

# SharePoint Search

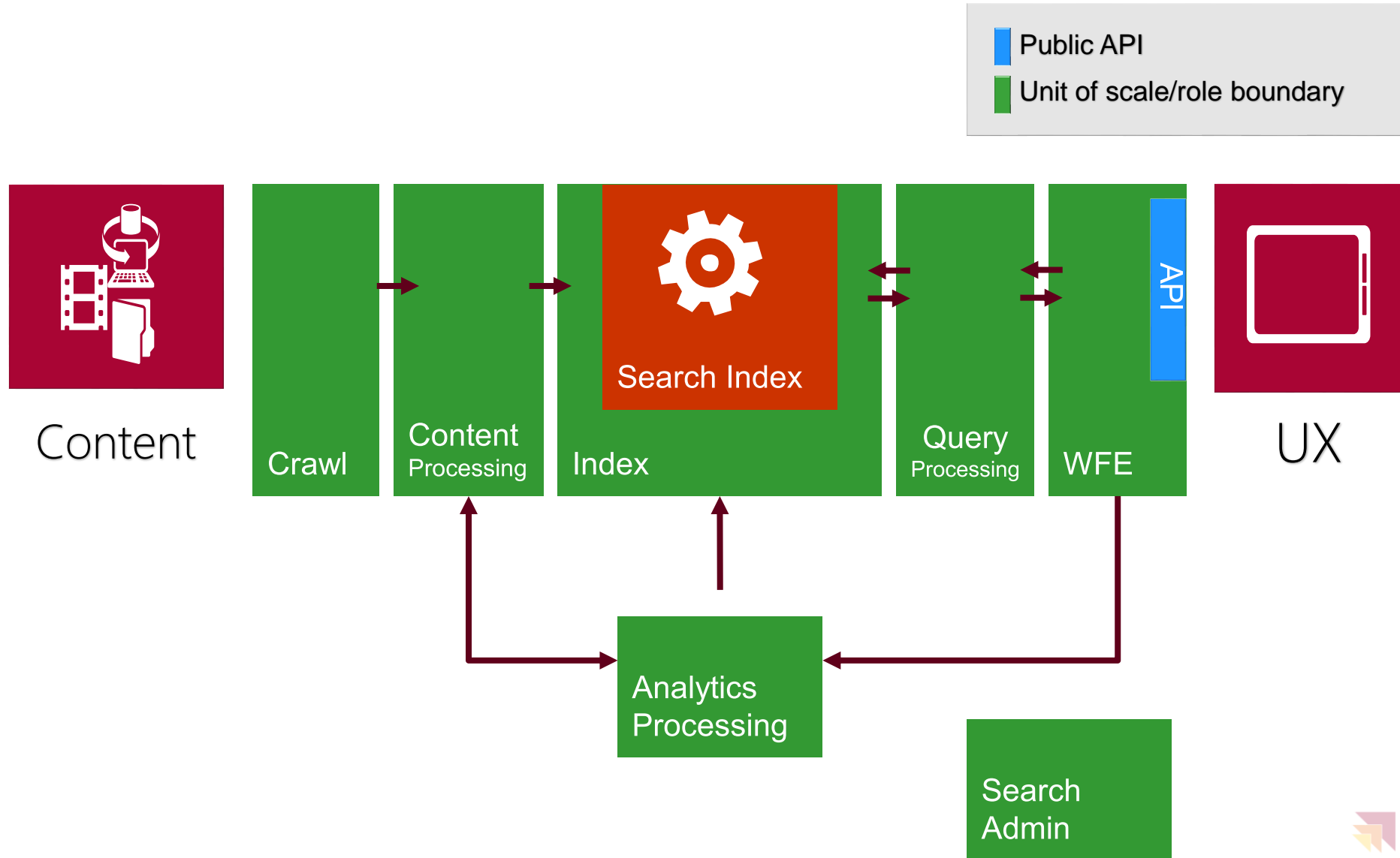


# Agenda

- SharePoint 2016 Search Architecture
- Managed Properties
- Query Execution using KQL
- Result Sources and Query Rules
- Result Types and Custom Display Templates
- Executing Searches Programmatically



