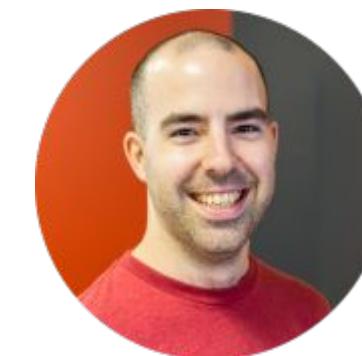


# Coveo for Sitecore 4

Developer Training  
September 2018



**Jean-François L'Heureux**  
Tech Evangelist, Sitecore MVP  
[@jflh](https://twitter.com/jflh)

# Housekeeping

## Hands on training

Theory, Demo, Lab, Repeat...

Do not be shy to say when you are not done with a lab. We want nobody lagging behind.

## Make It interactive

It's your training: Ask questions, suggest ideas...

If you already have experience with Coveo, try to go further!

## There is more online

[Developers.coveo.com](http://Developers.coveo.com)

[OnlineHelp.coveo.com](http://OnlineHelp.coveo.com)

[Answers.coveo.com](http://Answers.coveo.com)

# Agenda

## 1. Administration

- Installation
- Best Practices
- Architecture
- Coveo Cloud

## 2. Customization

- Creating a Search Interface
- Global Searchbox
- Indexing Scope
- Coveo JavaScript Search Framework
- Query Syntax
- Free-Text Search
- External Content
- CSS

## 3. Personalization & Usage Analytics

- xDB Profiles Basics
- Personalization of the Search Page Layout & Results
- Coveo Cloud Usage Analytics & Sitecore xDB
- Query Pipelines
- Machine Learning

## 4. Scaling, Upgrading & Troubleshooting

- Index Redundancy
- Scaling CD and CM servers
- Upgrading Sitecore & Coveo for Sitecore
- Standard Troubleshooting



## 1. Administration

# 1. Administration

1.Training Requirements & Materials

2.Coveo for Sitecore

- a. Installation
- b. History
- c. Editions
- d. Architecture Overview
- e. Configuration Files

3.Web Site Search Best Practices

4.Sitecore and Search

5.Coveo Cloud

- a. Administration Console



## Training Requirements & Materials

# Training Requirements and Materials

## Requirements:

- A computer or virtual machine with:
  - SQL Server 2008 R2 SP1 or newer
  - MongoDB 3.X or 2.6.X running as a Windows service
  - [Sitecore Experience Platform 8.2 Update-5 \(rev. 170728\)](#)
  - Launch Sitecore 8.2 MVC
  - Sitecore published to the web database
  - At least 5GB of free space the C drive

## Provided materials to download:

- Training slides and materials in Google Drive:
  - Coveo for Sitecore package (Coveo for Sitecore 82 4.1.XXX.XX.zip)
  - Visual Studio solution (Coveo.Training.LaunchSitecore MONTH YEAR.zip)

# Solutions ZIP Archive

This training lab solutions are included in the Coveo.Training.LaunchSitecore MONTH YEAR.zip archive. It contains:

1. A Visual Studio 2012 project with one class for advanced functionality
2. A Website folder with:
  - a. All the code and configuration examples
  - b. A bin folder with the precompiled output assembly of the project

If you want to compile the project, simply change the project references to your local Sitecore instance bin folder.

# Extracting the Coveo.Training.LaunchSitecore Archive

1. Extract the "Coveo.Training.LaunchSitecore MONTH YEAR.zip" archive somewhere on your hard drive
  - a. It is recommended to extract it at the root of one of your hard drives as it contains long folder paths

# Lab

1. Extract the Coveo.Training.LaunchSitecore archive on your hard drive

# Validating Sitecore

1. Open your web browser
2. Navigate to  
<http://<YourSitecoreInstanceName>>
3. Validate that the Launch Sitecore website is displayed without error

If the Launch Sitecore site isn't displayed, smart publish the site in Sitecore

If the carousel is missing the images, rebuild the Lucene master and web indexes





# Installation

# Installation

Coveo for Sitecore

Coveo for Sitecore 82 4.1.XXX.XX.zip

**WARNING:** Make sure you are following the below procedure and use the proper package for your Sitecore version!

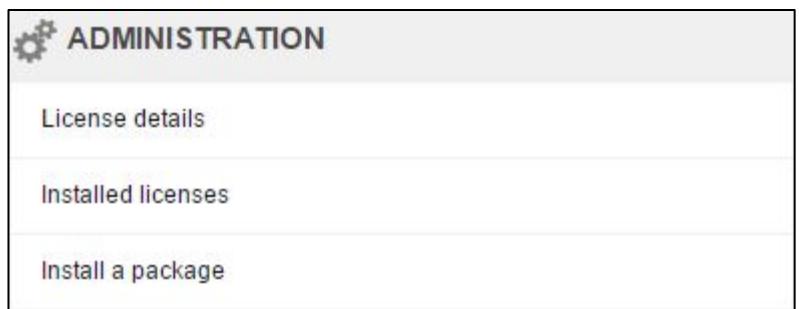
Time to get started!

<https://developers.coveo.com/display/SitecoreV4/Coveo+for+Sitecore+Installation+Guide>

# Installation - Coveo for Sitecore

## 1. Open the Sitecore Installation Wizard

- a. Sitecore Control Panel > Administration > Install a package



## 2. Click Upload package button

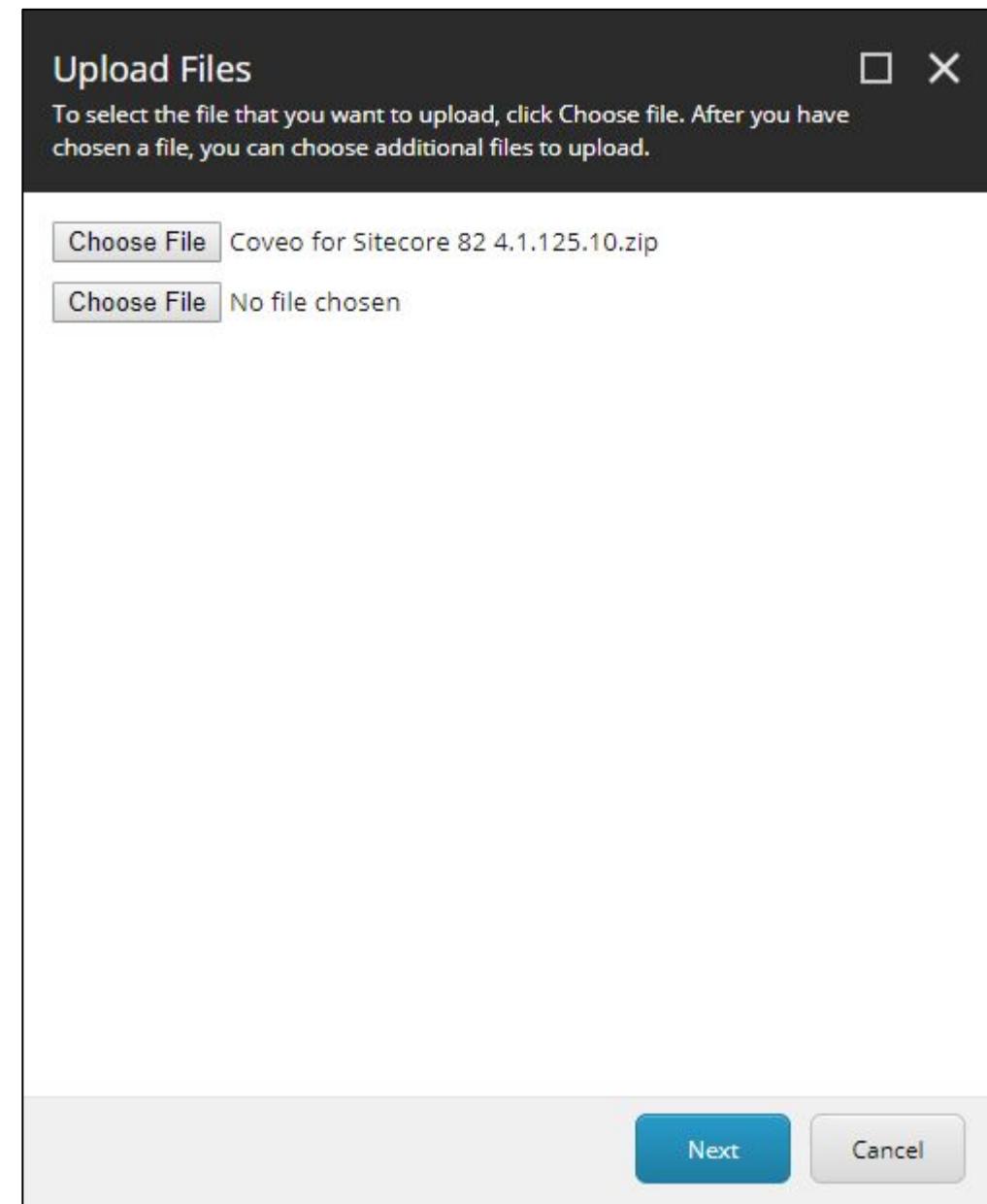
## 3. Choose Coveo for Sitecore 82 4.1.XXX.XX.zip

