Q&A in Chat during Live Webinar: Building Great Search Experiences with Search UI

Is the product being shown an add-on we need to purchase?

Search UI is open source and available with Elastic App Search. Elastic App Search itself is available as a self-managed version with a basic Elasticsearch license. There's also a SaaS version (for which there's a monthly cost, as we host/scale for you).

Does Site Search UI work with the on-premises subscription? Are all components installed and hosted internally? / Can this search engine be utilized internally without access to external resources? Basically, setup is an intranet web page behind a firewall, and no external resource can be utilized?

Search UI can work for an on-prem/self-managed implementation of App Search.

Is Search UI open-source? If not, what is the licensing approach?

Search UI is open source; it's under Apache 2.0

Search UI front end - can it talk to elasticsearch to get results? In other words, can the source be other than elasticsearch?

It can. Refer to these links:
https://github.com/elastic/search-ui#does-search-ui-only-work-with-app-search
https://github.com/elastic/search-ui#how-do-i-use-this-with-elasticsearch

Does Search UI work for on-premise implementation for an intranet search?

Search UI can work for an on-prem/self-managed implementation of App Search.

How do you drop an endpoint w/ query behind a component, like a search bar for example?

We would like to help you with this. Could you provide more details about your use case in our gitter? https://gitter.im/elastic-search-ui/community

Is there documentation for Reference UI?

Documentation for Reference UI can be found here:
https://swiftype.com/documentation/app-search/guides/reference-ui

Is there a documentation or example of how to bind back end to a custom search engine?

We have 2 out-of-the-box connectors for Search UI: site-search-connector and app-search connector: https://github.com/elastic/search-ui/tree/master/packages. You could use these examples to bind Search UI to your search engine. If you have any questions - feel free to ask us at our gitter: https://gitter.im/elastic-search-ui/community

How does it suggest functions precisely? Does it use user previous searches?

(Jason answered in the webinar); there's also more documentation on how Query Suggestions work here: https://swiftype.com/documentation/app-search/guides/query-suggestions

Is there a community for app search, to look for any questions and answers?

https://discuss.elastic.co/c/app-search

Is there a demo environment for Reference UI?

There's a preview environment for Reference UI built into App Search. You can try building one there. If you want a demo environment for Search UI, which is the underlying library, you can find that here: https://codesandbox.io/s/national-parks-example-kdyms?fontsize=14

Do you have to use React for on-prem or can you roll your own front-end with Vanilla JS?

We have React components available, but you can use Vanilla JS too. Refer to this part of docs: https://github.com/elastic/search-ui#is-search-ui-only-for-react

What do you think about user mistakes, like a wrong OrthoGraph. Does the search need to be exact or should we try to approximate the search like using levenshtein distance?

Being kind to your user's faults helps the experience :) App Search does bake in typo tolerance so you don't need to build it out.

Does enabling auto complete have a cost on index size?

For App Search, another index is already built for autocomplete (with optimizations for speed), so there isn't any "enabling" of autocomplete.

What about the source of data? It must be retrieved only from an Elasticsearch cluster? Or it is open for several types of storage?

If you're using App Search, you do need to ingest it through an App Search endpoint as we make a lot of optimizations under the hood on your behalf.

One area I would love to know is: Elastic use in contextualization of user experience, wrapping search keywords by context inputs (additional filters) and result scoring to implement context weight.

App Search does allow you to tune relevance with each search call; so you could provide / interpret the additional context inputs and just adjust the API call: https://swiftype.com/documentation/app-search/guides/relevance-tuning

Can you explain more on the connectors? I see Search UI supports Site Search engine...does it support Site Search Crawler Engine? / Do we have examples of other back end other than App Search like Site Search Crawler Engine with connects with Search UI ?

Search UI supports Site Search which is fed by the Crawler Engine; the Site Search crawler on its own still needs a search backend.

Our use case involves keyword searching in html documents such as pipes MUST iron. Does the standard Search UI support this?

Search UI sends the responses to the backend; it's up to the backend power the search to interpret: App Search allows for some Lucene Query Syntax which should allow for MUST (double quoted strings): https://swiftype.com/documentation/app-search/guides/relevance-tuning

Second Interest: Geo filters in data filtering. Example: looking for medical help for a patient address. We know medical practitioner's address and each practitioner's willingness to travel.

App Search and Search UI support Geolocation fields and filtering on them.

App search API is the same as ES Rest API? / If I understand correctly, Search API embeds an elasticsearch instance where data are saved for search purposes? / Does App Search use elasticsearch at the back?

You could consider App Search as an overlay that abstracts away Elasticsearch for you. It provides API endpoints for ingestion, search, as well as a dashboard for relevance management.

Possible to get a link to the vue demo pls? / Do you have VueJS ui components?

We don't have Vue components yet, but we have a Vue demo:
https://elastic.github.io/vue-search-ui-demo/?size=n_20_n
Github: https://github.com/elastic/vue-search-ui-demo

In continuation to my query, I have tried using Site Search Crawling engine with Search UI following the docs. I couldn't fetch the results though, but I see the network operation hitting the search engine and fetch results are just not shown in any component or even custom component.

We would like to help you with this. Could you provide more details about your use case in our gitter? https://gitter.im/elastic-search-ui/community

How do you feel about integrating the generated React code with some PHP with the generated UI?

(Jason answered in the webinar)

How does the indexing of files work?

Indexing will depend on your search engine! App Search abstracts indexing away; it provides an API endpoint for ingestion and builds indices on an underlying Elasticsearch cluster. For the indices themselves, App Search make a lot of optimizations for relevance and speed, and the details are a bit out of scope for this webinar :)

Do we need to build a separate index for autocomplete to work?

No! App Search does that for you (re: Autocomplete)

I can't see if someone has asked this before, but could you possibly share a link for how to connect App Search / Search UI to existing elasticsearch backend engine / indices ?

To connect Search UI to Elasticsearch:
https://github.com/elastic/search-ui#how-do-i-use-this-with-elasticsearch
To ingest data from Elasticsearch to App Search, you can use Logstash:
https://www.elastic.co/guide/en/logstash/current/plugins-outputs-elastic_app_search.html

Other Resources & Links Shared in Chat

- https://codesandbox.io/embed/search-ui-national-parks-example-kdyms
- https://www.elastic.co/
- https://swiftype.com/
- https://constance.dev/seattle-indies-expo-search
- https://github.com/elastic/search-ui
- https://swiftype.com/search-ui
- https://www.elastic.co/blog/search-ui-1-0-0-released