# SharePoint 2016 Search Architecture



# Building Blocks for Extending Search

- Managed Properties
- Result Sources
- Query Rules
- Result Types
- Search Navigation



# Configuring the SharePoint Search Service

- Central Administration (SSA)
  - Content Sources
  - Result Sources
  - Query Rules
  - New Managed Properties
- Site Collection and Site
  - Result Sources
  - Result Types
  - Query Rules
  - Reassign Managed Properties







**DEMO**

# Search Administration

# Agenda

- ✓ SharePoint 2016 Search Architecture
- Managed Properties
  - Query Execution using KQL
  - Result Sources and Query Rules
  - Result Types and Custom Display Templates
  - Executing Searches Programmatically



# Managed Properties

- Mapped to Crawled Properties
  - Tenancy level
  - Site Collection level
- Site Columns Become Managed Properties
  - Publishing and Multiple Lines prefixed with `ows_r_<four-letter-code>`
  - Managed Metadata prefixed with `ows_taxId`
  - All others prefixed with `ows_q_<four-letter-code>`





# Managed Properties

Property	Description
Queryable	Managed Property can be used in property-based searches
Refinable	Managed Property can be used as a refiner
Retrievable	Managed Property can be returned in the results
Searchable	Includes the value of the Managed Property in the Search Index
Sortable	Managed Property can be used for sorting results
Type	The data type of the Managed Property
Safe	Can be returned to anonymous users
Alias	Friendly names
Multi-valued	Managed Property can have multiple values





**DEMO**

**Managed Properties**

# Agenda

- ✓ SharePoint 2016 Search Architecture
- ✓ Managed Properties
- Query Execution using KQL
  - Result Sources and Query Rules
  - Result Types and Custom Display Templates
  - Executing Searches Programmatically



# Query Language Options

- Keyword Query Language (KQL)
  - Default query language
- FAST Query Language (FQL)
  - Available, though disabled by default
- SQL Query Language
  - Removed from SharePoint 2013





- Free text
  - SharePoint
- Wildcard
  - Share\*
- Managed Property
  - Author:McD\*, ContentType:Image
- NEAR, ONEAR
  - SharePoint NEAR Social





**DEMO**

**Search Queries**

# Agenda

- ✓ SharePoint 2016 Search Architecture
- ✓ Managed Properties
- ✓ Query Execution using KQL
- Result Sources and Query Rules
  - Result Types and Custom Display Templates
  - Executing Searches Programmatically



# Result Sources

- Select a Source
  - Local SharePoint Index
  - Remote SharePoint Index
  - OpenSearch
  - Exchange
- Apply a Query Transformation
  - Defines a subset of the source

The screenshot displays the 'Build Your Query' interface, which is divided into two main sections: 'Build Your Query' on the left and 'SEARCH RESULT PREVIEW' on the right.

**Build Your Query Section:**

- BASICS** (selected), SORTING, TEST
- Keyword filter:** A dropdown menu labeled 'Query after all transformations' with an 'Add keyword filter' button below it.
- Property filter:** A dropdown menu labeled 'Author' with a 'Contains' dropdown and a 'Name of the user who runs' dropdown, followed by an 'Add property filter' button.
- Query text:** A text area containing the query: `{searchTerms} ContentClass:STS_ListItem_Tasks AssignedToOWSUSER:(User.Name) PercentCompleteOWSNMBR<->'100'`. The text is highlighted in yellow.
- Test query:** A button at the bottom right of the 'Build Your Query' section, circled in red.

**SEARCH RESULT PREVIEW Section:**

- RelevantResults (2):** A list of two search results.
  - [Create a search-based sample app](#)  
`connect2014.sharepoint.com/Lists/DispForm.aspx?ID=1`
  - [Create a Search Center sample](#)  
`connect2014.sharepoint.com/Lists/DispForm.aspx?ID=2`

A red arrow points from the 'Test query' button to the 'SEARCH RESULT PREVIEW' section.