## 4. Click Next, Upload, Close and Next buttons

## 5. Accept the license agreement

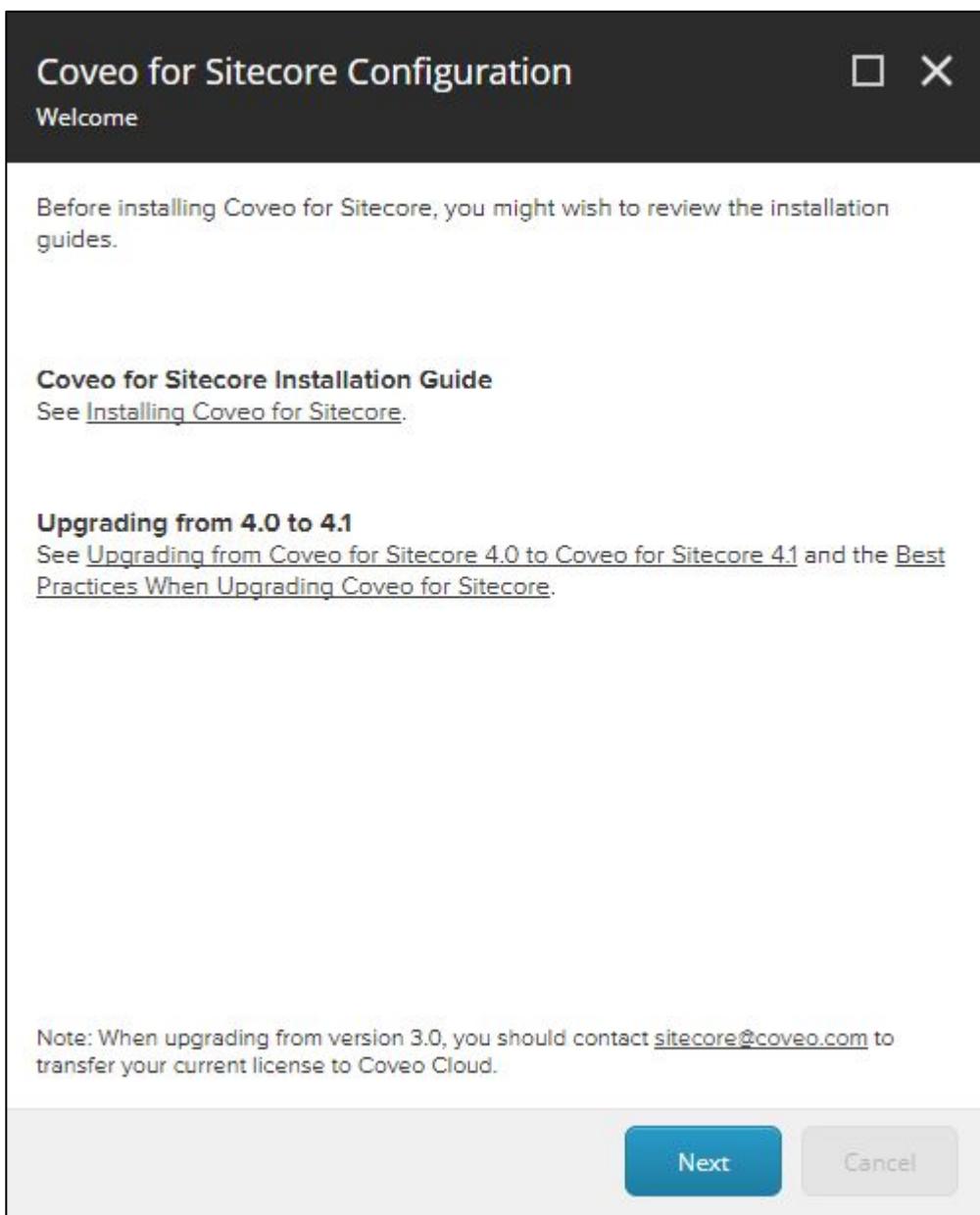
## 6. Click Next and Install buttons

- a. Wait for the package to install



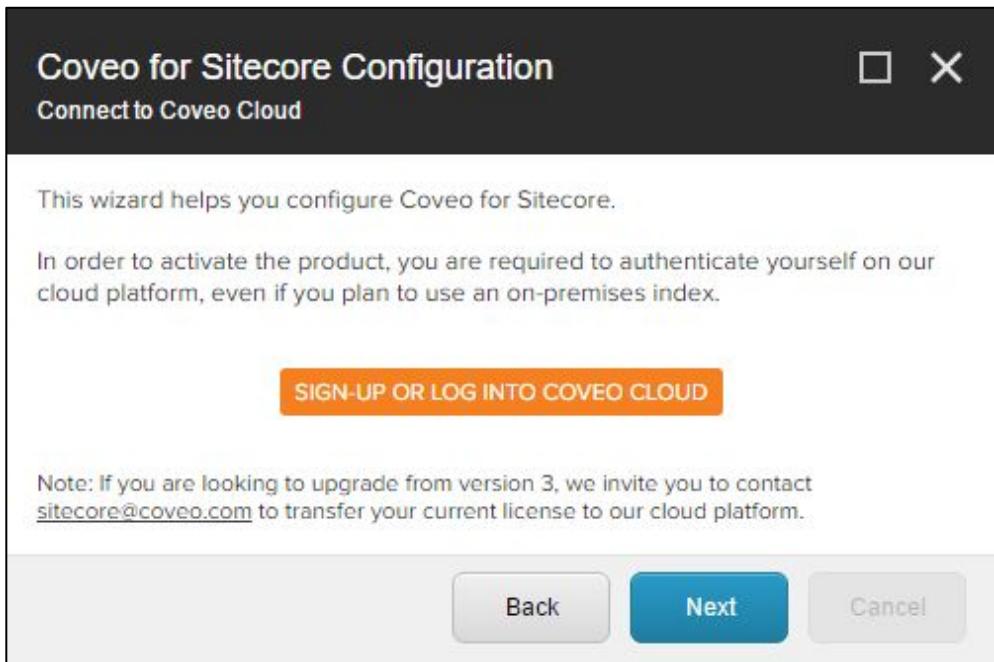
# Installation - Coveo for Sitecore

7. Read the welcome screen
8. Click Next button



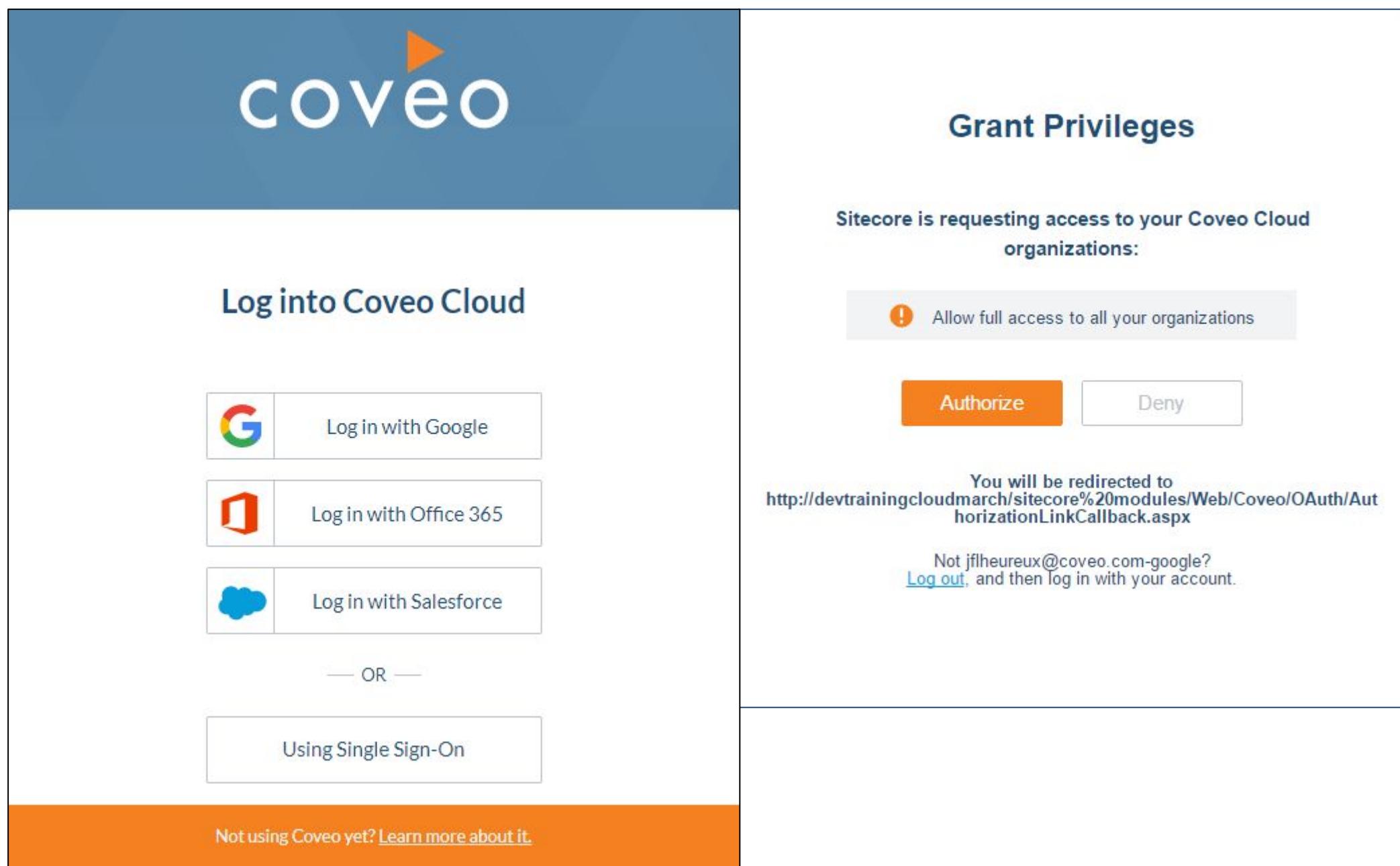
# Installation - Coveo for Sitecore

9. Click "Sign-Up Or Log Into Coveo Cloud" button
  - a. A new tab will open for you to sign-up or log into Coveo Cloud



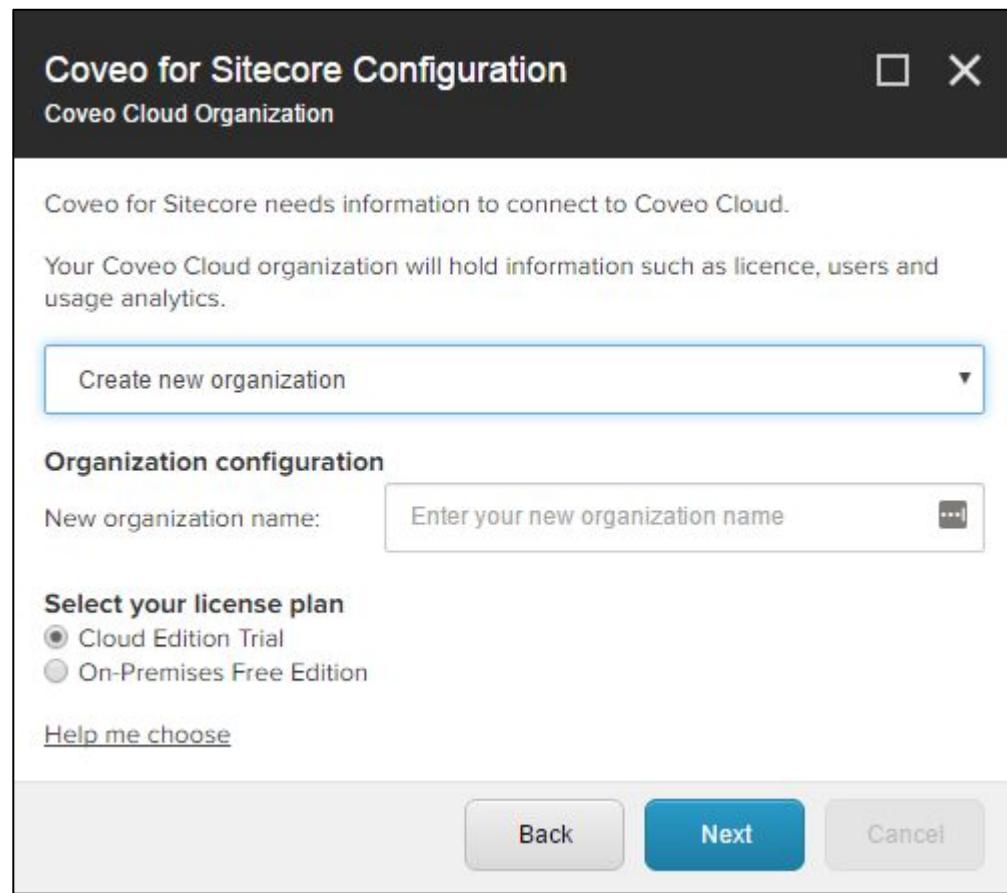
# Installation - Coveo for Sitecore

10. Log into Coveo Cloud using one provider
11. Click Authorize to allow Sitecore full access to all your organizations
  - a. The new tab will close and return to the Coveo for Sitecore Configuration wizard



# Installation - Coveo for Sitecore

12. In the "Select existing organization" dropdown, select "Create new organization"
13. Enter a name for your new Coveo Cloud organization
  - a. For the training, it is suggested to include your name in your organization name
14. Choose the Cloud Edition Trial license plan
15. Click Next button

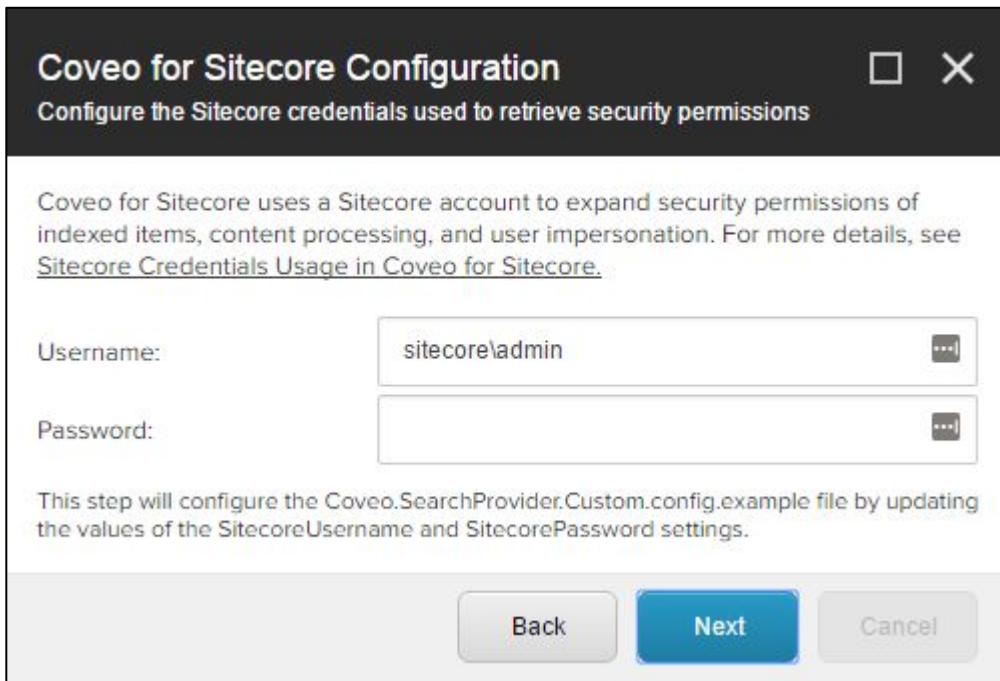


# Installation - Coveo for Sitecore

**16.** Enter the credentials of a Sitecore administrator

- a. Username: sitecore\admin
- b. Password: The password of your sitecore\admin user

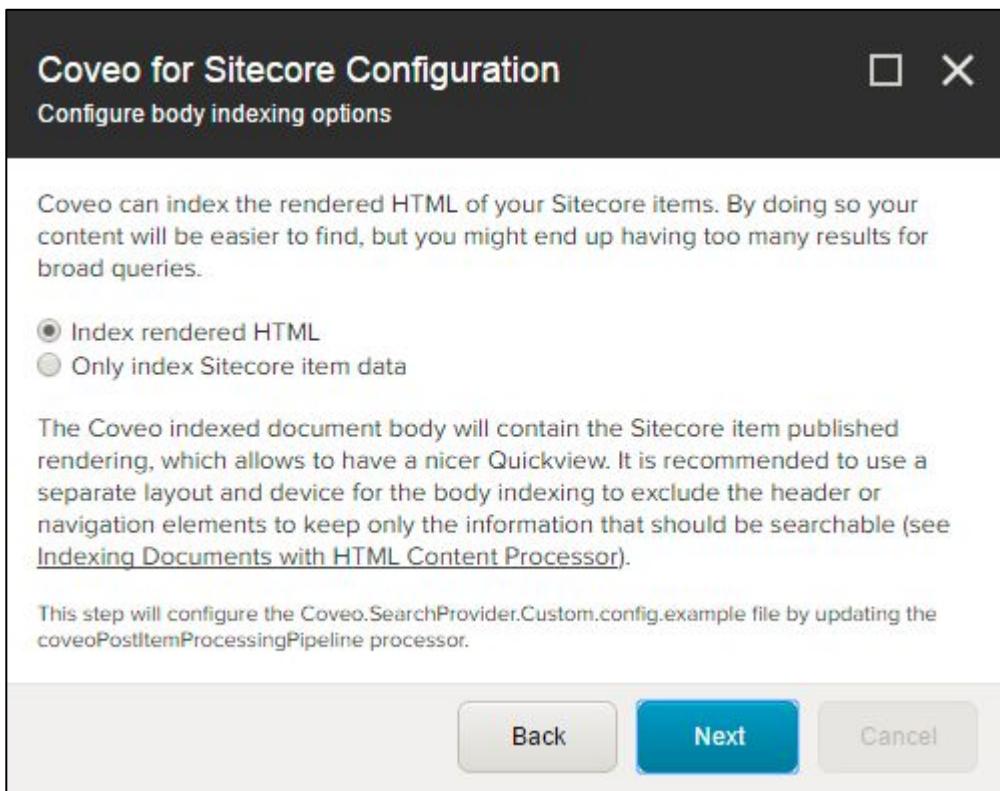
**17.** Click Next button



The password can be changed later in the Control Panel of Sitecore

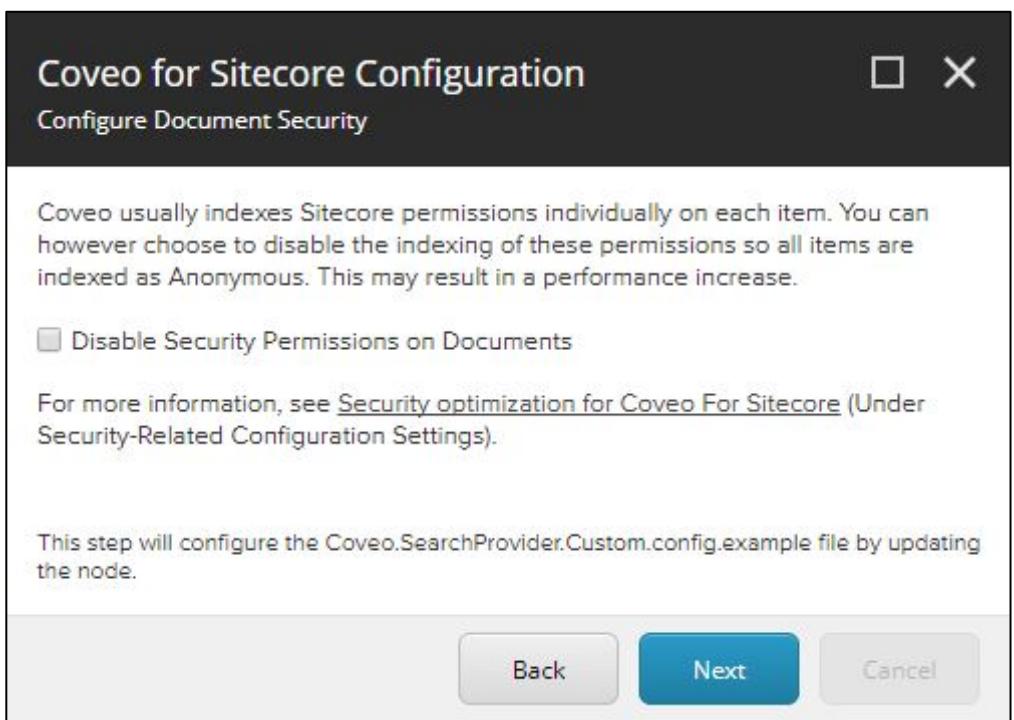
# Installation - Coveo for Sitecore

18. Choose "Index rendered HTML" option
19. Click Next button



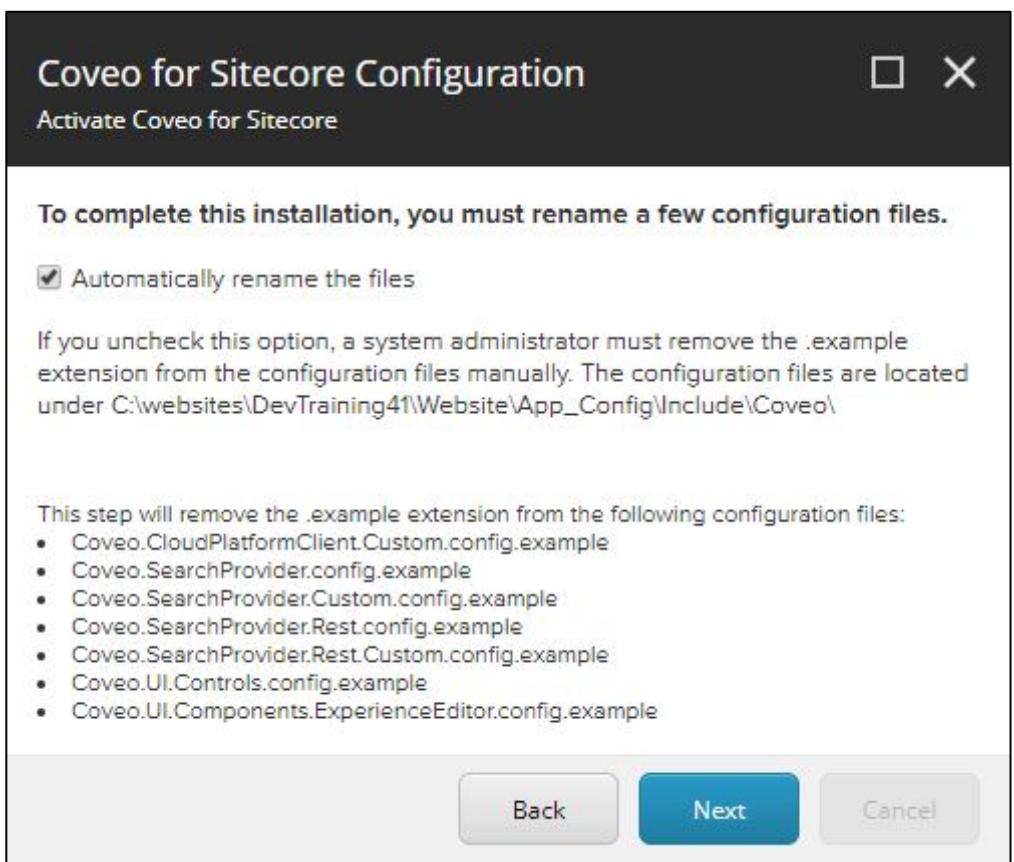
# Installation - Coveo for Sitecore

20. Do not change anything in this page
21. Click Next button



# Installation - Coveo for Sitecore

- 22.** Ensure the "Automatically rename the files" option is checked
- 23.** Click Next button

