to quickly implement search experiences without re-inventing the wheel.

What exactly is React?

<https://reactjs.org/> - A very popular JavaScript library created by Facebook

React is built on the concept of "components". components are like lego blocks for building web applications. Very similar to HTML.

HTML

```
<form>
  <input type="text" />
  <input type="submit" />
</form>
```

React

```
<form>
  <SearchBox />
  <SearchResults />
</form>
```

Search UI components are derived from standard search patterns

<SearchBox />

<Sorting />

<Facets />

<PagingInfo />

<ResultsPerPage />

<Results />

<Paging />

The interface displays a search results page with the following components:

- SearchBar:** A text input field with a "Search" button.
- Sorting:** A dropdown menu currently set to "Relevance".
- Facets:**
 - STATES:** A list of states with checkboxes and counts: California (3), Alaska (2), Arizona (2), Florida (2), South Dakota (2). Arizona is selected.
 - VISITORS:** A range selector showing "500001 - 1000000" with a "Remove" link.
 - DATE ESTABLISHED:** Radio buttons for "More than 100 years ago" (0), "50 - 100 years ago" (1), and "Within the last 50 years" (1).
 - DISTANCE:** Radio buttons for "Nearby" (0), "A longer drive" (0), and "Perhaps fly?" (2).
 - ACRES:** A dropdown menu with "Select..." and a chevron icon.
- PagingInfo:** A section showing "Showing 1 - 2 out of 2".
- ResultsPerPage:** A dropdown menu currently set to "20".
- Results:** A list of search results, each with a title and a JSON object of metadata.
 - Petrified Forest:**
 - nps_link: https://www.nps.gov/pefo/index.htm
 - title: Petrified Forest
 - date_established: 1962-12-09T06:00:00+00:00
 - world_heritage_site: false
 - states: Arizona
 - description: This portion of the Chinle Formation has a large concentration of 225-million-year-old petrified
 - visitors: 643274
 - id: park_petrified-forest
 - location: 35.07,-109.78
 - square_km: 896
 - acres: 221415.77
 - Saguaro:**
 - nps_link: https://www.nps.gov/sagu/index.htm
 - title: Saguaro
 - date_established: 1994-10-14T05:00:00+00:00
 - world_heritage_site: false
 - states: Arizona
 - description: Split into the separate Rincon Mountain and Tucson Mountain districts, this park is evidence
 - visitors: 820426
 - id: park_saguaro
 - location: 32.25,-110.5
 - square_km: 371.2
 - acres: 91715.72
- Paging:** A section at the bottom showing "< 1 >".