# Query Rules

- Applied to a given Result Source
- Processed under specified conditions
  - Keyword match
  - Topic category match
- Affects the query results
  - Alter the user's query
  - Generate a Result Block



# Query Rules

- Intercept and change a users query
- Promote Special Content
  - Best Bets
  - Detect the User
  - Act on user “intent”
  - Perform another query
  - Create banner ads
- Change the rank of results
- So much more...





**DEMO**

# Query Rules

# Agenda

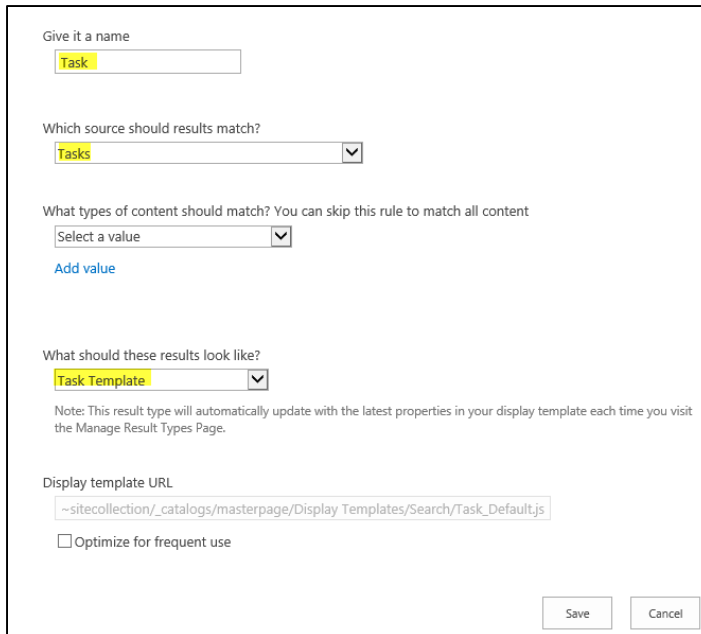
- ✓ SharePoint 2016 Search Architecture
- ✓ Managed Properties
- ✓ Query Execution using KQL
- ✓ Result Sources and Query Rules
- Result Types and Custom Display Templates
  - Executing Searches Programmatically





# Result Types

- Bound to a given Result Source
- Defined by a rule
- Associated with a template for display
  - Determines how Result Type appears
  - Created as an HTML file referencing Managed Properties



Give it a name

Task

Which source should results match?

Tasks

What types of content should match? You can skip this rule to match all content

Select a value

[Add value](#)

What should these results look like?

Task Template

Note: This result type will automatically update with the latest properties in your display template each time you visit the Manage Result Types Page.

Display template URL

~sitecollection/\_catalogs/masterpage/Display Templates/Search/Task\_Default.js

☐ Optimize for frequent use

Save Cancel



# Display Templates

- HTML Template for
  - Search Result
  - Refiner
  - Content By Search
  - Controls
  - Containers
  - Etc...
- Upload HTML and SharePoint Converts it to \*.JS
- Use any editor
  - NotePad++, Dreamweaver...
  - even SharePoint Designer!



# Display Templates

```
<div id="_#=$htmlEncode(itemId) =#_" name="Item" data-displaytemplate="TaskItem"
  class="ms-srch-item" onmouseover="_#= ctx.currentItem_ShowHoverPanelCallback =#_"
  onmouseout="_#= ctx.currentItem_HideHoverPanelCallback =#_">
  <h3>
    
    <span>_#=ctx.CurrentItem.Title=#_</span><br />
    <span>Assigned To: _#=ctx.CurrentItem.AssignedTo=#_</span><br />
    <span style="_#=getDueDateStyle()=#_">Due Date: _#=getFormattedDate()=#_</span>
  </h3>
  <div id="_#=$htmlEncode(hoverId) =#_" class="ms-srch-hover-outerContainer"></div>
</div>
```