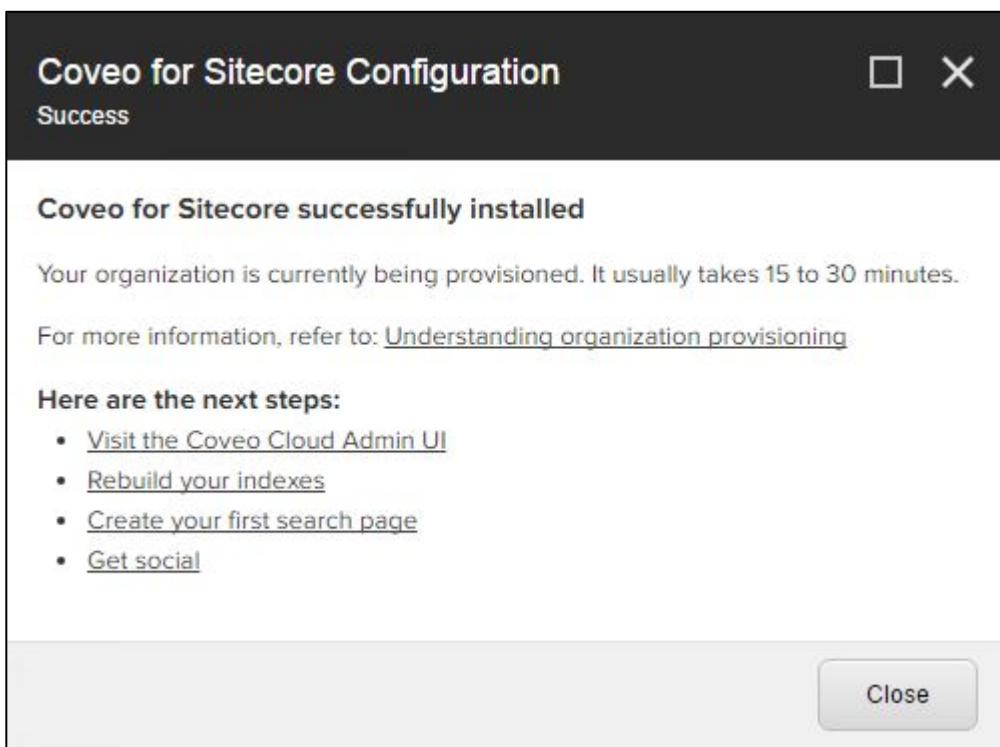


# Installation - Coveo for Sitecore

**24.** Read the success screen

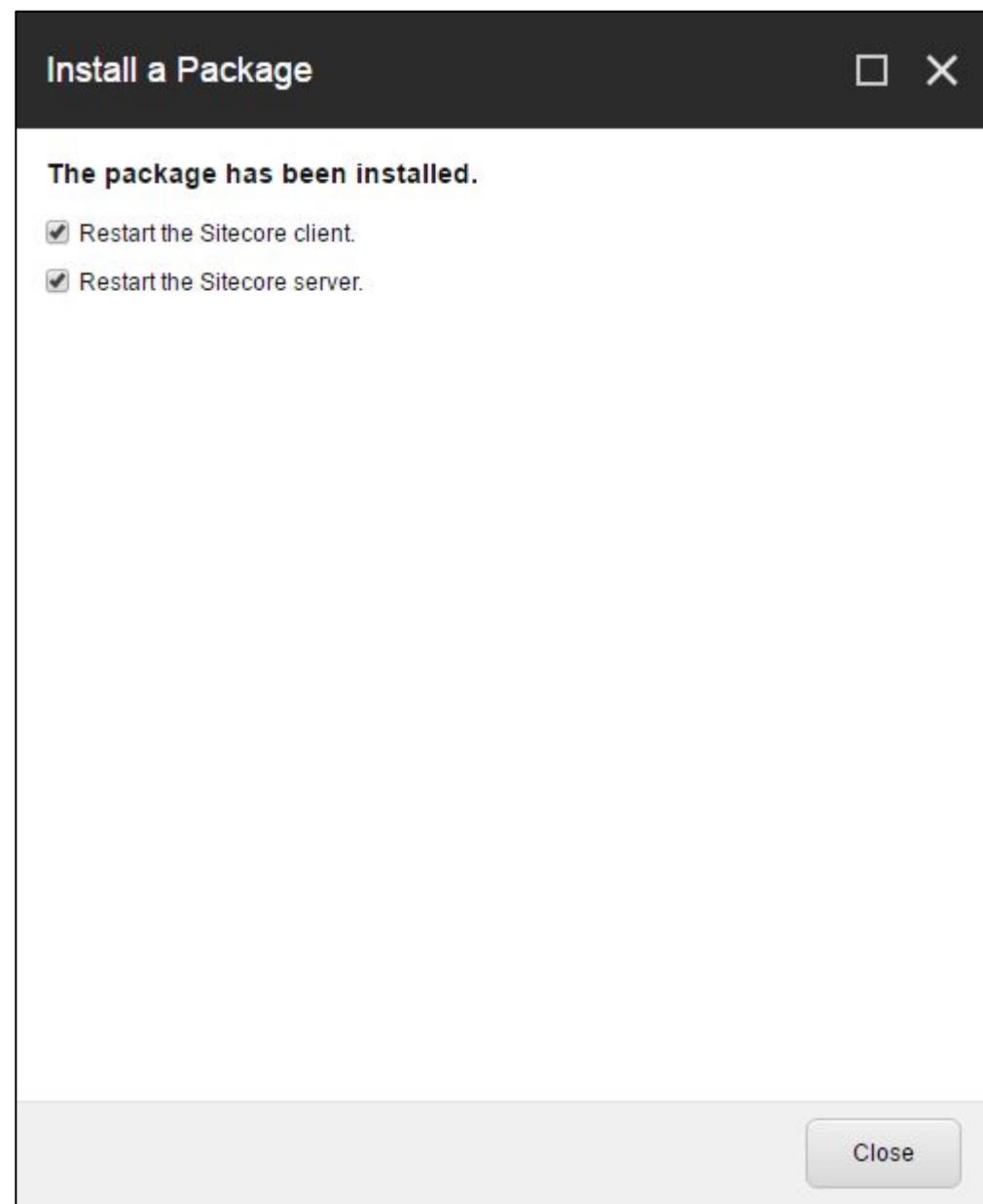
**25.** Click Close button

a. The Install a Package wizard will be displayed again



# Installation - Coveo for Sitecore

26. Ensure the 2 restart options are checked
27. Click Close button
  - a. Sitecore doesn't reload the Control Panel automatically
28. Reload the page in the browser



# Launch Sitecore Compatibility Issue

Launch Sitecore extensively uses the xDB and is throwing errors when the analytics context is null.

Coveo for Sitecore sets the analytics context to null when indexing items with a layout to avoid logging usage analytics for every indexed item.

The combination of both is causing all indexed pages to be error pages.

Also, Launch Sitecore replaces spaces in item names with hyphens. This prevents the creation of Coveo for Sitecore example search pages.

# Creating a Configuration Patch File

1. From the file system, browse to the Include configuration folder
  - a. <your sitecore instance folder>\Website\App\_Config\Include
2. Create and open a "Training" folder
3. Create and open a "LaunchSitecoreFix.config" file

# Patch the Launch Sitecore Compatibility Issue

## 1. Initialize the file with this configuration

```
<configuration xmlns:patch="http://www.sitecore.net/xmlconfig/">
  <sitecore>
    <events>
      <event name="item:saving" >
        <handler type="LaunchSitecore.Configuration.Events.ItemSaving.ItemNameHyphenHandler, LaunchSitecore" method="OnItemSaving">
          <patch:delete />
        </handler>
      </event>
    </events>
    <pipelines>
      <startAnalytics>
        <processor type="Coveo.Analytics.Processors.ExcludeCoveoUserAgentProcessor, Coveo.Analytics">
          <patch:delete />
        </processor>
      </startAnalytics>
    </pipelines>
  </sitecore>
</configuration>
```

## 2. Save the file

# Rebuilding the Coveo Indexes

## 1. Open the Sitecore Indexing Manager

- Sitecore Control Panel > Indexing > Indexing manager



## 2. Select only the 2 Coveo indexes

## 3. Click Rebuild button

A screenshot of the Sitecore Indexing Manager dialog box. The title bar says "Indexing Manager" and "Select the search indexes that you want to rebuild".

**Rebuild search index**

[Select all](#) | [Unselect all](#)

**Local Indexes**

<input checked="" type="checkbox"/> Coveo_master_index	<b>Coveo_master_index</b>
<input checked="" type="checkbox"/> Coveo_web_index	Rebuild Time: Never Run
<input type="checkbox"/> sitecore_core_index	Approximate Throughput: 0 items per second
<input type="checkbox"/> sitecore_master_index	Has Deletions: False
<input type="checkbox"/> sitecore_web_index	Is Clean: True
<input type="checkbox"/> sitecore_marketing_asset_index_master	Out of Date : False
<input type="checkbox"/> sitecore_marketing_asset_index_web	Document Count : 0
<input type="checkbox"/> sitecore_marketingdefinitions_master	Is Healthy : True
<input type="checkbox"/> sitecore_marketingdefinitions_web	Number of Fields : 0
<input type="checkbox"/> sitecore_testing_index	Last Updated : 1/1/0001 - 12:00 AM (UTC)
<input type="checkbox"/> sitecore_suggested_test_index	Number of Terms : 0
<input type="checkbox"/> sitecore_fxm_master_index	
<input type="checkbox"/> sitecore_fxm_web_index	
<input type="checkbox"/> sitecore_list_index	
<input type="checkbox"/> social_messages_master	
<input type="checkbox"/> social_messages_web	

**Index statistics**

<b>Coveo_master_index</b>	Rebuild Time: Never Run
Approximate Throughput: 0 items per second	Has Deletions: False
Is Clean: True	Out of Date : False
Document Count : 0	Is Healthy : True
Number of Fields : 0	Last Updated : 1/1/0001 - 12:00 AM (UTC)
Number of Terms : 0	Number of Terms : 0

**Coveo\_web\_index**

Rebuild Time: Never Run	Approximate Throughput: 0 items per second
Has Deletions: False	Is Clean: True
Is Clean: True	Out of Date : False
Document Count : 0	Is Healthy : True

**Buttons:** [Rebuild](#) | [Cancel](#)

# Accessing the Coveo Cloud Administration Console

Coveo Cloud organization, index and usage analytics data are managed from the Coveo Cloud Administration Console, which is a web application

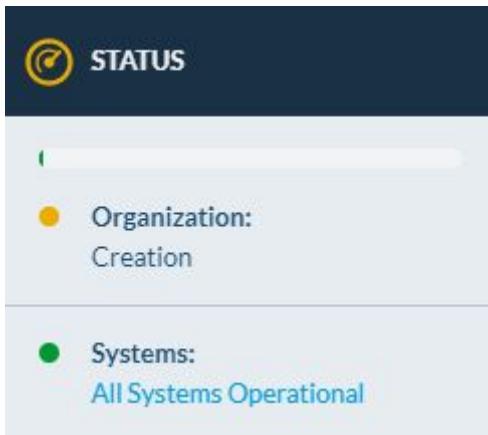
To open the Coveo Cloud Administration Console:

- 1.** Either:
  - Click "Cloud admin UI" link in the Sitecore Control Panel
    - Sitecore Control Panel > Coveo Search > Cloud admin UI
  - In a web browser, navigate to <https://platform.cloud.coveo.com>
- 2.** Log in using the same account and provider you used at installation time

# Organization Provisioning

A new Coveo Cloud organization need to be provisioned before being able to index documents and answer search queries. This takes about 15 to 30 minutes.

1. Log into the Coveo Cloud Administration Console
2. Hover the STATUS section in the bottom-left corner to see the status of your organization and of Coveo Cloud in general



# Lab

1. Install Coveo for Sitecore
2. Patch the Sitecore configuration
3. Rebuild your Coveo indexes
4. Open the Cloud Administration Console
5. Validate the status of your organisation



# Architecture

# Web Site Search - Best Practices

1. The Search Box
2. The Search Page
3. Search Driven Content Page
4. Search Analytics & Reporting

# Your Customers Expect a Good Search Experience

When people visit your site, they expect to find relevant information with ease. So it's no surprise that the sites that deliver great search experiences perform better: longer time-on-site, more repeat visitors, more conversions, and higher satisfaction.

But, as Forrester Research has noted, "search on many corporate websites is an understaffed, IT-funded afterthought." Search is critical, and it doesn't have to be complex to get right. To help you understand what works, we've created this collection of site search best practices. Follow these, and you'll be on the path to delivering the search experience your customers expect.

Visitors that perform  
a search convert

**5X**

more often than  
those that don't

<http://bit.ly/IT3X3bR>

Visitors that  
search spend

**48%**

more time-on-site than  
those that don't

<http://bit.ly/I07HCQI>

# The Search Box

- #1 Make it prominent
- #2 Make it mobile
- #3 Keep it simple
- #4 Make it smart
- #5 Test it

olives



Advanced Search ▾

Results 1-10 of 914 in 2.94 seconds

Relevancy Date

## 10 Facts about Olive Oil

http://www.ift.org/Knowledge-Center/Learn-About-Food-Science/Food-Facts/10-Facts-about-Olive-Oil.aspx 5/28/2015

Olive oil is known for its health-promoting properties and is very popular amongst consumers ... Olive oil production begins with harvesting the olives; traditionally they were handpicked but ...

Scope Not Defined

## 10 Facts about Olive Oil

http://www.ift.org/Newsroom/News-Releases/2015/May/14/10-Facts-about-Olive-Oil.aspx 5/14/2015

Olive oil production begins with harvesting the olives; traditionally they were handpicked but today ... grades of olive oil included virgin olive oil, refined olive oil and olive pomace oil.

Scope IFT News Releases

Focus Fats/Lipids/Oils

## 20A-9: IDENTIFICATION OF POSSIBLE CAUSES OF OILY PRECIPITATE IN CANNED OLIVE PRODUCTS

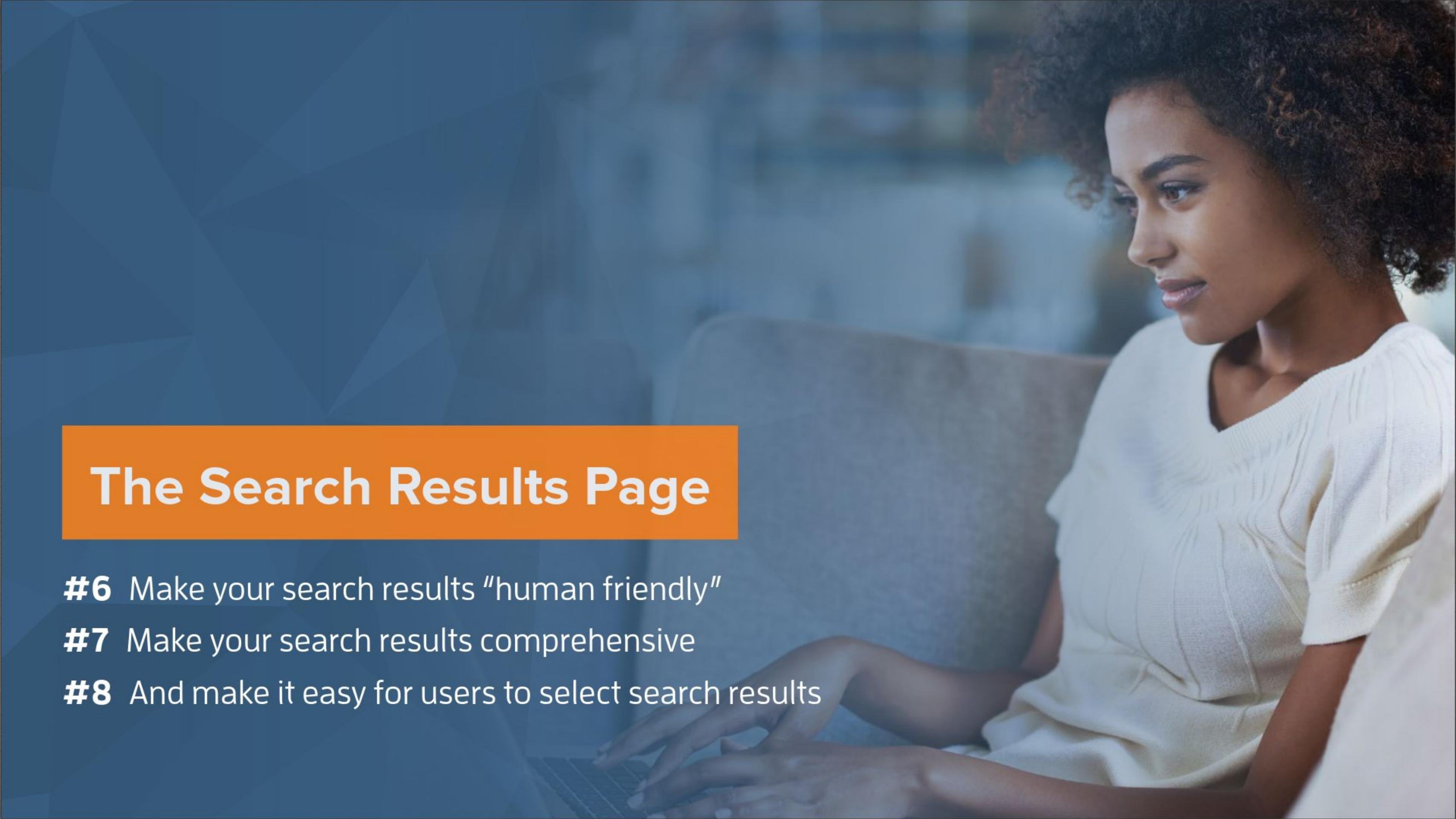
http://www.ift.org/Meetings-and-Events/Past-Meeting-Resources/Technical%20Abstract%20Search%20Details.aspx?id=30398 6/21/1998

Olive oil precipitate or scum, has been observed in some canned olives and is a major problem for commercial canned olive producers. ... XT2 Texture Analyzer.

Scope Technical Abstracts

## Antioxidant capacities and phenolic profiles of table olives from Aegean and Marmara regions of Turkey

http://www.ift.org/Meetings-and-Events/Past-Meeting-Resources/Technical%20Abstract%20Search%20Details.aspx?id=3362 6/6/2009

A woman with curly hair, wearing a white t-shirt, is sitting on a grey couch and looking down at a laptop screen. She appears to be focused on her work. The background is slightly blurred, showing a blue wall and some furniture.