Easy to get started, deeply customizable

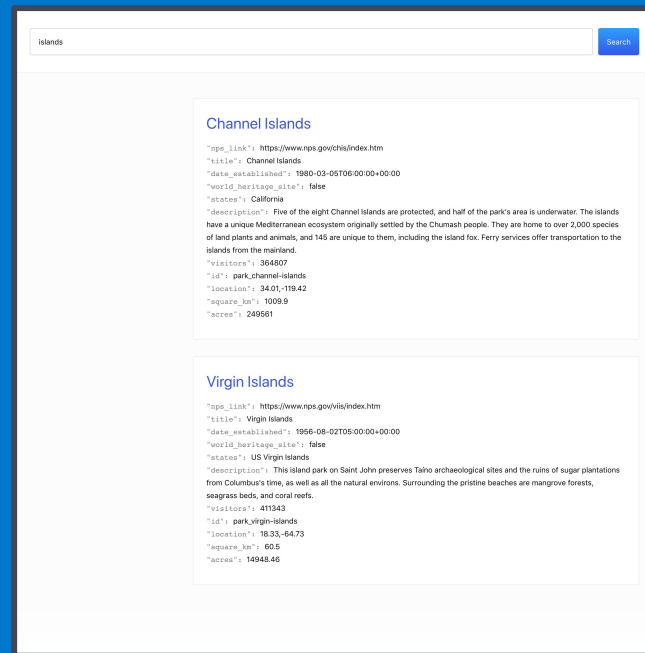
<https://codesandbox.io/s/getting-started-y9tnd>

```
import React from "react";
import AppSearchAPIConnector from "@elastic/search-ui-app-search-connector";
import { SearchProvider, Results, SearchBox } from "@elastic/react-search-ui";
import { Layout } from "@elastic/react-search-ui-views";

import "@elastic/react-search-ui-views/lib/styles/styles.css";

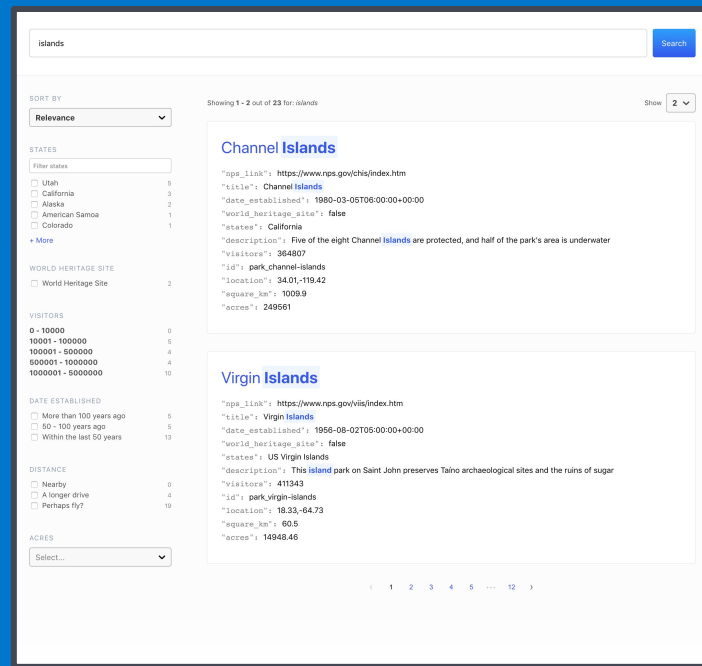
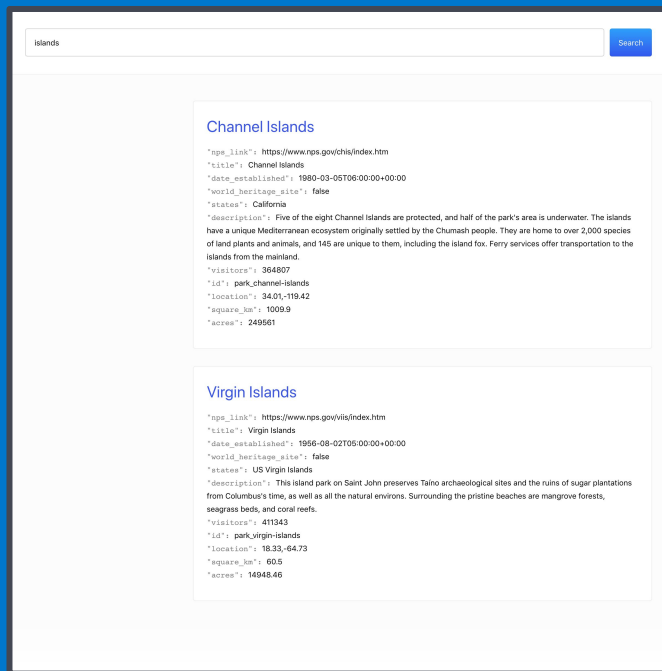
const connector = new AppSearchAPIConnector({
  searchKey: "search-371auk61r2bwqtdzocdgutmg",
  engineName: "search-ui-examples",
  hostIdentifier: "host-2376rb"
});

export default function App() {
  return (
    <SearchProvider
      config={{
        apiConnector: connector
      }}
    >
      <div className="App">
        <Layout
          header={<SearchBox />}
          bodyContent={<Results titleField="title" urlField="nps_link" />}
        />
      </div>
    </SearchProvider>
  );
}
```



