Federated Search with Drupal, Solr, and React

Matt Carmichael & Dan Montgomery • MidCamp 2019
Today’s Session

+ Introductions
+ Project Overview
+ Live Demo
+ Application Components
+ Getting Started
+ Question & Answer
Matt Carmichael

+ Engineer, Palantir.net
Dan Montgomery

+ Senior Engineer and Technical Architect, Palantir.net
Project Overview
Our Use Case

Existing Drupal 7 sites.
New Drupal 8 sites.
Domain Access.
No guarantees of common architecture.
Make it Simple!

Make it work like this.
Unlock your data with the power and relevance of Google Search, customized and secured in an appliance for use inside your business.
Add the power of Google.com technology to your website's search box, helping customers quickly find what they're looking for.
Unified Search

+ A simple way to store, retrieve, and parse content.
+ A cross-platform search application.
+ A speedy, usable, responsive front-end.
+ A flexible, extensible, reusable model.
+ A drop-in replacement for deprecated Google Products
Let’s Do It Live!
Three Part Solution
Solr components

Solr expects a **uniform schema** for proper results across an index.

An index can have multiple sources.

<table>
<thead>
<tr>
<th>Label</th>
<th>Machine Name</th>
<th>Type</th>
<th>Required?</th>
<th>Single/Multi-value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Federated Title</td>
<td>federated_title</td>
<td>string</td>
<td>Yes</td>
<td>single</td>
<td>The title of the item. Displayed as the title of each search result.</td>
</tr>
<tr>
<td>Federated Date</td>
<td>federated_date</td>
<td>date</td>
<td>No</td>
<td>single</td>
<td>Usually the date the content was created. Used to provide date-based filtering.</td>
</tr>
<tr>
<td>Federated Type</td>
<td>federated_type</td>
<td>string</td>
<td>Yes</td>
<td>single</td>
<td>The shared type label for faceting. Also used to label each result.</td>
</tr>
<tr>
<td>Federated Terms</td>
<td>federated_terms</td>
<td>string</td>
<td>No</td>
<td>multi</td>
<td>Terms for additional, topic-based facets, mapped to shared topic terms if necessary.</td>
</tr>
<tr>
<td>Federated Image</td>
<td>federated_image</td>
<td>string</td>
<td>No</td>
<td>single</td>
<td>An absolute url to an image which, if it exists, will be displayed with the text. Recommended image size: <strong>x</strong></td>
</tr>
<tr>
<td>Rendered HTML output</td>
<td>rendered_item</td>
<td>fulltext</td>
<td>Yes</td>
<td>multi</td>
<td>The full text of the item, with HTML stripped.</td>
</tr>
<tr>
<td>URI</td>
<td>url</td>
<td>string</td>
<td>Yes</td>
<td>single</td>
<td>The absolute path to the item, used to provide a link to each result.</td>
</tr>
<tr>
<td>Site Name</td>
<td>site_name</td>
<td>string</td>
<td>Yes</td>
<td>single</td>
<td>The descriptive name of the source site. Used to provide site-based filtering.</td>
</tr>
<tr>
<td>Site</td>
<td>site</td>
<td>string</td>
<td>Automatic</td>
<td>single</td>
<td>The base url of the source site, like <a href="https://labblog.uofmhealth.edu">https://labblog.uofmhealth.edu</a>. This will be sent automatically by Drupal, and is required for external content sources.</td>
</tr>
</tbody>
</table>
Solr already knows how to return JSON objects in response to a search request.
✅ Solr

Drupal

React
React applications love to consume JSON and format its results.
The existing library has the components, but not the design we need.

*It’s also MIT Licensed for reuse.*
We did usability testing around search components and how people use facets.
Our forked application was designed to be extended and skinned.

```javascript
export default {
  searchFields: {
    text: TextSearch,
    "list-facet": ListFacet,
    "range-facet": RangeFacet,
    "period-range-facet": RangeFacet,
    container: SearchFieldContainer,
    currentQuery: CurrentQuery
  },
  results: {
    result: Result,
    resultCount: CountLabel,
    header: ResultHeader,
    list: ResultList,
    container: ResultContainer,
    pending: ResultPending,
    preloadIndicator: PreloadIndicator,
    csvExport: CsvExport,
    paginate: ResultPagination
  },
  sortFields: {
    menu: SortMenu
  }
}
```
Templates use common markup that can be enhanced through a single CSS file.

```javascript
render() {
    const { doc, highlight } = this.props;

    return (<
        li onClick={() => this.props.onSelect(doc)}> 
            {doc.ss_federated_image && <div className="search-results__container--left"> <img src={doc.ss_federated_image} alt=""/> </div> }
            <div className="search-results__container--right"> <span className="search-results__label">{doc.ss_federated_type}</span> <h3 className="search-results__heading"><a href={doc.ss_url}>{doc.ss_federated_title}</a></h3> <div className="search-results__meta"> <cite className="search-results__citation">{doc.ss_site_name}</cite> {this.dateFormat(doc.ds_federated_date)} </div> <p className="search-results__teaser" dangerouslySetInnerHTML={{__html: highlight.tm_rendered_item}} /> </div> </li>)

}
Solr

Drupal

React
We have a working application.

We merely have to tell Solr how to index the content consistently.
Do you want to know a secret?
Solr

✅ CMS of choice

Drupal is not a hard requirement

✅ React

#midcamp @palantir
Drupal search templates simply call the React application.

Any HTML can.