## The Search Results Page

- #6** Make your search results "human friendly"
- #7** Make your search results comprehensive
- #8** And make it easy for users to select search results

## Search-Driven Content Pages

- #9** Use search to drive your listing pages
- #10** Offer facets to filter and navigate search-driven content
- #11** Leverage all the “contextual clues” you can
- #12** Personalize with the Sitecore xDB

## Search Analytics & Reporting

- #13 Track visitor activity
- #14 Slice-and-dice by attribute
- #15 Track every search experience
- #16 Map search activity to KPIs
- #17 Curate your content
- #18 Curate your experience to drive engagement



# Sitecore and Search - History

- Sitecore 7
  - Introduction of the Search Provider module and ContentSearch API
    - Allows to modularize and swap the index engine
  - Lucene used by default for all indexes
    - Configuration files to switch to SOLR are provided out of the box
    - Other Search Providers support are provided by search vendors and the community
  - 3 search indexes: Core, Master, Web
  - Sitecore 7.0 and 7.1
    - Only one Search Provider at a time
  - Sitecore 7.2
    - Multiple Search Providers manage their own set of indexes
  - Sitecore 7.5
    - Introduction of the analytics index

# Sitecore and Search - History

- Sitecore 8.0
  - Introduction of 9 new indexes
    - For marketing, testing, FXM and social
- Sitecore 8.2 Update-1
  - Introduction of the Microsoft Azure Search Provider to replace Lucene when deploying to Microsoft Azure Web Apps
- Sitecore 9
  - Solr is the new default search provider, required for xConnect
  - Lucene can only be used in CMS-only mode

# Sitecore and Search - Index Element

- Configuration
  - Fields
- Locations
  - Crawler
    - Database
    - Root
- Strategies
  - Strategy
    - Database
- Property Store
  - Database



&lt;/&gt; Coveo.SearchProvider.config x

```
454 <indexes hint="list:AddIndex">
455   <index id="Coveo_master_index" type="Coveo.SearchProvider.ProviderIndex, Coveo.SearchProvider" patch:before = "*[1]">
456     <param desc="p_Name">$(id)</param>
457     <configuration ref="coveo/defaultIndexConfiguration">
458       <!-- Whether to push permissions when rebuilding the index. -->
459       <pushPermissionsOnRebuild>true</pushPermissionsOnRebuild>
460       <queryConfiguration ref="coveo/defaultIndexConfiguration/queryConfiguration">
461         <!-- The maximumAge element defines the delay that query results are cached before being refreshed by a new query on the index.
462           -->
463         <maximumAge>00:00:00</maximumAge>
464       </queryConfiguration>
465     </configuration>
466     <locations hint="list:AddCrawler">
467       <crawler name="ContentCrawler" type="Sitecore.ContentSearch.SitecoreItemCrawler, Sitecore.ContentSearch">
468         <database>master</database>
469         <root>/sitecore/content</root>
470         <stopOnError>true</stopOnError>
471       </crawler>
472       <crawler name="MediaLibraryCrawler" type="Sitecore.ContentSearch.SitecoreItemCrawler, Sitecore.ContentSearch">
473         <database>master</database>
474         <root>/sitecore/media library/Files</root>
475         <stopOnError>true</stopOnError>
476       </crawler>
477     </locations>
478     <strategies hint="list:AddStrategy">
479       <strategy ref="contentSearch/indexConfigurations/indexUpdateStrategies/coveoSyncMaster"/>
480     </strategies>
481     <propertyStore ref="coveo/databasePropertyStore" param1="$(id)"/>
482     <enableItemLanguageFallback>false</enableItemLanguageFallback>
483     <enableFieldLanguageFallback>false</enableFieldLanguageFallback>
484   </index>
485   <index id="Coveo_web_index" type="Coveo.SearchProvider.ProviderIndex, Coveo.SearchProvider" patch:before = "*[2]">
486     <param desc="p_Name">$(id)</param>
487     <configuration ref="coveo/defaultIndexConfiguration" />
     <locations hint="list:AddCrawler">
```

&lt;/&gt; Coveo.SearchProvider.config x

```
454     <indexes hint="list:AddIndex">
455         <index id="Coveo_master_index" type="Coveo.SearchProvider.ProviderIndex, Coveo.SearchProvider" patch:before = "*[1]">
456             <param desc="p_Name">$({id})</param>
457             <configuration ref="coveo/defaultIndexConfiguration">
458                 <!-- Whether to push permissions when rebuilding the index. -->
459                 <pushPermissionsOnRebuild>true</pushPermissionsOnRebuild>
460                 <queryConfiguration ref="coveo/defaultIndexConfiguration/queryConfiguration">
461                     <!-- The maximumAge element defines the delay that query results are cached before being refreshed by a new query on the index.
462                         -->
463                         <maximumAge>00:00:00</maximumAge>
464                     </queryConfiguration>
465                 </configuration>
466                 <locations hint="list:AddCrawler">
467                     <crawler name="ContentCrawler" type="Sitecore.ContentSearch.SitecoreItemCrawler, Sitecore.ContentSearch">
468                         <database>master</database>
469                         <root>/sitecore/content</root>
470                         <stopOnError>true</stopOnError>
471                     </crawler>
472                     <crawler name="MediaLibraryCrawler" type="Sitecore.ContentSearch.SitecoreItemCrawler, Sitecore.ContentSearch">
473                         <database>master</database>
474                         <root>/sitecore/media library/Files</root>
475                         <stopOnError>true</stopOnError>
476                     </crawler>
477                 </locations>
478                 <strategies hint="list:AddStrategy">
479                     <strategy ref="contentSearch/indexConfigurations/indexUpdateStrategies/coveoSyncMaster"/>
480                 </strategies>
481                 <propertyStore ref="coveo/databasePropertyStore" param1="$({id})"/>
482                 <enableItemLanguageFallback>false</enableItemLanguageFallback>
483                 <enableFieldLanguageFallback>false</enableFieldLanguageFallback>
484             </index>
485             <index id="Coveo_web_index" type="Coveo.SearchProvider.ProviderIndex, Coveo.SearchProvider" patch:before = "*[2]">
486                 <param desc="p_Name">$({id})</param>
487                 <configuration ref="coveo/defaultIndexConfiguration" />
488                 <locations hint="list:AddCrawler">
```

&lt;/&gt; Coveo.SearchProvider.config x

```
454     <indexes hint="list:AddIndex">
455         <index id="Coveo_master_index" type="Coveo.SearchProvider.ProviderIndex, Coveo.SearchProvider" patch:before = "*[1]">
456             <param desc="p_Name">$({id})</param>
457             <configuration ref="coveo/defaultIndexConfiguration">
458                 <!-- Whether to push permissions when rebuilding the index. -->
459                 <pushPermissionsOnRebuild>true</pushPermissionsOnRebuild>
460                 <queryConfiguration ref="coveo/defaultIndexConfiguration/queryConfiguration">
461                     <!-- The maximumAge element defines the delay that query results are cached before being refreshed by a new query on the index.
462                         -->
463                     <maximumAge>00:00:00</maximumAge>
464                 </queryConfiguration>
465             </configuration>
466             <locations hint="list:AddCrawler">
467                 <crawler name="ContentCrawler" type="Sitecore.ContentSearch.SitecoreItemCrawler, Sitecore.ContentSearch">
468                     <database>master</database>
469                     <root>/sitecore/content</root>
470                     <stopOnError>true</stopOnError>
471                 </crawler>
472                 <crawler name="MediaLibraryCrawler" type="Sitecore.ContentSearch.SitecoreItemCrawler, Sitecore.ContentSearch">
473                     <database>master</database>
474                     <root>/sitecore/media library/Files</root>
475                     <stopOnError>true</stopOnError>
476                 </crawler>
477             </locations>
478             <strategies hint="list:AddStrategy">
479                 <strategy ref="contentSearch/indexConfigurations/indexUpdateStrategies/coveoSyncMaster"/>
480             </strategies>
481             <propertyStore ref="coveo/databasePropertyStore" param1="$({id})"/>
482             <enableItemLanguageFallback>false</enableItemLanguageFallback>
483             <enableFieldLanguageFallback>false</enableFieldLanguageFallback>
484         </index>
485         <index id="Coveo_web_index" type="Coveo.SearchProvider.ProviderIndex, Coveo.SearchProvider" patch:before = "*[2]">
486             <param desc="p_Name">$({id})</param>
487             <configuration ref="coveo/defaultIndexConfiguration" />
             <locations hint="list:AddCrawler">
```

&lt;/&gt; Coveo.SearchProvider.config x

```
454     <indexes hint="list:AddIndex">
455         <index id="Coveo_master_index" type="Coveo.SearchProvider.ProviderIndex, Coveo.SearchProvider" patch:before = "*[1]">
456             <param desc="p_Name">$({id})</param>
457             <configuration ref="coveo/defaultIndexConfiguration">
458                 <!-- Whether to push permissions when rebuilding the index. -->
459                 <pushPermissionsOnRebuild>true</pushPermissionsOnRebuild>
460                 <queryConfiguration ref="coveo/defaultIndexConfiguration/queryConfiguration">
461                     <!-- The maximumAge element defines the delay that query results are cached before being refreshed by a new query on the index.
462                         -->
463                     <maximumAge>00:00:00</maximumAge>
464                 </queryConfiguration>
465             </configuration>
466             <locations hint="list:AddCrawler">
467                 <crawler name="ContentCrawler" type="Sitecore.ContentSearch.SitecoreItemCrawler, Sitecore.ContentSearch">
468                     <database>master</database>
469                     <root>/sitecore/content</root>
470                     <stopOnError>true</stopOnError>
471                 </crawler>
472                 <crawler name="MediaLibraryCrawler" type="Sitecore.ContentSearch.SitecoreItemCrawler, Sitecore.ContentSearch">
473                     <database>master</database>
474                     <root>/sitecore/media library/Files</root>
475                     <stopOnError>true</stopOnError>
476                 </crawler>
477             </locations>
478             <strategies hint="list:AddStrategy">
479                 <strategy ref="contentSearch/indexConfigurations/indexUpdateStrategies/coveoSyncMaster"/>
480             </strategies>
481             <propertyStore ref="coveo/databasePropertyStore" param1="$({id})"/>
482             <enableItemLanguageFallback>false</enableItemLanguageFallback>
483             <enableFieldLanguageFallback>false</enableFieldLanguageFallback>
484         </index>
485         <index id="Coveo_web_index" type="Coveo.SearchProvider.ProviderIndex, Coveo.SearchProvider" patch:before = "*[2]">
486             <param desc="p_Name">$({id})</param>
487             <configuration ref="coveo/defaultIndexConfiguration" />
488             <locations hint="list:AddCrawler">
```

&lt;/&gt; Coveo.SearchProvider.config x

```
454 <indexes hint="list:AddIndex">
455   <index id="Coveo_master_index" type="Coveo.SearchProvider.ProviderIndex, Coveo.SearchProvider" patch:before = "*[1]">
456     <param desc="p_Name">$(id)</param>
457     <configuration ref="coveo/defaultIndexConfiguration">
458       <!-- Whether to push permissions when rebuilding the index. -->
459       <pushPermissionsOnRebuild>true</pushPermissionsOnRebuild>
460       <queryConfiguration ref="coveo/defaultIndexConfiguration/queryConfiguration">
461         <!-- The maximumAge element defines the delay that query results are cached before being refreshed by a new query on the index.
462           -->
463         <maximumAge>00:00:00</maximumAge>
464       </queryConfiguration>
465     </configuration>
466     <locations hint="list:AddCrawler">
467       <crawler name="ContentCrawler" type="Sitecore.ContentSearch.SitecoreItemCrawler, Sitecore.ContentSearch">
468         <database>master</database>
469         <root>/sitecore/content</root>
470         <stopOnError>true</stopOnError>
471       </crawler>
472       <crawler name="MediaLibraryCrawler" type="Sitecore.ContentSearch.SitecoreItemCrawler, Sitecore.ContentSearch">
473         <database>master</database>
474         <root>/sitecore/media library/Files</root>
475         <stopOnError>true</stopOnError>
476       </crawler>
477     </locations>
478     <strategies hint="list:AddStrategy">
479       <strategy ref="contentSearch/indexConfigurations/indexUpdateStrategies/coveoSyncMaster"/>
480     </strategies>
481     <propertyStore ref="coveo/databasePropertyStore" param1="$(id)"/>
482     <enableItemLanguageFallback>false</enableItemLanguageFallback>
483     <enableFieldLanguageFallback>false</enableFieldLanguageFallback>
484   </index>
485   <index id="Coveo_web_index" type="Coveo.SearchProvider.ProviderIndex, Coveo.SearchProvider" patch:before = "*[2]">
486     <param desc="p_Name">$(id)</param>
487     <configuration ref="coveo/defaultIndexConfiguration" />
488     <locations hint="list:AddCrawler">
```

## &lt;/&gt; Coveo.SearchProvider.config x

```
454     <indexes hint="list:AddIndex">
455         <index id="Coveo_master_index" type="Coveo.SearchProvider.ProviderIndex, Coveo.SearchProvider" patch:before = "*[1]">
456             <param desc="p_Name">$({id})</param>
457             <configuration ref="coveo/defaultIndexConfiguration">...
458             </configuration>
459             <locations hint="list:AddCrawler">...
460             </locations>
461             <strategies hint="list:AddStrategy">
462                 <strategy ref="contentSearch/indexConfigurations/indexUpdateStrategies/coveoSyncMaster"/>
463             </strategies>
464             <propertyStore ref="coveo/databasePropertyStore" param1="$({id})"/>
465             <enableItemLanguageFallback>false</enableItemLanguageFallback>
466             <enableFieldLanguageFallback>false</enableFieldLanguageFallback>
467         </index>
468         <index id="Coveo_web_index" type="Coveo.SearchProvider.ProviderIndex, Coveo.SearchProvider" patch:before = "*[2]">
469             <param desc="p_Name">$({id})</param>
470             <configuration ref="coveo/defaultIndexConfiguration" />
471             <locations hint="list:AddCrawler">...
472             </locations>
473             <strategies hint="list:AddStrategy">
474                 <strategy ref="contentSearch/indexConfigurations/indexUpdateStrategies/coveoOnPublishEndAsync"/>
475             </strategies>
476             <propertyStore ref="coveo/databasePropertyStore" param1="$({id})"/>
477             <enableItemLanguageFallback>false</enableItemLanguageFallback>
478             <enableFieldLanguageFallback>false</enableFieldLanguageFallback>
479         </index>
480     </indexes>
481     </configuration>
482     </contentSearch>
483     <scheduling>...
484     </scheduling>
485     <settings>...
486     </settings>
487     <sites>...
488     </sites>
489 
```

# Coveo for Sitecore - What Is It?

- A solution to build great search pages for a website visitors
- A Sitecore package
- Separate Sitecore indexes: Coveo\_master\_index, Coveo\_web\_index
  - Only index needed fields
  - Index HTML version of the items
- Index location options:
  - Search as a Service in a Coveo Cloud organization
  - On-Premises in a Coveo Enterprise Search (CES) index
- Sitecore UI components to create search pages, search-driven lists and add search boxes
  - Uses Coveo JavaScript Search Framework, a client-side search UI
- Supports Sitecore personalization, xDB, profiles, and much more

# Coveo for Sitecore - What It Isn't?

- Not meant to replace Lucene/SOLR for out of the box Sitecore indexes
- Not meant for indexes unrelated to visitor search functionality
  - Not for the new Sitecore 8 indexes
  - Not a centralized index solution for the Sitecore analytics index

# Solutions History



Available



Recommended

Solution	Description	Sitecore Version						
		8.X	7.5	7.2	7.0 - 7.1	6.6	6.0 - 6.5	5.3
<b>Indexing</b>								
Coveo Cloud V2	Latest Search-as-a-Service offering. Actively developed.	thumb up	thumb up	thumb up				
Coveo Enterprise Search 7.0	Latest on-premises index. Also referred as CES. No new features developed.	checkmark	checkmark	checkmark	checkmark	checkmark	checkmark	checkmark
<b>Crawling</b>								
Sitecore Connector	2nd Sitecore solution. Also referred as Sitecore2. Soon to be deprecated. Not recommended on Sitecore 7.X.			checkmark	checkmark	thumb up	thumb up	
Sitecore Legacy Connector	1st Sitecore solution. Now deprecated.						checkmark	thumb up
Web Pages Connector	Not recommended.	checkmark	checkmark	checkmark	checkmark	checkmark	checkmark	checkmark

# Solutions History



Available



Recommended

Solution	Description	Sitecore Version				
		8.2	8.0 - 8.1	7.2 - 7.5	7.0 - 7.1	6.X
<b>All-In-One Solutions - Recommended</b>						
Coveo for Sitecore 4.1	<p>Same as Coveo for Sitecore 4.0 and fully backwards compatible.</p> <p>New Coveo for Sitecore Hive modular UI framework.</p> <p>Embeds:</p> <ul style="list-style-type: none"> <li>• Coveo JavaScript Search Framework 1.0 (legacy UI) and 2.0 (Hive)</li> </ul>					
Coveo for Sitecore 4.0	<p>Same as Coveo for Sitecore 3.0 for on-premises.</p> <p>Can also index in Cloud using:</p> <ul style="list-style-type: none"> <li>• Coveo Cloud Platform V2 (and Push API)</li> </ul> <p>Embeds:</p> <ul style="list-style-type: none"> <li>• Coveo JavaScript Search Framework 1.0</li> </ul>					
Coveo for Sitecore 3.0	<p>Requires:</p> <ul style="list-style-type: none"> <li>• Coveo Search API 8.0</li> <li>• Coveo Enterprise Search 7.0 (and Queue crawler)</li> </ul> <p>Embeds:</p> <ul style="list-style-type: none"> <li>• Coveo JavaScript Search Framework 0.9</li> </ul>					

# Solutions History



Available



Recommended

		Sitecore Version			
Solution	Description	8.X	7.X	6.X	5.3
<b>Search Interface</b>					
Coveo JavaScript Search Framework	Also referred as JS UI. Latest search interface. Works with any backend. Actively developed.	thumb up	thumb up	checkmark	checkmark
Coveo .Net Front-End 12.0	Old search interface. Works only with ASP.Net websites. No new features developed. Not recommended for new implementations.	checkmark	checkmark	checkmark	checkmark
<b>Search API</b>					
Coveo Search API 8.0	On-premises bridge between JS UI and Coveo Search Service. The only search API available in Coveo Cloud Platform.	thumb up	thumb up	checkmark	checkmark
Coveo Search Service	SOAP service included in Coveo Enterprise Search 7.0. Used by on-premises Coveo Search API.	checkmark	checkmark	checkmark	checkmark

# Supported Sitecore Versions

December 2017 release (4.1.184.19) of Coveo for Sitecore supports all Sitecore versions from 7.2 to 8.2 Update-5 (rev. 170728)

An [issue](#) with Sitecore prevents the Experience Editor to save a page when a component field defines rules in versions:

- 8.0 Update-6 and Update-7
- 8.1 Update-1
- 8.2 (all revisions)

This issue affects only the legacy UI components. Not the Coveo for Sitecore Hive framework UI components.