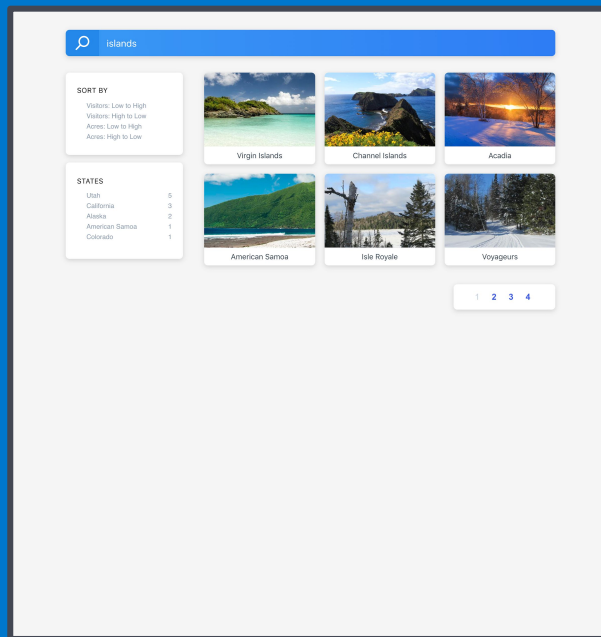
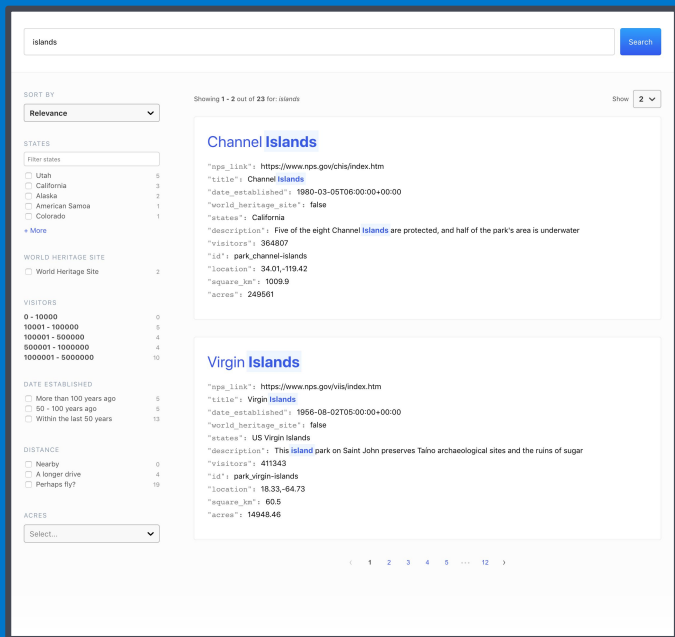
Many great options right out of the box

<https://codesandbox.io/embed/search-ui-national-parks-example-kdylms>



Everything is customizable

<https://swifttype.com/search-ui>

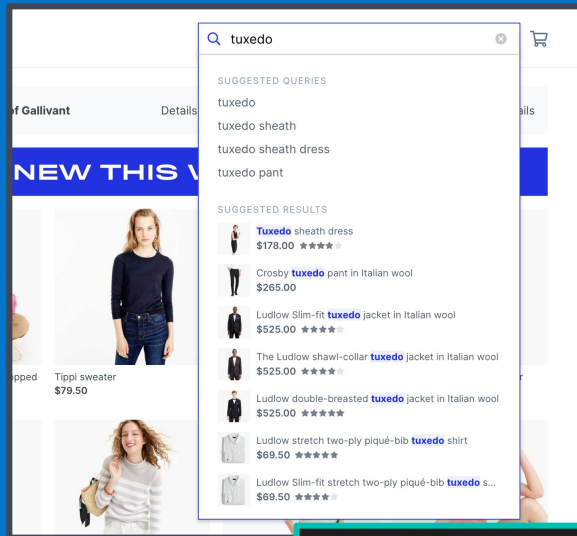


Best practices and advice for building search

- Don't underestimate search
- Follow established patterns
- Focus on responsiveness
- Avoid "No Results" screens

Autocomplete

- Get a user to what they're looking for in the least number of steps possible
- Results vs Query Suggestions
- Autocomplete should:
 - Make fast queries
 - Support throttling
 - Have keyboard navigation
 - Support Caching (back button)



```
<SearchBox
  autocompleteMinimumCharacters={3}
  autocompleteResults={{
    sectionTitle: "Results",
    titleField: "title",
    urlField: "nps_link"
  }}
  autocompleteSuggestions={true}
  debounceLength={100}
/>
```

Search As You Type

```
<SearchBox searchAsYouType={true} />
```

bryce

I

Search

SORT BY

Relevance

STATES

Filter states

Utah

1

VISITORS

0 - 10000

0

10001 - 500000

0

100001 - 500000

0

500001 - 1000000

0

1000001 - 5000000

1

DATE ESTABLISHED

More than 100 years ago

0

50 - 100 years ago

1

Within the last 50 years

0

DISTANCE

Nearby

0

A longer drive

0

Perhaps fly?