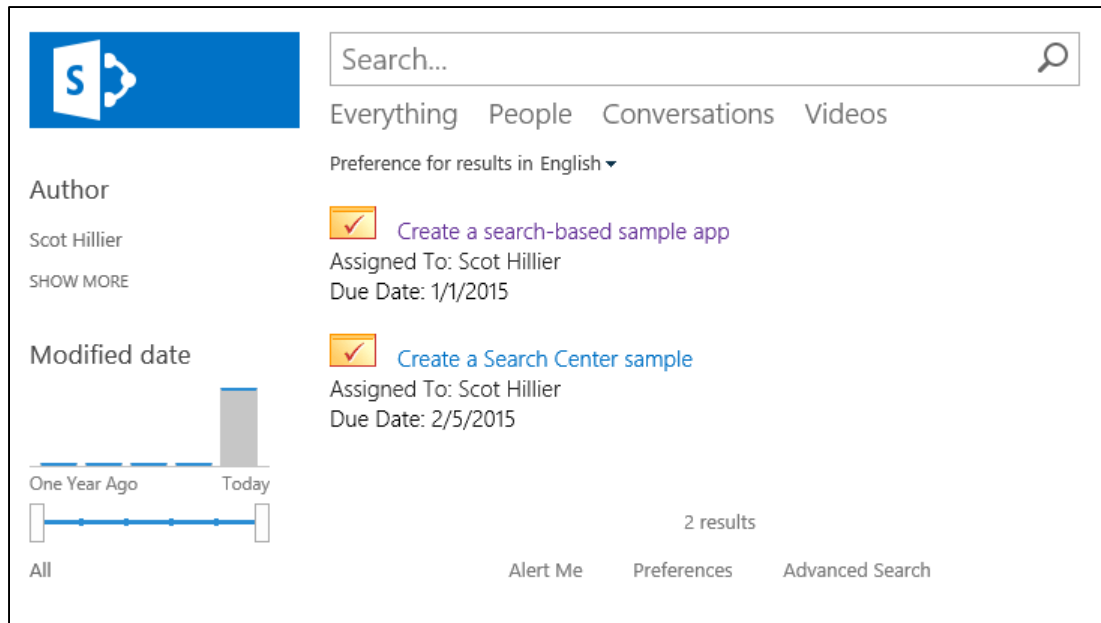
```
getFormattedDate = function() {
  var d = new Date(ctx.CurrentItem.DueDateOWSDATE);
  return (d.getMonth() + 1) + "/" + d.getDate() + "/" + d.getFullYear();
};

getDueDateStyle = function() {
  var d1 = new Date(ctx.CurrentItem.DueDateOWSDATE);
  var d2 = new Date();
  if(d2>d1)
    return 'color:red';
  else
    return 'color:black';
};
```



# Custom Search Results

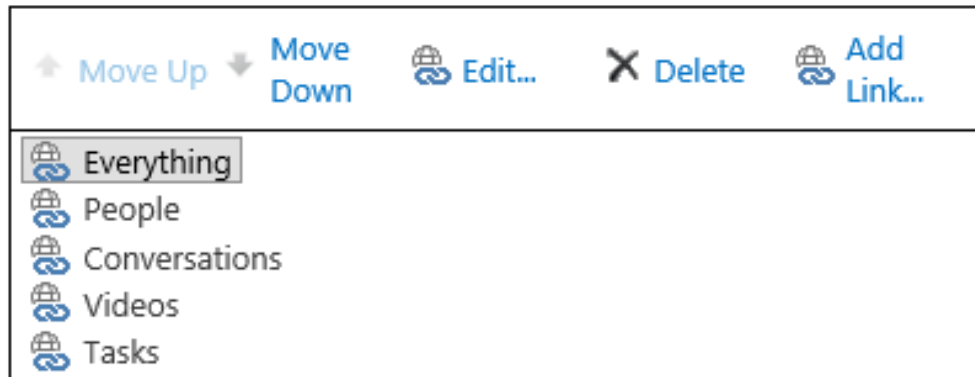
- Use building blocks to create custom solutions in Search Center