# Editions

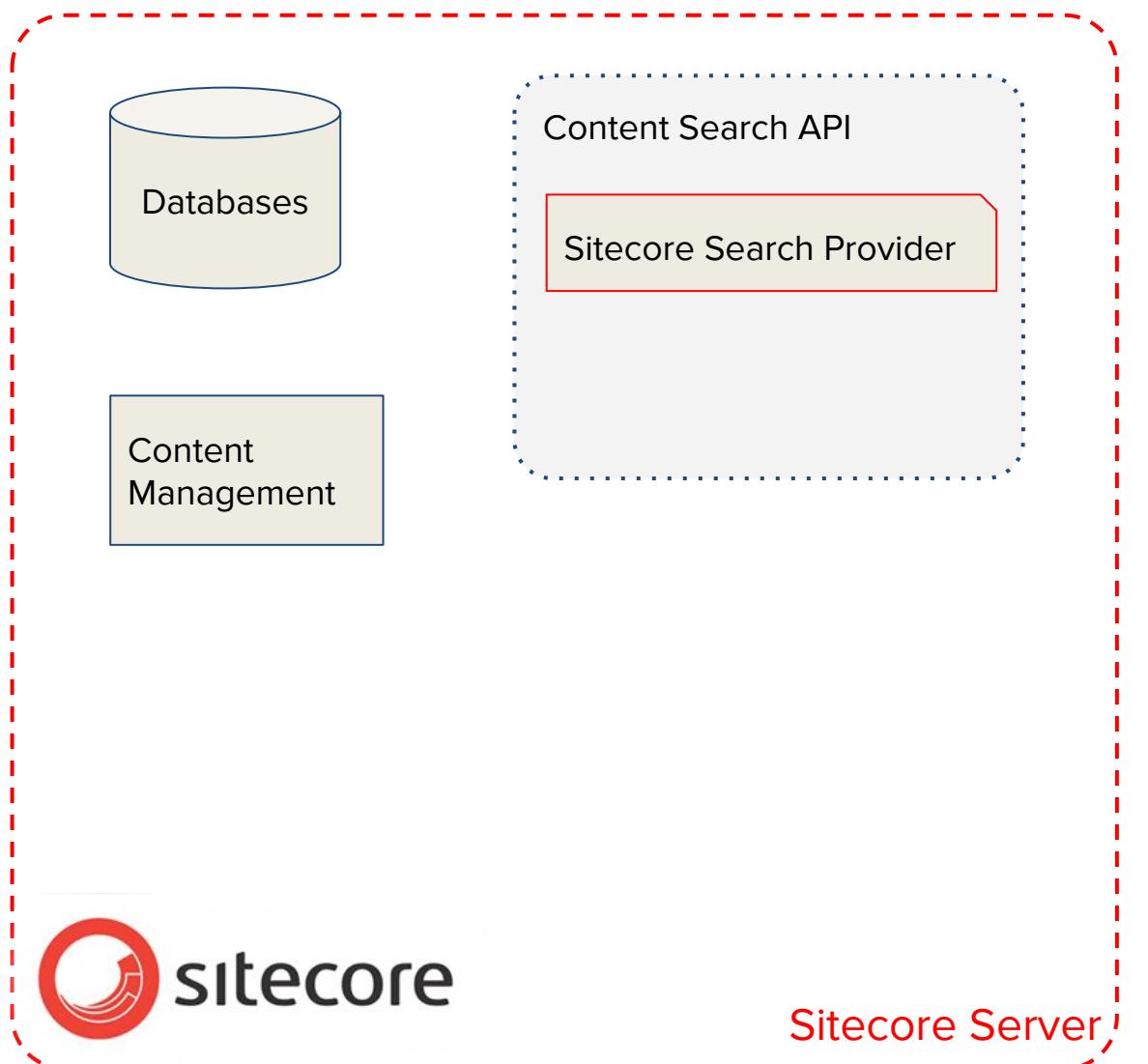
Coveo for Sitecore 4 gives you the opportunity to choose between two index hosting options and 3 editions:

- Cloud
  - Pro Cloud
  - Enterprise Cloud (Trial available)
- On-Premises
  - Free

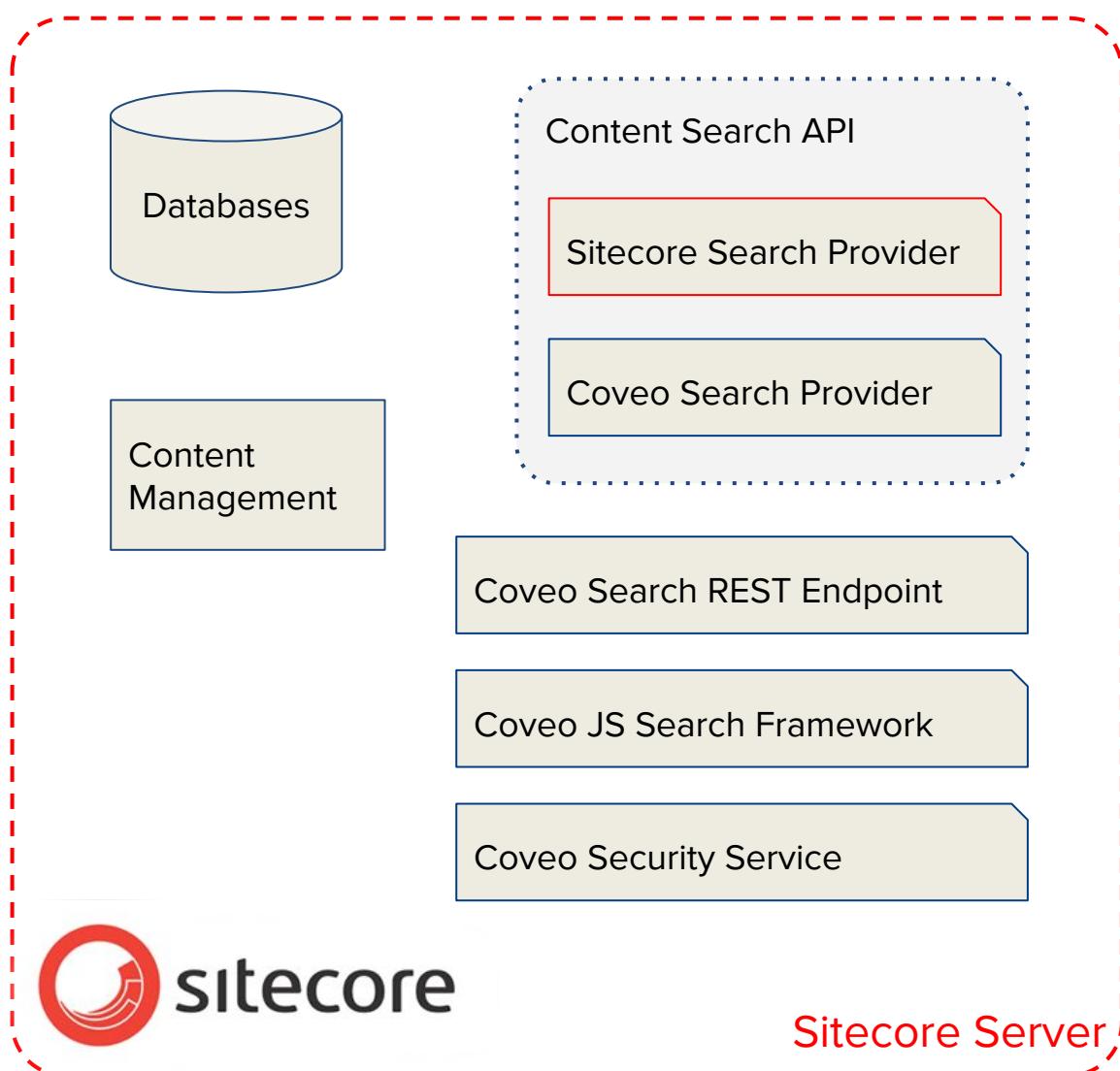
The two paid on-premises editions are not sold anymore:

- Pro
- Enterprise

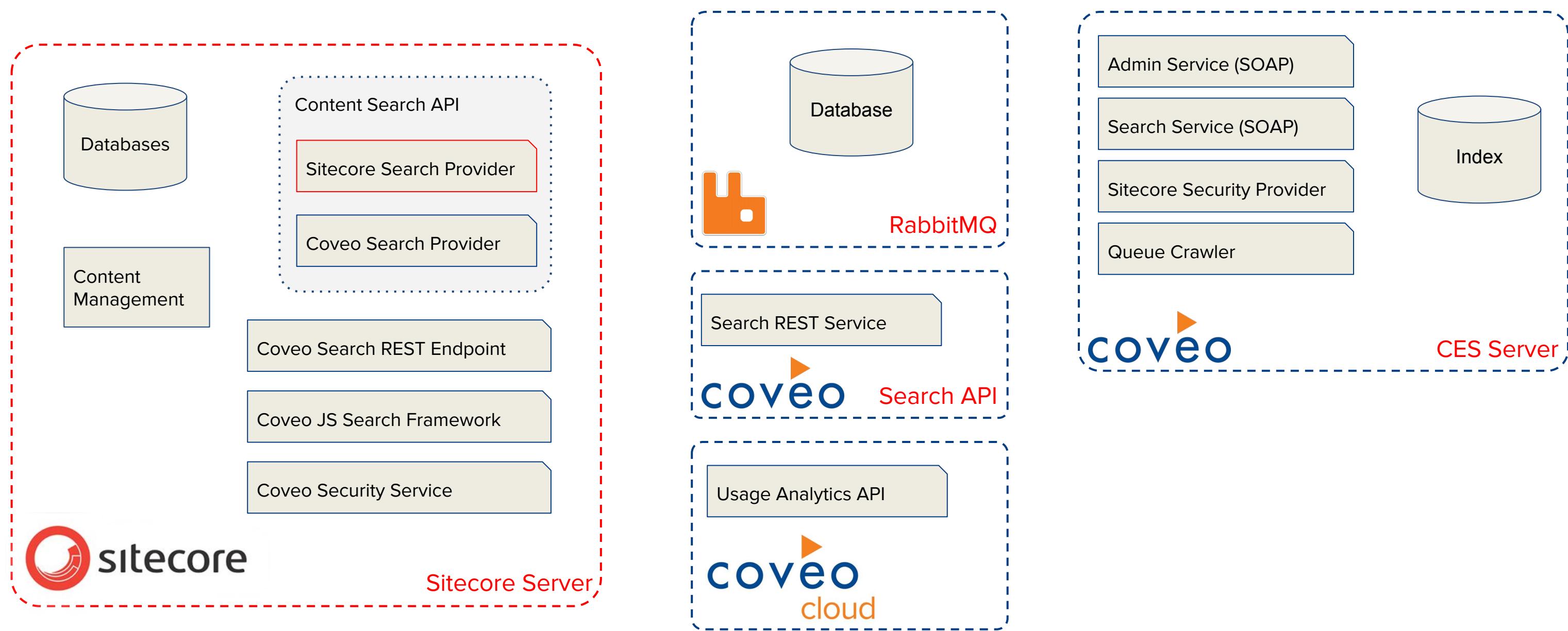
# Architecture Overview - Sitecore



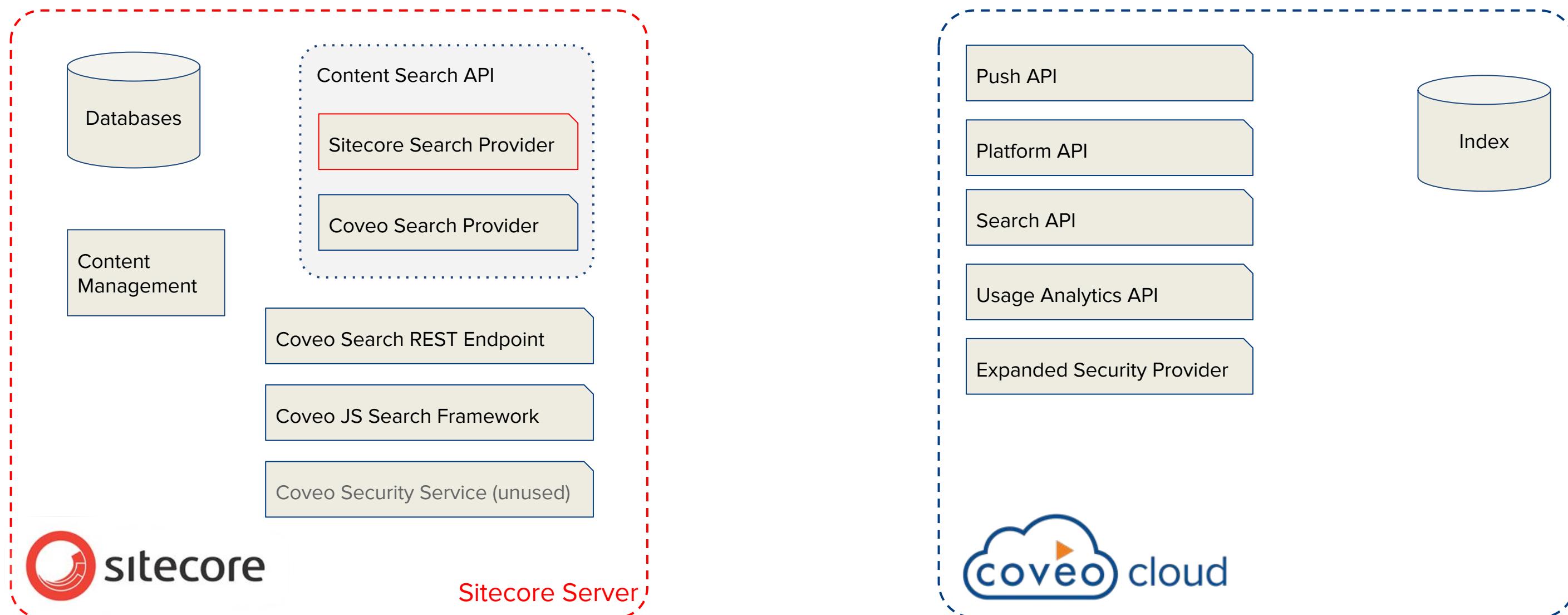
# Architecture Overview - Coveo for Sitecore



# Architecture Overview - On-Premises



# Architecture Overview - Cloud





Coveo Cloud

# Validating Coveo for Sitecore Using the Diagnostic Page

1. Open the Diagnostic Page
  - a. Sitecore Control Panel > Coveo Search > Diagnostic page
2. Validate that all the components are working correctly

**Coveo for Sitecore components state**

[Download Diagnostic Package](#) To see if any state has changed: [Refresh](#)

Component Name	State
Cloud Service	<span style="color: green;">● Up and running</span>
Cloud Service Organization	<span style="color: green;">● Up and running</span>
Cloud Service Sources Endpoint	<span style="color: green;">● Up and running</span>
Coveo for Sitecore is enabled	<span style="color: green;">● Up and running</span>
Coveo Index Crawlers	<span style="color: green;">● Up and running</span>
Coveo Pipeline Names Configuration	<span style="color: green;">● Up and running</span>
Coveo Search REST Endpoint	<span style="color: green;">● Up and running</span>
Coveo Security Service	<span style="color: green;">● Up and running</span>
Sitecore's Configuration	<span style="color: green;">● Up and running</span>

**Coveo for Sitecore version information**

Current Coveo for Sitecore version: 4.0.1258.0

Current Sitecore version: 8.0.150812

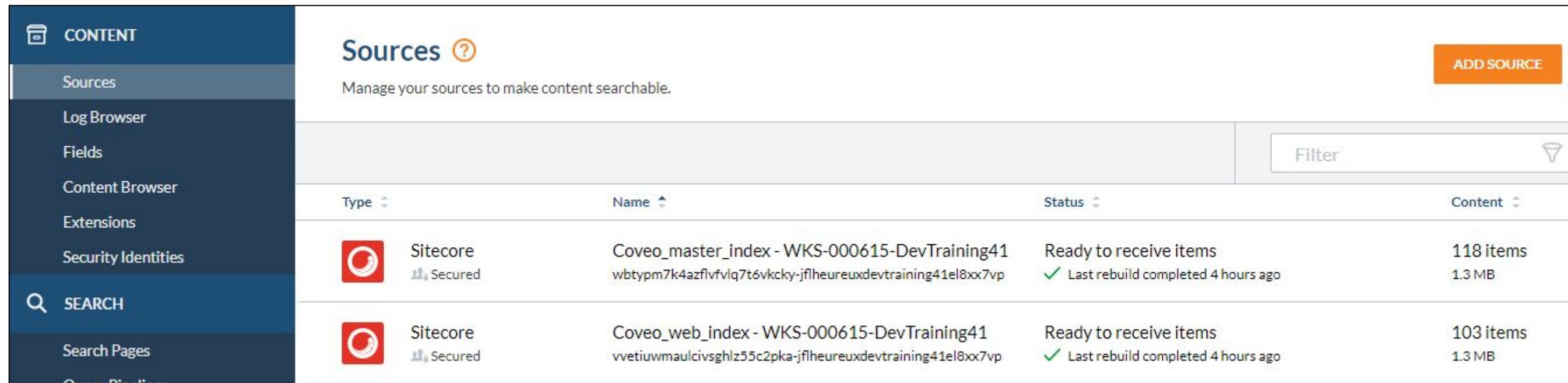
Compatibility status: these versions are compatible