1

ACRES

Select...

Showing 1 - 1 out of 1 for: bryce

Show20

Bryce Canyon

"nps_link": <https://www.nps.gov/brca/index.htm>

"title": Bryce Canyon

"date_established": 1928-02-25T06:00:00+00:00

"world_heritage_site": false

"states": Utah

"description": Bryce Canyon is a geological amphitheater on the Paunsaugut Plateau with hundreds of tall

"visitors": 2365110

"id": park_bryce-canyon

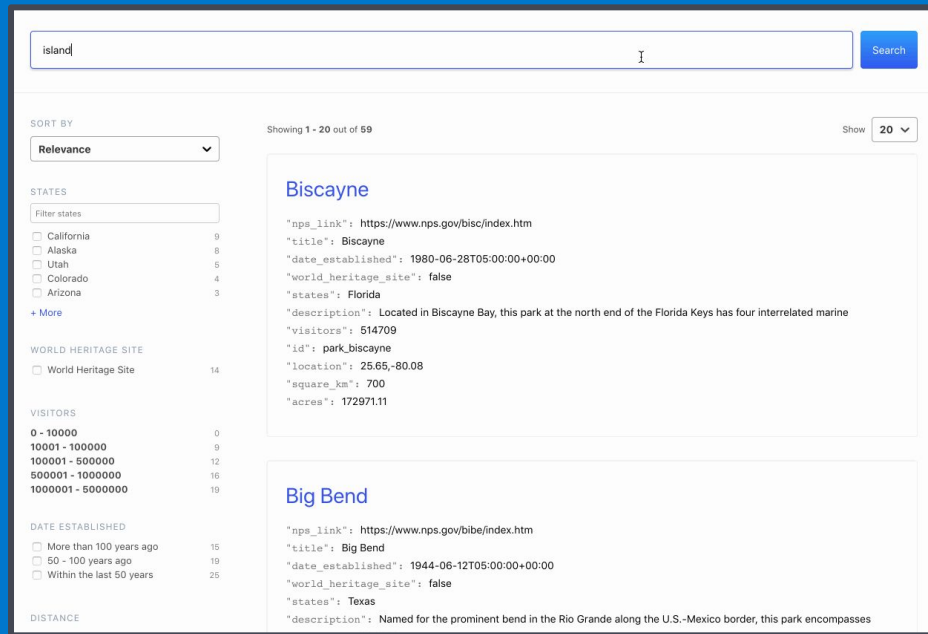
"location": 37.57,-112.18

"square_km": 145

"acres": 35835.08

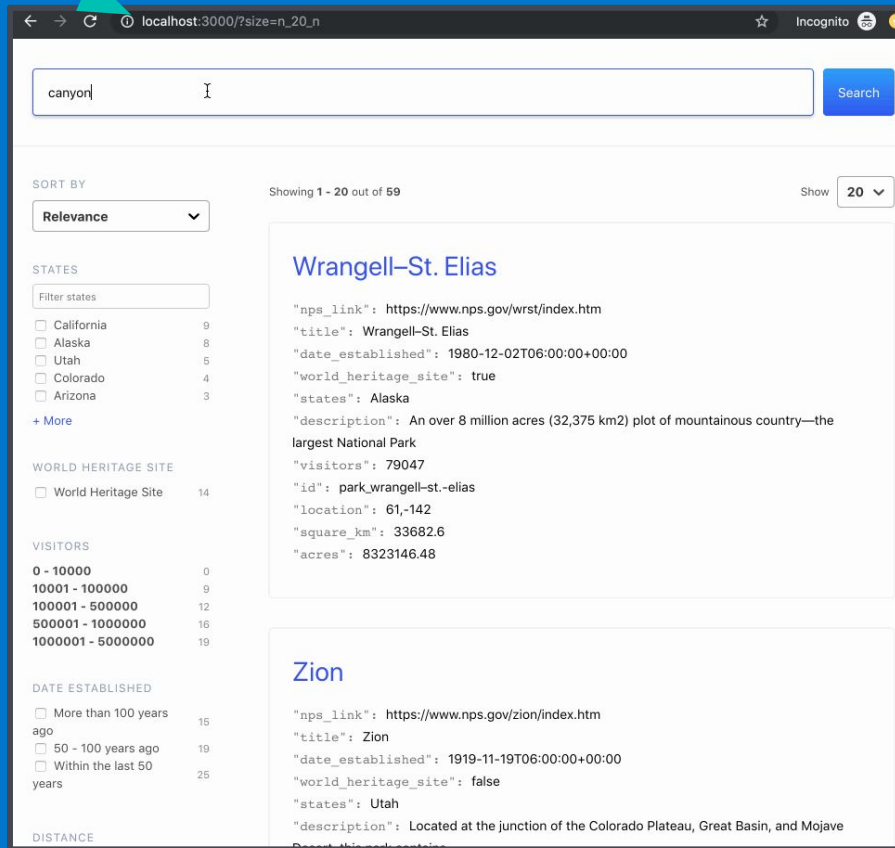
