

Disaster Management Case Study

Setting up Solr-powered data search for a disaster relief organization



Industry
Disaster Management

year founded
1990s

location
USA

The client is a disaster management center that works with government companies and non-profit organizations across the globe. It provides early warning, risk assessment, first response, and recovery efforts coordination technology for natural and man-made disasters (floods, earthquakes, hurricanes, etc.)

Challenge

When it comes to search, the client's system presents a number of challenges:

- Too much disaster-related data, too many duplicates (data comes from over 5,000 sources).
- Data providers use their own maps, datasets, and indexing rules. One of the challenges is that the data is poorly organized (no meta data exists for some documents, different formats are used).
- A need to tie all data points to their geographical locations.
- Bottom line: they needed a search solution that would index these disparate datasets and allow one to find relevant docs in the sea of data for a defined area on the map.

The client uses an ArcGIS-based solution for cartography. They also have proprietary systems written around the ArcGIS core. The goal is to be able to search through those datasets for documents that are (a) relevant to the query, and (b) relevant to the specified area.

Solution

1. Indexing app.

ObjectStyle built an application for indexing relevant datasets. It relies on meta data, where possible, to put together a meaningful/searchable index.

2. User interface (backend).

We also created a user interface for managing the indexing process and including/excluding selected data to/from index. The UI also lets one tweak search result titles. For example, one can name hurricane search results after the areas in which they occurred or according to their official names.

3. REST app for API access.

Since the client also provides data to partners through API, ObjectStyle also built a REST application that would allow partners to use the search functionality on their end.

4. New website and client-facing app.

We had to fine-tune Solr to rank data by distance. Solr has a smart way of determining which docs are more relevant and giving them more “weight.” This helps the client find the right data. In addition, each data instance should be matched to an outlined geographical area - be it a city or a tsunami.

So, there are two dimensions to search: the keyword and the area radius. This is how you find relevant data for a given situation.

Results

We handed a working search facility to the client. They are now testing the beta version.

Technology stack

- SolrCloud 7.3 (Zookeeper, AWS)
- Java
- Bootique
- PostgreSQL
- (Additional) JQuery, Bootstrap

Time Span and Resources

Duration:	10 months
Effort:	1650 man-hours