**COVEO SEARCH**

- [Actions](#)
- [Cloud admin UI](#)
- [Configuration](#)
- [Diagnostic page](#)
- [Developer & support resources](#)
- [License details](#)

# Validating the Number of Indexed Documents

1. Log into the Coveo Cloud Administration Console
2. Navigate to the Sources section (Content > Sources)
3. You should have 118 documents in the master source and 103 in the web source
  - a. The difference are the 3 draft items and 12 non-English items in Master



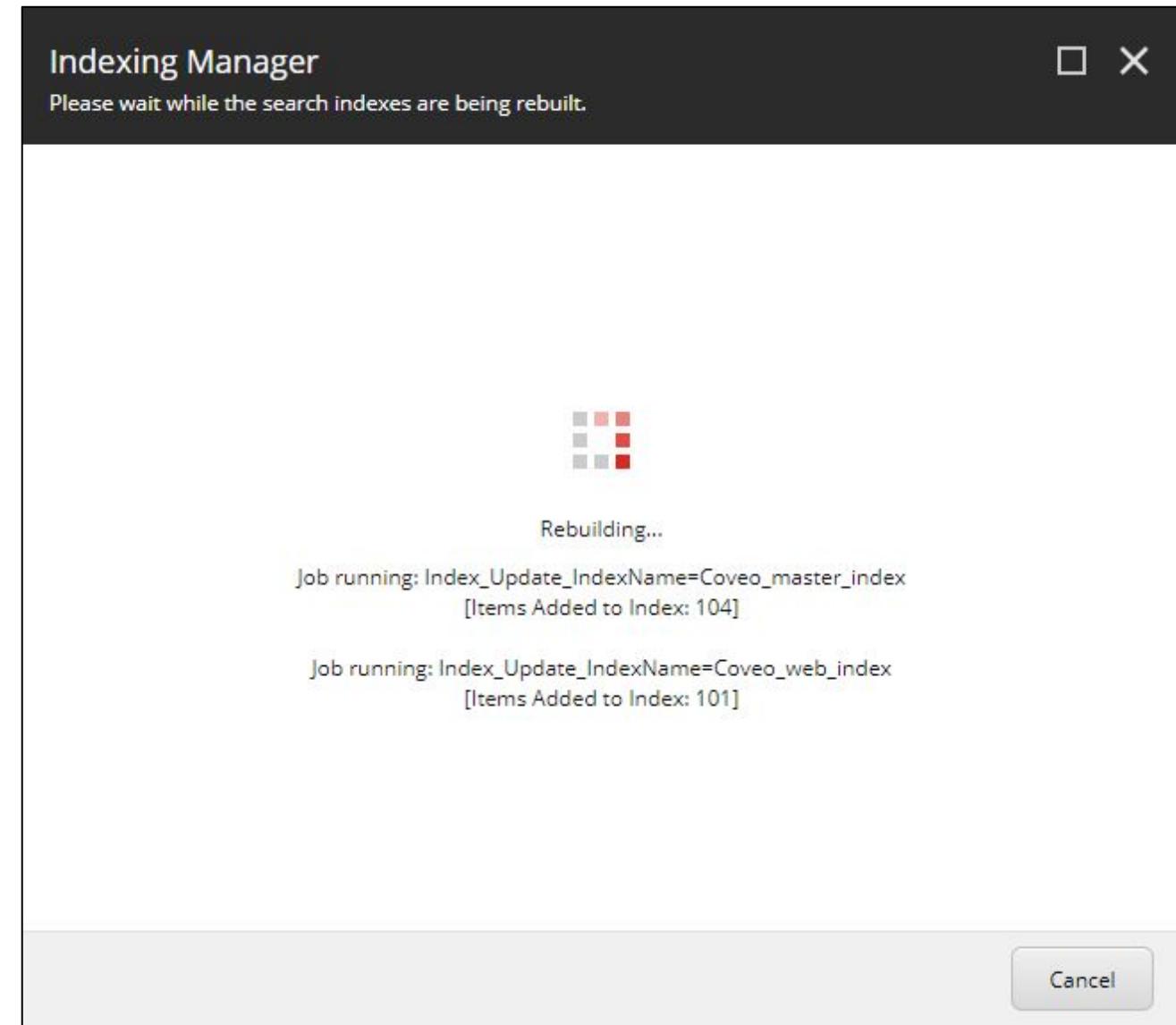
The screenshot shows the 'Sources' page in the Coveo Cloud Administration Console. The left sidebar has 'CONTENT' and 'SEARCH' sections with links for 'Sources', 'Log Browser', 'Fields', 'Content Browser', 'Extensions', and 'Security Identities'. The main area is titled 'Sources' with a help icon. It says 'Manage your sources to make content searchable.' and features an 'ADD SOURCE' button. A 'Filter' button with a funnel icon is also present. The table lists two sources: 'Coveo\_master\_index - WKS-000615-DevTraining41' and 'Coveo\_web\_index - WKS-000615-DevTraining41'. Both are Sitecore sources, secured, and ready to receive items. The last rebuild was completed 4 hours ago. The 'Coveo\_master\_index' has 118 items (1.3 MB) and the 'Coveo\_web\_index' has 103 items (1.3 MB).

Type	Name	Status	Content
Sitecore Secured	Coveo_master_index - WKS-000615-DevTraining41 wbtypm7k4azflvfvqlq7t6vkcky-jfheureuxdevtraining41el8xx7vp	Ready to receive items ✓ Last rebuild completed 4 hours ago	118 items 1.3 MB
Sitecore Secured	Coveo_web_index - WKS-000615-DevTraining41 vvetiuwmaulcivsghlz55c2pka-jfheureuxdevtraining41el8xx7vp	Ready to receive items ✓ Last rebuild completed 4 hours ago	103 items 1.3 MB

# Why Is the Indexed Item Count Different in Sitecore?

In Sitecore, the Indexing Manager will display more indexed items than what is shown in the Coveo Cloud Content Browser.

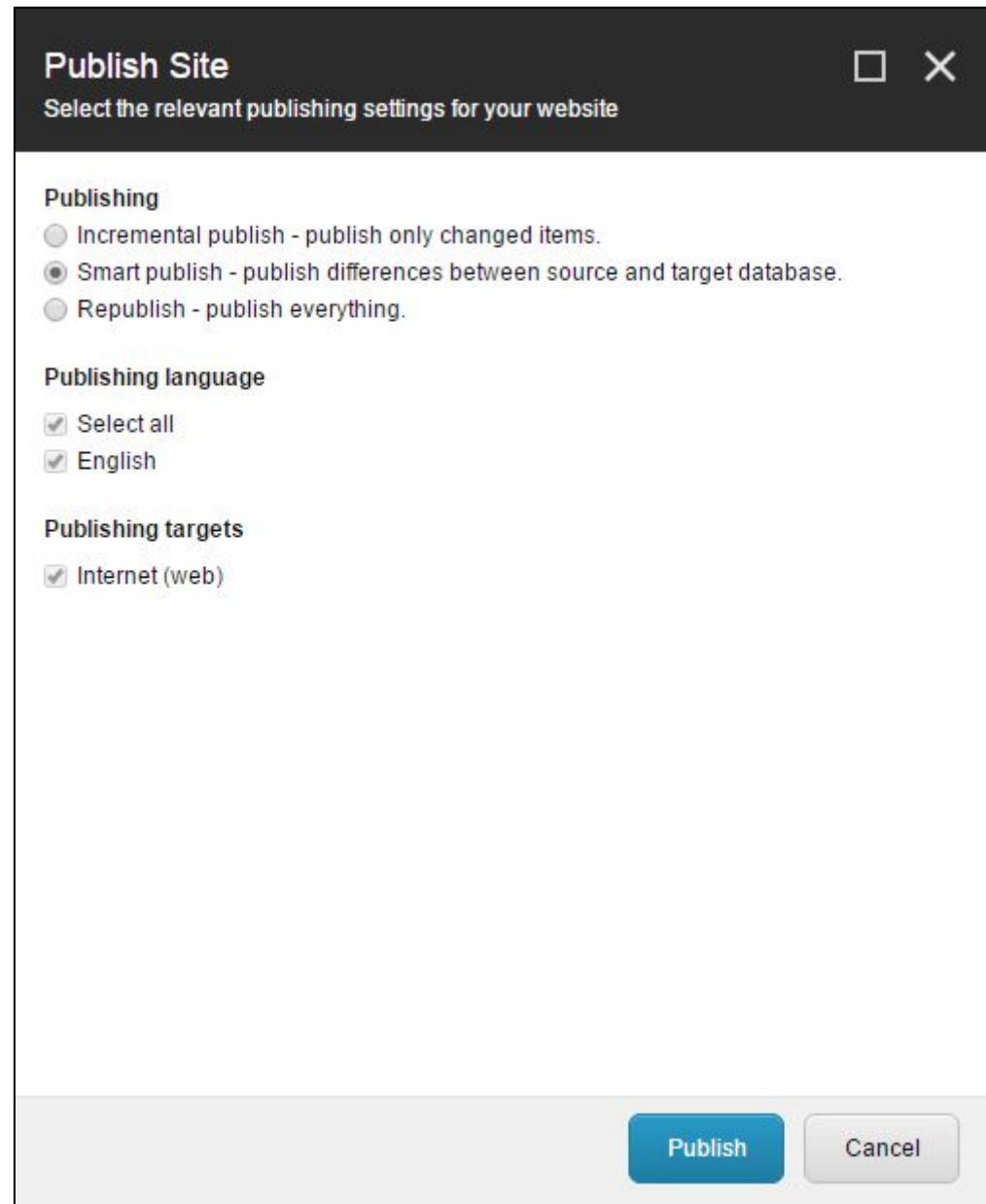
This is due to inbound filters in the Sitecore indexing pipeline.



# Publishing the Site

Coveo for Sitecore added items in the master database but not in the web database. To be able to use the UI components in the published website, we need to publish those items.

1. Open the Sitecore Desktop
  - a. Launchpad > Desktop
2. Open the start menu
3. Click Publish Site item
4. Choose Smart publish
5. Click Publish button

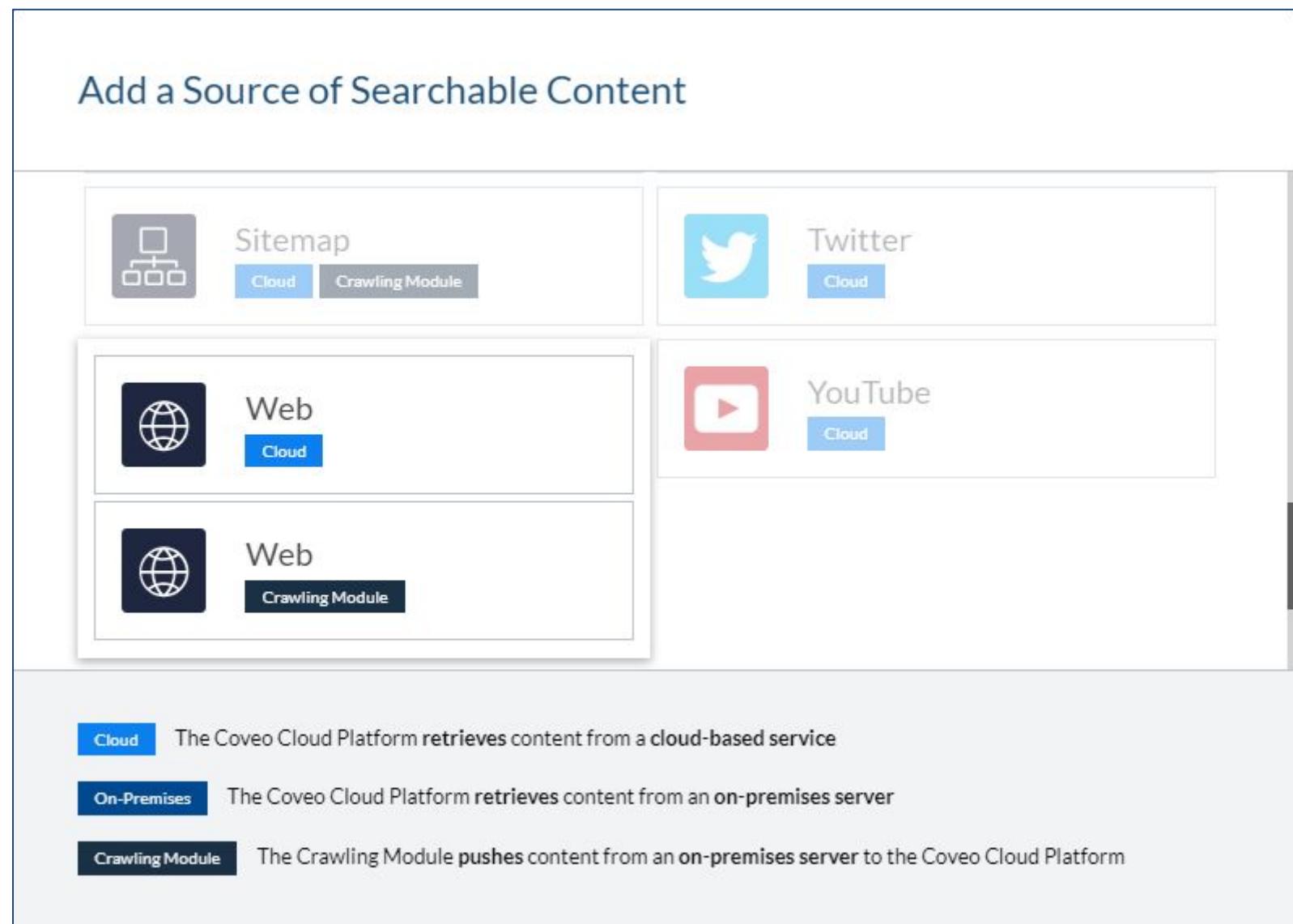


# Lab

1. Validate the state of Coveo for Sitecore using the Coveo Diagnostic Page
2. Validate the number of indexed documents using the Coveo Cloud Administration Console
3. Smart publish your site

# Creating a Web Source (1/3)

1. From the Coveo Cloud Platform Administration Console, open the Sources section (Content > Sources)
2. Click the "Add Source" button in the section header
3. In the "Add a Source of Searchable Content" dialog
  - a. Select "Web" as the source type
  - b. Select "Web (Cloud)" as the source flavor



# Creating a Web Source (2/3)

## 4. In the "Add a Web source" dialog

- a. Source name: Enter "Helix"
- b. Site URL: Enter "https://helix.sitecore.net/"
- c. Security: Ensure "Shared" is selected
- d. Click Add Source button

Add a Web Source ?

Coveo Cloud Platform will pull content from a given website.

CONFIGURATION	CONFIGURATION ACCESS	AUTHENTICATION
Source name Helix		
Site URL https://helix.sitecore.net/		
	<span style="color: orange;">+</span>	
User agent <span style="color: orange;">i</span> Mozilla/5.0 AppleWebKit/537.36		
Content Security	<span style="background-color: #004a89; color: white; padding: 2px 10px;">SHARED</span> <span style="border: 1px solid #ccc; padding: 2px 10px;">PRIVATE</span>	
<span style="background-color: orange; color: white; padding: 5px 20px; border-radius: 5px;">ADD AND BUILD SOURCE</span> <span style="border: 1px solid #ccc; padding: 2px 10px;">ADD SOURCE</span> <span style="border: 1px solid #ccc; padding: 2px 10px;">CANCEL</span>		

# Creating a Web Source (3/3)

- 5.** In the list of sources
  - a. Select the "Helix" source
  - b. Click the "More" button in the section header and choose Manage schedules...
- 6.** In the "Schedule" dialog
  - a. Disable the Rescan schedule by clicking the checkbox
  - b. Choose Update Schedule
- 7.** In the list of sources
  - a. Select the "Helix" source
  - b. Click the "Rebuild" button in the section header
  - c. Click the "Activity" button in the section header
- 8.** Wait for the provisioning wait, addition, schedule deletion, and initial build

Once the rebuild is completed, you should have about 77 documents

# Index Transactions and Commit

- The indexing operation can be monitored from the Sources or Activity Browser sections
- The indexed documents are added to a transaction, a temporary location where documents wait to be added to the index in batches
- Every minute, the pending transactions are committed to the index, then various caches are refreshed
- Documents are now searchable

# Lab

## 1. Index Habitat website

# Coveo Cloud Administration Console - Demo

- Content
  - Sources
  - Log Browser
  - Fields
  - Security Identities
- Organization
  - Groups
  - Members
  - API Keys
  - Activity Browser
- Settings

# Exploring Indexed Documents

From the Coveo Cloud Platform Administration Console, open the Content Browser section (Content > Content Browser)

- Source and other document attributes can be selected on the left side facets to filter documents
- Results can be sorted by modified or indexed date
- A result fields, permissions, and quick view can be viewed by:
  - Clicking the Properties button in the results list header
  - Double-clicking on the result (not on a link)
- (...) > Preferences > "View all content" checkbox is useful as an admin to see all the documents, no matter if you really have access to them or not

# Lab

1. Validate the number of indexed documents in the Habitat source
2. Explore the indexed documents

# Inviting a Member to a Group

1. From the Coveo Cloud Platform Administration Console, open the Groups section (Organization > Groups)
2. Double-click on the "Administrators" group
3. In the "Edit a Group" dialog
  - a. Navigate to the "Members" tab
  - b. Click the "Invite Member" button
  - c. In the "Provider" field, select "Any listed"
  - d. In the "Email" field, enter the email address your instructor will give you
  - e. Click the "Invite Member" button under the "Email" field
  - f. Click "Save"