Incremental Refinement

- Start with large results set
- Apply refinements immediately
- Facets avoid "No Results"
- Don't clear that search box!



Shareable URLs

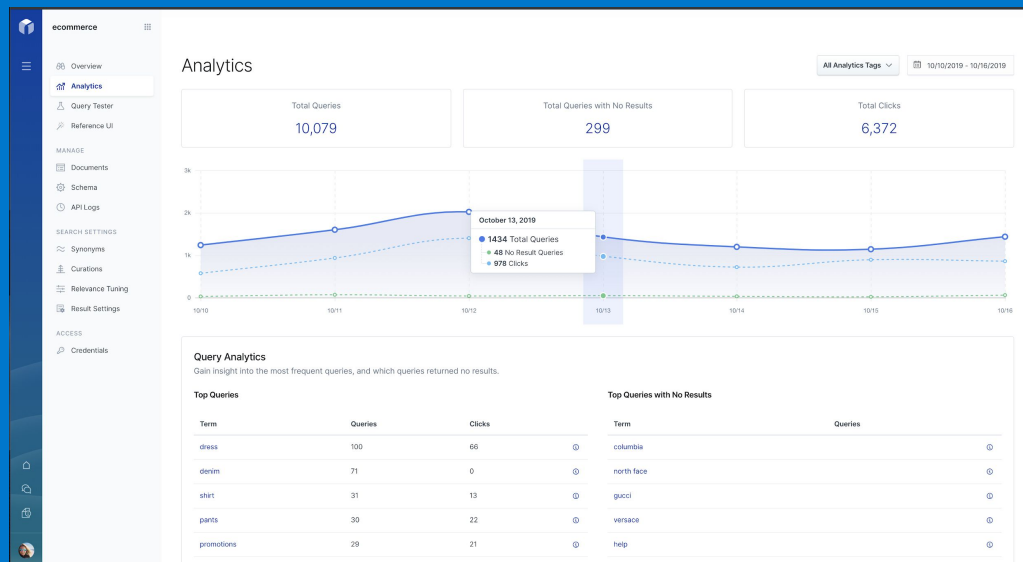
- URL state is tracked by Search UI
- Browser back/forward navigation
- Shareable urls



Track Result Click Throughs

For a given search term, what result is clicked the most?

Boost the relevance for that result.



Advanced Features

- **Low - level API for custom components**
- **Don't have to use React**
- **Swappable back-ends**
- **Mobile friendly**