# Search Navigation

- Provides navigation across pages in Search Center



# Export/Import Search Settings

- Configuration export
  - SearchConfiguration.xml
  - Handles rules, sources, managed properties, etc.
  - Does not handle master pages, templates, and web parts
- Configuration import
  - Import into site, site collection, tenancy
  - Programmatically import with app





**DEMO**

## **Display Templates and Result Types**

# Agenda

- ✓ SharePoint 2016 Search Architecture
- ✓ Managed Properties
- ✓ Query Execution using KQL
- ✓ Result Sources and Query Rules
- ✓ Result Types and Custom Display Templates
- Executing Searches Programmatically





# REST Endpoint

Keywords

`https://tenant/site/_api/search/query?querytext='{KQL Query}'`

Selecting Properties

`https://tenant/site/_api/search/query?querytext='test'&selectproperties='Title,Rank'`

Sorting

`https://tenant/site/_api/search/query?querytext='test'&sortlist='LastModifiedTime:descending'`



# JavaScript and REST

```
$.ajax(  
    {  
        url: "http://site/_api/search/" +  
            "query?querytext='{KQL Query}'",  
        method: "GET",  
        headers: {  
            "accept": "application/json;odata=verbose",  
        },  
        success: onSuccess,  
        error: onError  
    }  
);
```



# JavaScript and CSOM

```
var context = SP.ClientContext.get_current();

var keywordQuery =
new Microsoft.SharePoint.Client.Search.Query.KeywordQuery(context);

keywordQuery.set_queryText("sharepoint");

var searchExecutor =
new Microsoft.SharePoint.Client.Search.Query.SearchExecutor(context);

results = searchExecutor.executeQuery(keywordQuery);

context.executeQueryAsync(onGetEventsSuccess, onGetEventsFail);
```



# C# and REST

```
var spContext = SharePointContextProvider.Current.GetSharePointContext(HttpContext);
string accessToken = spContext.UserAccessTokenForSPAppWeb;

string url = "http://site/_api/search/query?querytext='sharepoint'";

HttpClient client = new HttpClient();
HttpRequestMessage request = new HttpRequestMessage(HttpMethod.Get, url);
request.Headers.Accept.Add(new MediaTypeWithQualityHeaderValue("application/xml"));
request.Headers.Authorization = new AuthenticationHeaderValue("Bearer", accessToken);

HttpResponseMessage response = await client.SendAsync(request);
string responseString = await response.Content.ReadAsStringAsync();
```



# C# and CSOM

```
var spContext = SharePointContextProvider.Current.GetSharePointContext(Context);

using (var cctx = spContext.CreateUserClientContextForSPHost())
{
    KeywordQuery query = new KeywordQuery(cctx);
    query.QueryText = "{KQL}";
    SearchExecutor executor = new SearchExecutor(cctx);
    ClientResult<ResultTableCollection> results = executor.ExecuteQuery(query);
    cctx.ExecuteQuery();
}
```





# App Permissions

```
<AppPermissionRequest  
  Scope=http://sharepoint/search  
  Right="QueryAsUserIgnoreAppPrincipal" />
```

Do you trust Employee Directory?

Let it access basic information about this site.

Let it access basic information about the users of this site.

Allow this app to execute search queries on your behalf, ignoring the app's permissions on result items.



Employee Directory

Trust It

Cancel



# Summary

- ✓ SharePoint 2016 Search Architecture
- ✓ Managed Properties
- ✓ Query Execution using KQL
- ✓ Result Sources and Query Rules
- ✓ Result Types and Custom Display Templates
- ✓ Executing Searches Programmatically