# Lab

1. Invite your instructor to the "Administrators" group of your Coveo Cloud organization

# Managing Coveo Cloud from Sitecore

The following Coveo Cloud elements are managed automatically from Sitecore:

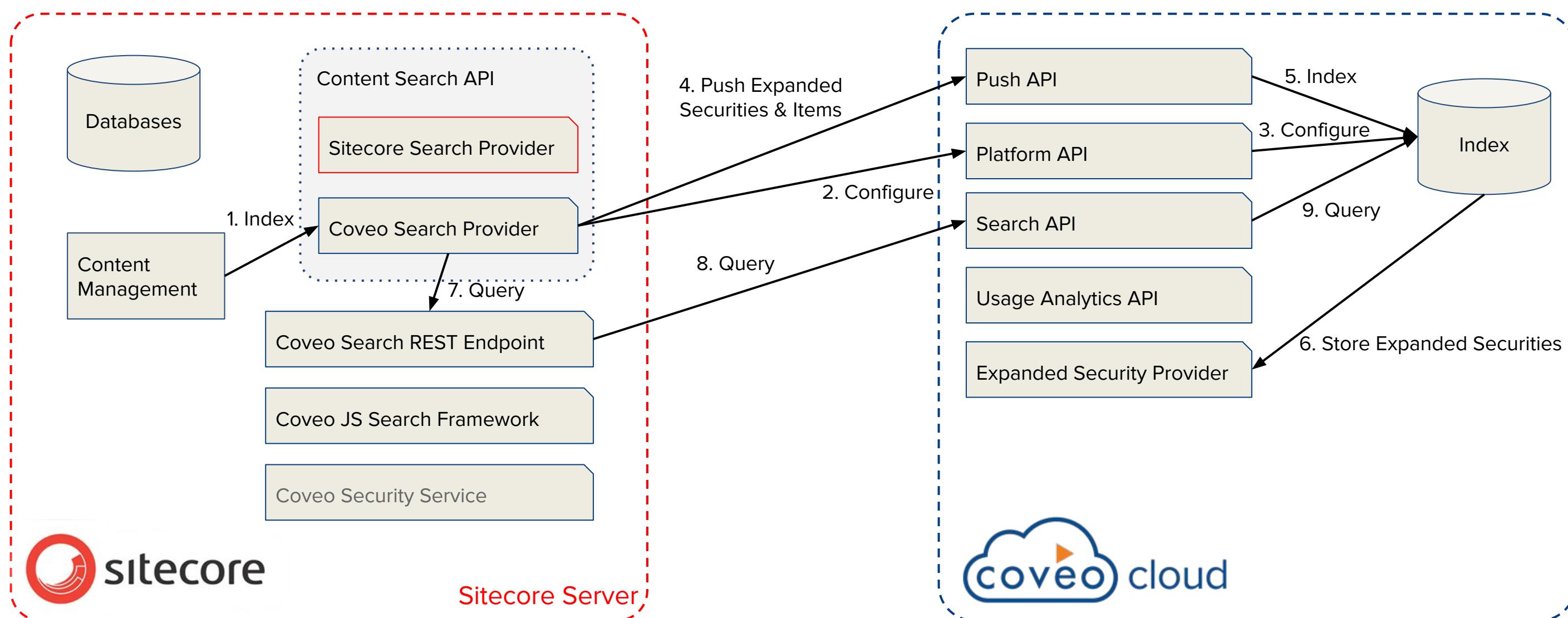
- Organisation creation
- Sources
- Expanded Security Provider
- Fields

Modifying a Sitecore source, field or security provider from the Coveo Cloud Administration Console is not recommended since the change will be automatically overwritten at the next Sitecore indexing operation

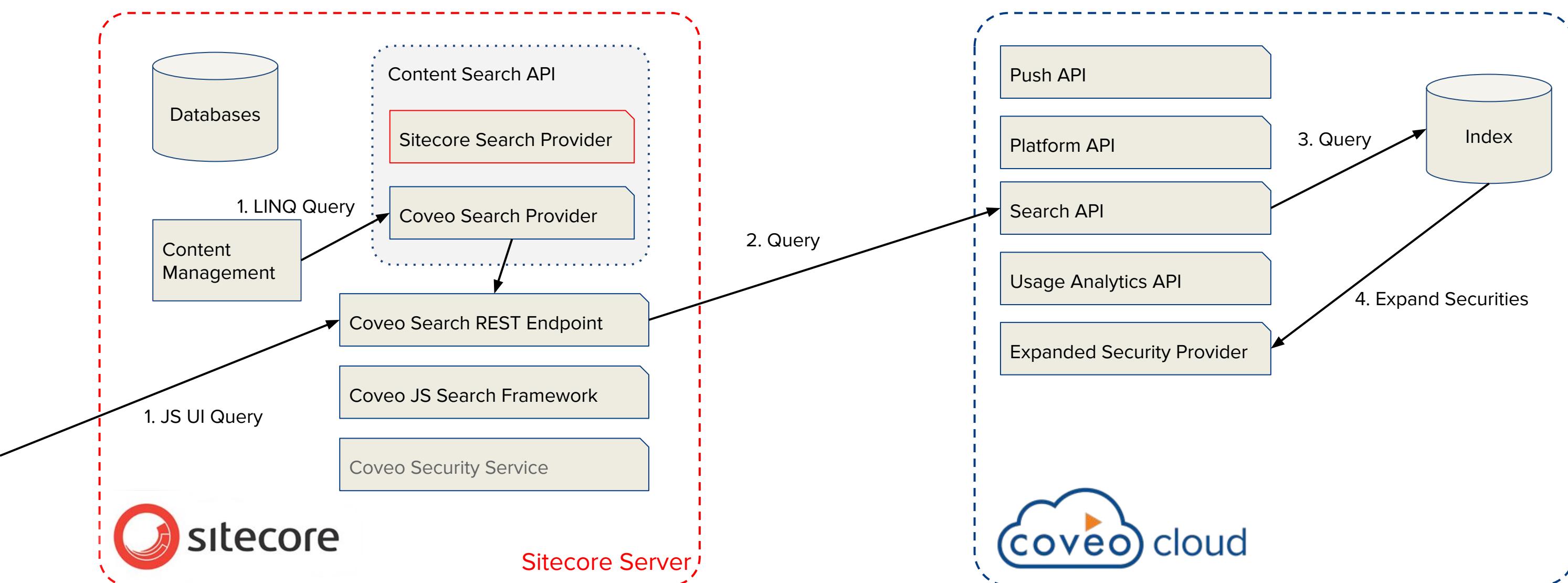
# Coveo Cloud Endpoints

- Administration Console: <https://platform.cloud.coveo.com>
  - Documentation: [https://onlinehelp.coveo.com/en/cloud/coveo\\_cloud\\_help.htm](https://onlinehelp.coveo.com/en/cloud/coveo_cloud_help.htm)
- Platform API: <https://platform.cloud.coveo.com>
  - API Documentation: <https://platform.cloud.coveo.com/docs>
- Usage Analytics API: <https://usageanalytics.coveo.com>
  - API Documentation:
    - <https://usageanalytics.coveo.com/docs/read/>
    - <https://usageanalytics.coveo.com/docs/write/>
- Push API: <https://push.cloud.coveo.com>
  - Documentation
    - API: <https://platform.cloud.coveo.com/docs>
    - Usage: <https://developers.coveo.com/display/CloudPlatform/Push+API+Usage+Overview>

# Architecture Overview - Indexing Sitecore Items



# Architecture Overview - Performing Search Queries





# Configuration Files

# Configuration Files

- \Website\App\_Config\Include
  - \Coveo
    - {■ Coveo.CloudPlatformClient.config  
■ Coveo.CloudPlatformClient.Custom.config }
    - {■ **Coveo.SearchProvider.config**  
■ **Coveo.SearchProvider.Custom.config**}
    - Coveo.SearchProvider.Licensing.config
    - {■ **Coveo.SearchProvider.Rest.config**  
■ **Coveo.SearchProvider.Rest.Custom.config**}
    - Coveo.UI.Components.ExperienceEditor.config
    - Coveo.UI.Controls.config
    - Coveo.UI.Controls.Specific.config
    - SwitchMasterToWeb.Coveo.config.example
  - \ExperienceAnalyticsCoveo
    - Coveo.Analytics.xDB.config
  - \ExperienceProfileCoveo
    - Coveo.ExperienceProfile.config

# Coveo.SearchProvider.config

- events
- pipelines
- coveo
  - defaultIndexConfiguration
    - queryConfiguration, fieldMap, fields (Computed Index Fields), excluded templates...
- contentSearch
  - configuration
    - Indexes
      - Coveo\_master\_index, Coveo\_web\_index
- sites
- scheduling, settings, factories, and retryGroups

Coveo.SearchProvider.config - Visual Studio Code [Administrator]

File Edit Selection View Go Debug Tasks Help

⤵ Coveo.SearchProvider.config ✎

```
1 <!--
2 Do not modify this configuration file.
3 Instead, you should patch this file using Coveo.SearchProvider.Custom.config.
4 See Coveo.SearchProvider.Custom.config for additional description of the configuration elements.
5 -->
6 <configuration xmlns:patch="http://www.sitecore.net/xmlconfig/">
7   <sitecore>
8     <events timingLevel="custom">...
67   </events>
68   <factories>...
77   </factories>
78   <pipelines>...
208   </pipelines>
209   <coveo>...
435   </coveo>
436   <contentSearch>...
508   </contentSearch>
509   <scheduling>...
513   </scheduling>
514   <settings>...
516   </settings>
517   <sites>...
524   </sites>
525 </sitecore>
526 </configuration>
```

⤵ Coveo.SearchProvider.config

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF XML ☺

## Coveo.SearchProvider.config ✘

```
8     <events timingLevel="custom">
9         +     <event name="coveo:config:cloudConnectionChanged"> ...
11        </event>
12        +     <event name="coveo:indexing:afterSetupRequirements"> ...
14        </event>
15        +     <event name="item:created"> ...
17        </event>
18        +     <event name="item:templateChanged"> ...
20        </event>
21        +     <event name="item:deleting"> ...
23        </event>
24        +     <event name="item:moved"> ...
26        </event>
27        +     <event name="item:saved"> ...
30        </event>
31        +     <event name="user:created"> ...
34        </event>
35        +     <event name="user:updated"> ...
38        </event>
39        +     <event name="user:deleted"> ...
42        </event>
43        +     <event name="role:created"> ...
46        </event>
47        +     <event name="role:deleted"> ...
50        </event>
51        +     <event name="roles:usersAdded"> ...
54        </event>
55        +     <event name="roles:usersRemoved"> ...
58        </events>
```

Coveo.SearchProvider.config - Visual Studio Code [Administrator]

File Edit Selection View Go Debug Tasks Help

⤵ Coveo.SearchProvider.config ✎

```
<pipelines>
  <getContentEditorWarnings> ...
  </getContentEditorWarnings>
  <getPageEditorNotifications> ...
  </getPageEditorNotifications>
  <startAnalytics> ...
  </startAnalytics>
  <!-- Pipeline receiving an item before it is processed for indexing -->
  <coveoItemProcessingPipeline>
  </coveoItemProcessingPipeline>
  <!-- Pipeline receiving an item after being processed, just before it is send to be indexed -->
  <coveoPostItemProcessingPipeline> ...
  </coveoPostItemProcessingPipeline>
  <!-- Pipeline to convert the fields from Sitecore to a format handled by Coveo -->
  <coveoFieldConverterPipeline> ...
  </coveoFieldConverterPipeline>
  <!-- Sitecore pipeline to get the index name corresponding to the currently selected item or context database.
       This pipeline can return search indexes from any provider. -->
  <contentSearch.getContextIndex> ...
  </contentSearch.getContextIndex>
  <!-- Coveo pipeline used to retrieve the Coveo index name corresponding to the currently selected item or
       context database.
       This pipeline should only return names for the Coveo search indexes. -->
  <coveoGetContextCoveoIndex> ...
  </coveoGetContextCoveoIndex>
  <!-- Coveo pipeline used to retrieve the fields to index for a regular search index. -->
  <coveoIndexingGetFields> ...
  </coveoIndexingGetFields>
  <!-- Coveo pipeline used to retrieve the templates with the fields that should be indexed. -->
```

⤵ Coveo.SearchProvider.config

Ln 99, Col 41 (445 selected) Spaces: 2 UTF-8 CRLF XML ☺

Coveo.SearchProvider.config - Visual Studio Code [Administrator]

File Edit Selection View Go Debug Tasks Help

⤵ Coveo.SearchProvider.config ✎

```
179      <!-- Each processor in this pipeline can use the following includedIndexNames and excludedIndexNames element.  
180      -->  
181      <!-- Add this to apply the processor only to the specified indexes. -->  
182      <!-- <includedIndexNames hint="list">  
183          <indexName>Coveo_web_index</indexName>  
184      </includedIndexNames> -->  
185      <!-- Add this to apply the processor to all indexes but those specified in the list. -->  
186      <!-- <excludedIndexNames hint="list">  
187          <indexName>Coveo_master_index</indexName>  
188      </excludedIndexNames> -->  
189      <!-- Exclusion has priority over inclusion. This means that if the same index is specified both in inclusion  
190      and exclusion, the processor will not be applied. -->  
191      <coveoInboundFilterPipeline> ...  
192      </coveoInboundFilterPipeline>  
193      <!-- Pipeline used to execute Sitecore inbound filter processors. This pipeline is run by the  
194      InvokeSitecoreInboundFilterPipeline processor in the coveoInboundFilterPipeline pipeline. -->  
195      <!-- Add your existing Sitecore inbound filter processors to be run only on Coveo indexes. -->  
196      <coveoSitecoreInboundFilterPipeline>  
197      </coveoSitecoreInboundFilterPipeline>  
198      <initialize> ...  
199      </initialize>  
200      <!-- Pipeline used to modify the user context. -->  
201      <coveoUserContext>  
202      </coveoUserContext>  
203      </pipelines>  
204      <coveo> ...  
205      </coveo>  
206      <contentSearch> ...  
207      </contentSearch>
```

⤵ Coveo.SearchProvider.config ✎

Ln 179, Col 1 (2052 selected) Spaces: 2 UTF-8 CRLF XML ☺

Coveo.SearchProvider.config - Visual Studio Code [Administrator]

File Edit Selection View Go Debug Tasks Help

⤵ Coveo.SearchProvider.config ✎

209 <coveo>  
210 <caches> ...  
212 </caches>  
213 <cesCommunicationFactory type="Coveo.AbstractLayer.Communication.CES.CESCommunicationFactory,  
Coveo.AbstractLayer"> ...  
215 </cesCommunicationFactory>  
216 <cloudPlatformCommunicationFactory  
type="Coveo.CloudPlatformClient.Communication.CloudPlatformCommunicationFactory, Coveo.CloudPlatformClient"> ...  
218 </cloudPlatformCommunicationFactory>  
219 <databasePropertyStore type="Sitecore.ContentSearch.Maintenance.IndexDatabasePropertyStore,  
Sitecore.ContentSearch"> ...  
222 </databasePropertyStore>  
223 <defaultIndexConfiguration type="Coveo.AbstractLayer.CoveoDefaultIndexConfiguration, Coveo.AbstractLayer"> ...  
414 </defaultIndexConfiguration>  
415 <retryGroups> ...  
434 </retryGroups>  
435 </coveo>  
436 <contentSearch> ...  
508 </contentSearch>  
509 <scheduling> ...  
513 </scheduling>  
514 <settings> ...  
516 </settings>  
517 <sites> ...  
524 </sites>  
525 </sitecore>  
526 </configuration>

⤵ Coveo.SearchProvider.config

Ln 78, Col 16 Spaces: 2 UTF-8 CRLF XML ☺

Coveo.SearchProvider.config - Visual Studio Code [Administrator]

File Edit Selection View Go Debug Tasks Help

⤵ Coveo.SearchProvider.config ✎

```
223     <defaultIndexConfiguration type="Coveo.AbstractLayer.CoveoDefaultIndexConfiguration, Coveo.AbstractLayer">
224         <param desc="coveoReflectionFactory" type="Coveo.Framework.Utils.CoveoReflectionFactory, Coveo.Framework" />
225         <indexCommunicationFactory ref="coveo/cesCommunicationFactory"/>
226         <cloudPlatformConfiguration ref="coveo/cloudPlatformConfiguration"/>
227         <!-- Coveo AdminService configuration -->
228         <adminServiceConfiguration type="Coveo.Framework.Configuration.AdminServiceConfiguration, Coveo.Framework">...
229             </adminServiceConfiguration>
230             <!-- Coveo security configuration -->
231             <securityConfiguration type="Coveo.Framework.Configuration.SecurityConfiguration, Coveo.Framework" />
232             <!-- Coveo Query configuration -->
233             <queryConfiguration type="Coveo.Framework.Configuration.QueryConfiguration, Coveo.Framework">...
234                 </queryConfiguration>
235                 <!-- This setting must be set to 'true' in order to index profile-related fields. These fields are mandatory
236                     for automatic boosting to work properly.
237                         Otherwise, set it to 'false' to prevent those fields from being indexed. One benefit of this is that
238                             indexing speed and query time will improve. -->
239             <indexAnalyticsFields>false</indexAnalyticsFields>
240             <!-- Set this to false if you want to disable indexing of Coveo-specific fields. -->
241             <indexCoveoFields>true</indexCoveoFields>
242             <!-- Set this to true if you want referrer items to be indexed on an item update -->
243             <indexReferrerItemsOnUpdate>false</indexReferrerItemsOnUpdate>
244             <!-- Coveo search configuration -->
245             <searchCertificatePath>C:\CES7\Config\Certificates\cert-iis.p12</searchCertificatePath>
246             <searchServiceUri>https://localhost:52810/7.0/CoveoSearchService</searchServiceUri>
247             <queueUri>amqp://localhost:5682/</queueUri>
248             <queueUsername>guest</queueUsername>
249             <encryptDataOnQueue>false</encryptDataOnQueue>
250                 <!-- The farmName element specifies the name that represents the Sitecore environment.
251                     When specified, this name is used to uniquely name various resources such as the index sources
252                         and the search service. -->
```