Some examples of Search UI at work

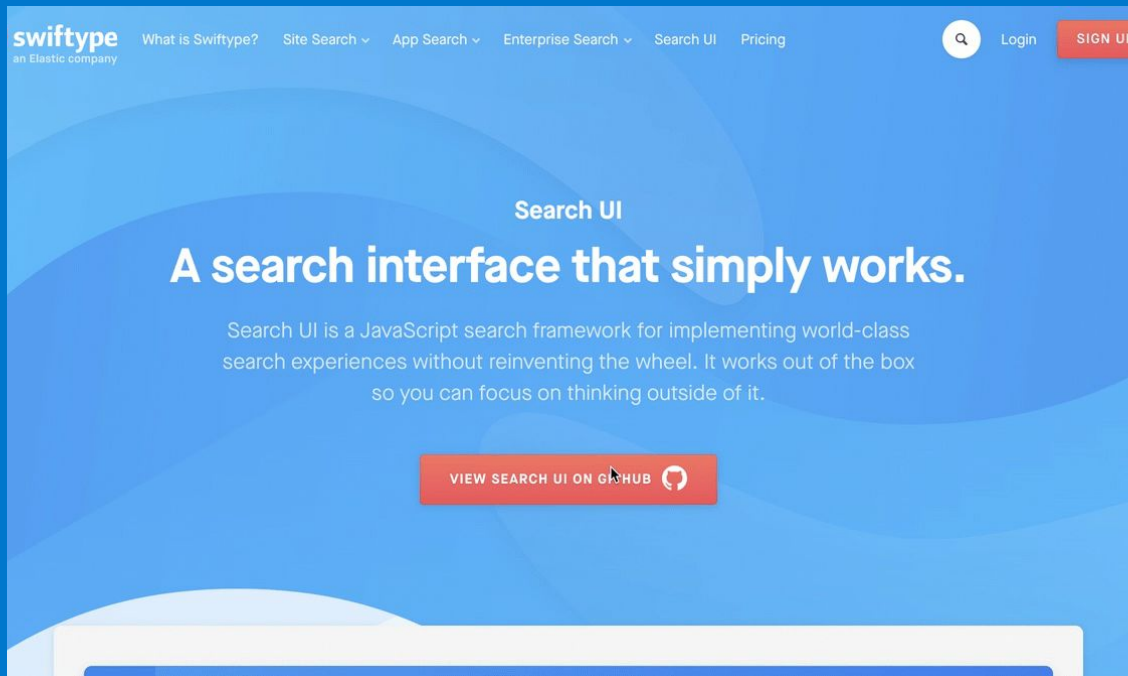
Get Started with Search UI

See how how easy it can be to create great search experiences

Leverage Search UI with Elastic App Search

1. Visit Search UI on our website
2. OR head directly to Github:
<https://github.com/elastic/search-ui>
3. We've got a Getting Started guide
4. Get started!

Not an Elastic App Search user? Start a free trial or download the self-managed version

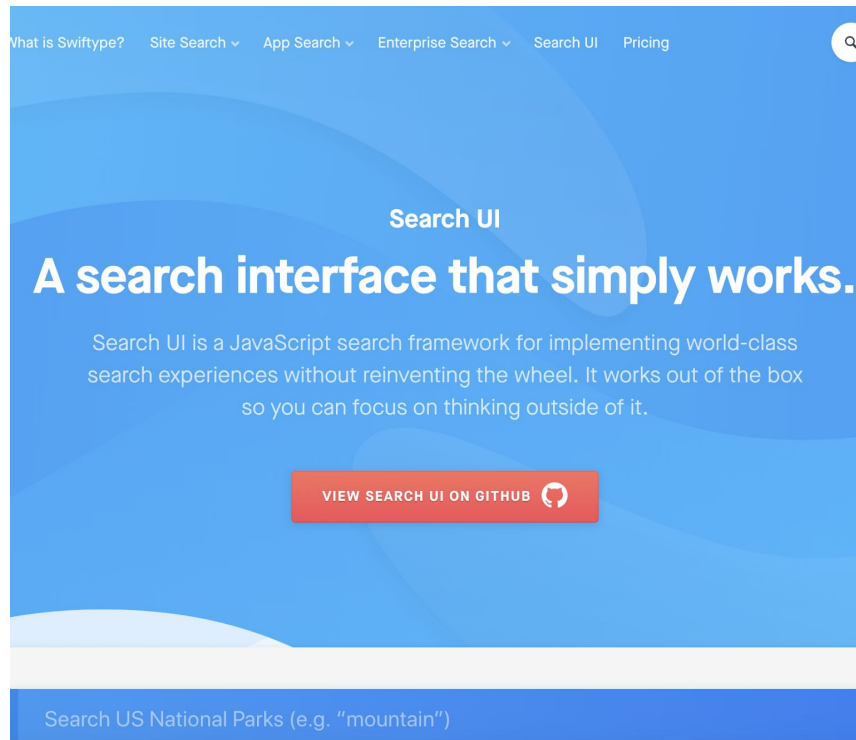


Additional Resources

Lots of help to get you start

- swiftype.com/search-ui
- See it directly on Github
- Our GA blog post:
<https://www.elastic.co/blog/search-ui-1-0-0-released>
- Our latest release blog post:
<https://www.elastic.co/blog/search-ui-1-2-0-released>
- A vast community of peers

**Plus we'll send you a follow-up email
with links to all of these and more**



Q&A