```twig
{% set app_config = federated_search_app_config|json_encode(constant('JSON_UNESCAPED_SLASHES'), constant('JSON_NUMERIC_CHECK')) %}

<noscript>This search page requires Javascript in order to function. <a href="https://www.whatismybrowser.com/guides/how-to-enable-javascript/auto">Learn how to enable Javascript in your browser</a>.</noscript>

<div id="root" data-federated-search-app-config="{{ app_config }}">  
  <p class="element-invisible" aria-hidden="true">Error message..."/>  
</div>
```
✅ Solr

Drupal does make it easier

✅ React
Drupal components

Existing Search API and Search API Solr modules for Drupal 8 and Drupal 7.

Standard Search API config

Basic local Solr server
Federated field mapping

We use conversions to map disparate data to a standard schema.
Federated field mapping

Simple string mappings.
Federated field mapping

Token support.
Federated field mapping

Preview images can pull from existing or custom fields and support multiple media types.

Browse available tokens.
Cancel

Save
Federated field mapping

We can’t really use Drupal’s site name since it’s not user-friendly, also we need the site name for special things.
### Federated field mapping

Site names also integration with Domain Access for those who need it.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bundle filter</strong></td>
<td>Enabled</td>
</tr>
<tr>
<td><strong>Complete entity view</strong></td>
<td>Enabled</td>
</tr>
<tr>
<td><strong>Site Name</strong></td>
<td>Enabled</td>
</tr>
<tr>
<td><strong>Federated Field</strong></td>
<td>Enabled</td>
</tr>
<tr>
<td><strong>Re-map Field Names</strong></td>
<td>Enabled</td>
</tr>
</tbody>
</table>
| **Michigan Medicine Domain Label** | * Michigan Medicine  
Map the Domain to a custom label for search. |
| **CS Mott Children’s Hospital | Michigan Medicine Domain Label** | * CS Mott Children’s Hospital  
Map the Domain to a custom label for search. |
| **Frankel Cardiovascular Center | Michigan Medicine Domain Label** | * Frankel Cardiovascular Center  
Map the Domain to a custom label for search. |
| **Von Voigtlander Women’s Hospital | Michigan Medicine Domain Label** | * Von Voigtlander Women’s Hospital  
Map the Domain to a custom label for search. |
| **Kellogg Eye Center | Michigan Medicine Domain Label** | * Kellogg Eye Center  
Map the Domain to a custom label for search. |
| **Victors Care Domain Label** | * Victors Care  
Map the Domain to a custom label for search. |
Taxonomy fields

Some fields are uniform across sites and easy to map.

Taxonomy is more complex.
We can’t use the token system to map taxonomies and their hierarchies.
Hierarchies are mapped on the term itself, providing aliases that can be used across sites with unique vocabularies and terms.
Putting it all together

Drupal provides indexing, a search page callback, and a search box block.
How to Get Started
Open Source components

We provide a starting point that supports Drupal 8 and Drupal 7.

+ Federated Search Demo
  + https://github.com/palantirnet/federated-search-demo

+ macOS development environment
  + Composer
  + VirtualBox
  + Ansible
  + Vagrant
    + vagrant-hostmaster
    + vagrant-auto-network
Build Your Own

Items installed by the demo box.

+ Search API Federated Solr
  + https://www.drupal.org/project/search_api_federated_solr

+ Search API Field Map
  + https://www.drupal.org/project/search_api_field_map (d8)

+ Federated Search React
  + https://github.com/palantirnet/federated-search-react

+ Drupal 8 and Drupal 7

+ Ubuntu 64bit version 16.04.4 LTS

+ Apache HTTP version 2.x

+ MySQL version 5.7.21

+ PHP version 7.1.16

+ Apache Solr version 4.5.1 (Acquia version)
Questions?
Thank You.

622 Davis Street, Suite 400
Evanston, IL 60201
773.645.4100

Ken Rickard
Avi Schwab
Jes Constantine
Michelle Jackson
Kelsey Bentham
Malak Desai
Luke Wertz
Kristen Mayer
Adrian Muresan
Bec White
James Sansbury
The University of Michigan
Thank You.

622 Davis Street, Suite 400
Evanston, IL 60201
773.645.4100

Matt Carmichael
Engineer

Dan Montgomery
Senior Engineer and
Technical Architect
PLEASE PROVIDE YOUR FEEDBACK!

mid.camp/272

The top rated sessions will be captioned, courtesy of Clarity Partners
CONTRIBUTION DAY
Saturday 10am to 4pm

You don’t have to know code to give back!

New Contributor training 10am to Noon
with AmyJune Hineline of Kanopi Studios