⤵ Coveo.SearchProvider.config

Ln 219, Col 123 Spaces: 2 UTF-8 CRLF XML ☺

Coveo.SearchProvider.config - Visual Studio Code [Administrator]

File Edit Selection View Go Debug Tasks Help

⤷ Coveo.SearchProvider.config ✎

```
288     <!-- The farmName element specifies the name that represents the Sitecore environment.  
289         When specified, this name is used to uniquely name various resources such as the index sources,  
290             fieldsets, security provider, and user identity. -->  
291     <farmName></farmName>  
292     <!-- The requirementsInvalidateDelay elements sets the amount of time, in seconds, before the  
293         index requirements are invalidated. The index requirements includes the index sources, the fieldsets,  
294             security provider, and user identity. -->  
294     <requirementsInvalidateDelay>60</requirementsInvalidateDelay>  
295     <!-- The securityProviderName element sets a static name for the CES security provider. This is useful when  
many Sitecore instances  
296         are querying the same Coveo indexes. It is often required when configuring Content Management and  
Content Delivery instances. -->  
297     <securityProviderName></securityProviderName>  
298     <!-- The sitecoreUsername element specifies the user identity used by the CES security provider. -->  
299     <sitecoreUsername>sitecore\admin</sitecoreUsername>  
300     <!-- The serverUrl element specifies the URL where other hosts can connect to the Sitecore instance.  
301         This should be specified when CES is on a separate host than the Sitecore instance. -->  
302     <serverUrl></serverUrl>  
303     <!-- Specifies whether the index Initialize() method should be called as soon as the index is added, or if it  
should wait for an external trigger. -->  
304     <initializeOnAdd>true</initializeOnAdd>  
305     <fieldReaders type="Sitecore.ContentSearch.FieldReaders.FieldReaderMap, Sitecore.ContentSearch">...  
311     </fieldReaders>  
312     <fieldMap type="Coveo.SearchProvider.CoveoFieldMap, Coveo.SearchProvider">...  
356     </fieldMap>  
357     <indexFieldStorageValueFormatter type="Coveo.SearchProvider.CoveoIndexFieldStorageValueFormatter,  
Coveo.SearchProvider" >...  
381     </indexFieldStorageValueFormatter>  
382     <virtualFields type="Sitecore.ContentSearch.VirtualFieldProcessorMap, Sitecore.ContentSearch">...
```

⤷ 0 ⚠ 0 ⚠ xml | ☰ Coveo.SearchProvider.config Ln 286, Col 45 Spaces: 2 UTF-8 CRLF XML ☺

Coveo.SearchProvider.config - Visual Studio Code [Administrator]

File Edit Selection View Go Debug Tasks Help

⤵ Coveo.SearchProvider.config ✘

security provider, and user identity. -->

294 <requirementsInvalidateDelay>60</requirementsInvalidateDelay>

295 <!-- The securityProviderName element sets a static name for the CES security provider. This is useful when

many Sitecore instances

296 | | are querying the same Coveo indexes. It is often required when configuring Content Management and

Content Delivery instances. -->

297 <securityProviderName></securityProviderName>

298 <!-- The sitecoreUsername element specifies the user identity used by the CES security provider. -->

299 <sitecoreUsername>sitecore\admin</sitecoreUsername>

300 <!-- The serverUrl element specifies the URL where other hosts can connect to the Sitecore instance.

301 | | This should be specified when CES is on a separate host than the Sitecore instance. -->

302 <serverUrl></serverUrl>

303 <!-- Specifies whether the index Initialize() method should be called as soon as the index is added, or if it

should wait for an external trigger. -->

304 <initializeOnAdd>true</initializeOnAdd>

305 + <fieldReaders type="Sitecore.ContentSearch.FieldReaderMap, Sitecore.ContentSearch"> ...

311 </fieldReaders>

312 + <fieldMap type="Coveo.SearchProvider.CoveoFieldMap, Coveo.SearchProvider"> ...

356 </fieldMap>

357 <indexFieldStorageValueFormatter type="Coveo.SearchProvider.CoveoIndexFieldStorageValueFormatter,

+ Coveo.SearchProvider" > ...

381 </indexFieldStorageValueFormatter>

382 + <virtualFields type="Sitecore.ContentSearch.VirtualFieldProcessorMap, Sitecore.ContentSearch"> ...

389 </virtualFields>

390 + <documentOptions type="Sitecore.ContentSearch.DocumentBuilderOptions, Sitecore.ContentSearch"> ...

412 </documentOptions>

413 <searchServiceProvider type="Coveo.SearchServiceProvider.Rest.ClientSessionWrapperFactory,

Coveo.SearchServiceProvider.Rest" />

414 </defaultIndexConfiguration>

⤵ 0 ! 0 ⌂ xml | Coveo.SearchProvider.config

Ln 434, Col 21 Spaces: 2 UTF-8 CRLF XML ☺

Coveo.SearchProvider.config - Visual Studio Code [Administrator]

File Edit Selection View Go Debug Tasks Help

⤵ Coveo.SearchProvider.config ✖

```
298     <!-- The sitecoreUsername element specifies the user identity used by the CES security provider. -->
299     <sitecoreUsername>sitecore\admin</sitecoreUsername>
300     <!-- The serverUrl element specifies the URL where other hosts can connect to the Sitecore instance.
301         This should be specified when CES is on a separate host than the Sitecore instance. -->
302     <serverUrl></serverUrl>
303     <!-- Specifies whether the index Initialize() method should be called as soon as the index is added, or if it
304         should wait for an external trigger. -->
305     <initializeOnAdd>true</initializeOnAdd>
306     <fieldReaders type="Sitecore.ContentSearch.FieldReaders.FieldReaderMap, Sitecore.ContentSearch"> ...
307     </fieldReaders>
308     <fieldMap type="Coveo.SearchProvider.CoveoFieldMap, Coveo.SearchProvider">
309         <param desc="coveoReflectionFactory" type="Coveo.Framework.Utils.CoveoReflectionFactory, Coveo.Framework" />
310         <fieldNames hint="raw:AddFieldByFieldName"> ...
311             </fieldNames>
312             <!-- Fields in this section are fields that do not exist on Sitecore items, but exist in CES or in external
313                 sources. -->
314             <externalFields hint="raw:AddExternalField"> ...
315             </externalFields>
316         </fieldMap>
317         <indexFieldStorageValueFormatter type="Coveo.SearchProvider.CoveoIndexFieldStorageValueFormatter,
318             Coveo.SearchProvider" >...
319         </indexFieldStorageValueFormatter>
320         <virtualFields type="Sitecore.ContentSearch.VirtualFieldProcessorMap, Sitecore.ContentSearch"> ...
321         </virtualFields>
322         <documentOptions type="Sitecore.ContentSearch.DocumentBuilderOptions, Sitecore.ContentSearch"> ...
323         </documentOptions>
324         <searchServiceProvider type="Coveo.SearchServiceProvider.Rest.ClientSessionWrapperFactory,
325             Coveo.SearchServiceProvider.Rest" />
326     </defaultIndexConfiguration>
```

⤵ Coveo.SearchProvider.config

Ln 312, Col 1 (4341 selected) Spaces: 2 UTF-8 CRLF XML ☺

## &lt;/&gt; Coveo.SearchProvider.config x

```
300      <!-- The serverUrl element specifies the URL where other nodes can connect to the Sitecore instance. -->
301      | This should be specified when CES is on a separate host than the Sitecore instance. -->
302      <serverUrl></serverUrl>
303      <!-- Specifies whether the index Initialize() method should be called as soon as the index is added, or if it
304      should wait for an external trigger. -->
305      <initializeOnAdd>true</initializeOnAdd>
306      <fieldReaders type="Sitecore.ContentSearch.FieldReaderMap, Sitecore.ContentSearch"> ...
307      </fieldReaders>
308      <fieldMap type="Coveo.SearchProvider.CoveoFieldMap, Coveo.SearchProvider"> ...
309      </fieldMap>
310      <indexFieldStorageValueFormatter type="Coveo.SearchProvider.CoveoIndexFieldStorageValueFormatter,
311      Coveo.SearchProvider" > ...
312      </indexFieldStorageValueFormatter>
313      <virtualFields type="Sitecore.ContentSearch.VirtualFieldProcessorMap, Sitecore.ContentSearch"> ...
314      </virtualFields>
315      <documentOptions type="Sitecore.ContentSearch.DocumentBuilderOptions, Sitecore.ContentSearch">
316          <indexAllFields>true</indexAllFields>
317          <fields hint="raw:AddComputedIndexField"> ...
318          </fields>
319          <exclude hint="list:AddExcludedTemplate"> ...
320          </exclude>
321          <exclude hint="list:AddExcludedField" ref="mandatory, do not delete" factory="excludedFieldsFactory" />
322      </documentOptions>
323
324      <searchServiceProvider type="Coveo.SearchServiceProvider.Rest.ClientSessionWrapperFactory,
325      Coveo.SearchServiceProvider.Rest" />
326
327      </defaultIndexConfiguration>
328      <retryGroups> ...
329      </retryGroups>
330
331  </coveo>
```

Coveo.SearchProvider.config - Visual Studio Code [Administrator]

File Edit Selection View Go Debug Tasks Help

✖ Coveo.SearchProvider.config ✎

```
436 <contentSearch>
437   <indexConfigurations>
438     <indexUpdateStrategies>...
446   </indexUpdateStrategies>
447 </indexConfigurations>
448 <!-- The configuration element appears twice for these reasons:
449   1. If no other search provider is defined, the type needs to be specified to create the configuration
450     element.
451   2. If an other search provider is configured already and overrides the type of the configuration element,
452     Coveo indexes are
453       added to the same configuration element as other indexes. -->
452 <configuration type="Sitecore.ContentSearch.ContentSearchConfiguration, Sitecore.ContentSearch" />
453 <configuration>
454   <indexes hint="list:AddIndex">
455     <index id="Coveo_master_index" type="Coveo.SearchProvider.ProviderIndex, Coveo.SearchProvider" patch:before
456       = "*[1]">...
483   </index>
484     <index id="Coveo_web_index" type="Coveo.SearchProvider.ProviderIndex, Coveo.SearchProvider" patch:before =
485       "*[2]">...
505   </index>
506 </indexes>
507 </configuration>
508 </contentSearch>
509 <scheduling>...
513 </scheduling>
514 <settings>...
516 </settings>
517 <sites>...
521 </sites>
```

✖ 0 ! 0 ! xml | Coveo.SearchProvider.config

Ln 209, Col 12 Spaces: 2 UTF-8 CRLF XML ☺

Coveo.SearchProvider.config - Visual Studio Code [Administrator]

File Edit Selection View Go Debug Tasks Help

⤵ Coveo.SearchProvider.config ✎

```
<factories> ...
</factories>
<pipelines> ...
</pipelines>
<coveo> ...
</coveo>
<contentSearch> ...
</contentSearch>
<scheduling> ...
</scheduling>
<settings> ...
</settings>
<sites>
    <!-- The site coveorest needs to be inserted as the first element, or before any other sites that have the coveorest site as a child (e.g. sites defined with virtualpath="/"). -->
    <site patch:before="*[1]" name="coveorest" virtualFolder="/coveo/rest" physicalFolder="/coveo/rest" enableTracking="false" database="web" domain="extranet" />
    <!-- The site coveoanalytics needs to be inserted before coveorest, as it requires analytics to be enabled. -->
    <!-- The coveoanalytics website database attribute value should match the published website database attribute value in order to allow Coveo Usage Analytics logging. -->
    <site patch:before="*[1]" name="coveoanalytics" virtualFolder="/coveo/rest/v6/analytics" enableTracking="true" database="web" domain="extranet" />
    <site patch:before="*[1]" name="coveo_website" virtualFolder="/sitecore modules/Web/Coveo" physicalFolder="/sitecore modules/Web/Coveo" rootPath="/sitecore/content" startItem="/home" language="en" database="web" domain="extranet" allowDebug="true" cacheHtml="true" loginPage="/sitecore/login" />
</sites>
</sitecore>
</configuration>
```

0 0 0 !xml | Coveo.SearchProvider.config

Ln 8, Col 34 Spaces: 2 UTF-8 CRLF XML ☺

Coveo.SearchProvider.config - Visual Studio Code [Administrator]

File Edit Selection View Go Debug Tasks Help

⤵ Coveo.SearchProvider.config ✎

68 **<factories>** ...  
77 **</factories>**  
78 **<pipelines>** ...  
208 **</pipelines>**  
209 **<coveo>** ...  
435 **</coveo>**  
436 **<contentSearch>** ...  
508 **</contentSearch>**  
509 **<scheduling>** ...  
513 **</scheduling>**  
514 **<settings>** ...  
516 **</settings>**  
517 **<sites>**  
518     **<!-- The site coveorest needs to be inserted as the first element, or before any other sites that have the coveorest site as a child (e.g. sites defined with virtualpath="/"). -->**  
519     **<site patch:before="\*[1]" name="coveorest" virtualFolder="/coveo/rest" physicalFolder="/coveo/rest" enableTracking="false" database="web" domain="extranet" />**  
520     **<!-- The site coveoanalytics needs to be inserted before coveorest, as it requires analytics to be enabled. -->**  
521     **<!-- The coveoanalytics website database attribute value should match the published website database attribute value in order to allow Coveo Usage Analytics logging. -->**  
522     **<site patch:before="\*[1]" name="coveoanalytics" virtualFolder="/coveo/rest/v6/analytics" enableTracking="true" database="web" domain="extranet" />**  
523     **<site patch:before="\*[1]" name="coveo\_website" virtualFolder="/sitecore modules/Web/Coveo" physicalFolder="/sitecore modules/Web/Coveo" rootPath="/sitecore/content" startItem="/home" language="en" database="web" domain="extranet" allowDebug="true" cacheHtml="true" loginPage="/sitecore/login" />**  
524     **</sites>**  
525     **</sitecore>**  
526     **</configuration>**

⤵ 0 ⚠ 0 ⚠ xml | Coveo.SearchProvider.config

Ln 519, Col 1 (164 selected) Spaces: 2 UTF-8 CRLF XML ☺

Coveo.SearchProvider.config - Visual Studio Code [Administrator]

File Edit Selection View Go Debug Tasks Help

✖ Coveo.SearchProvider.config ✎

```
68 <factories> ...
77 </factories>
78 <pipelines> ...
208 </pipelines>
209 <coveo> ...
435 </coveo>
436 <contentSearch> ...
508 </contentSearch>
509 <scheduling> ...
513 </scheduling>
514 <settings> ...
516 </settings>
517 <sites>
518     <!-- The site coveorest needs to be inserted as the first element, or before any other sites that have the coveorest site as a child (e.g. sites defined with virtualpath="/"). -->
519     <site patch:before="*[1]" name="coveorest" virtualFolder="/coveo/rest" physicalFolder="/coveo/rest"
enableTracking="false" database="web" domain="extranet" />
520     <!-- The site coveoanalytics needs to be inserted before coveorest, as it requires analytics to be enabled. -->
521     <!-- The coveoanalytics website database attribute value should match the published website database attribute
value in order to allow Coveo Usage Analytics logging. -->
522     <site patch:before="*[1]" name="coveoanalytics" virtualFolder="/coveo/rest/v6/analytics" enableTracking="true"
database="web" domain="extranet" />
523     <site patch:before="*[1]" name="coveo_website" virtualFolder="/sitecore modules/Web/Coveo"
physicalFolder="/sitecore modules/Web/Coveo" rootPath="/sitecore/content" startItem="/home" language="en"
database="web" domain="extranet" allowDebug="true" cacheHtml="true" loginPage="/sitecore/login" />
524 </sites>
525 </sitecore>
526 </configuration>
```

Ln 522, Col 1 (152 selected) Spaces: 2 UTF-8 CRLF XML ☺

Coveo.SearchProvider.config - Visual Studio Code [Administrator]

File Edit Selection View Go Debug Tasks Help

### Coveo.SearchProvider.config

```
</> Coveo.SearchProvider.config x
```

```
68 <factories> ...
77 </factories>
78 <pipelines> ...
208 </pipelines>
209 <coveo> ...
435 </coveo>
436 <contentSearch> ...
508 </contentSearch>
509 <scheduling> ...
513 </scheduling>
514 <settings> ...
516 </settings>
517 <sites>
518     <!-- The site coveorest needs to be inserted as the first element, or before any other sites that have the coveorest site as a child (e.g. sites defined with virtualpath="/"). -->
519     <site patch:before="*[1]" name="coveorest" virtualFolder="/coveo/rest" physicalFolder="/coveo/rest"
enableTracking="false" database="web" domain="extranet" />
520     <!-- The site coveoanalytics needs to be inserted before coveorest, as it requires analytics to be enabled. -->
521     <!-- The coveoanalytics website database attribute value should match the published website database attribute
value in order to allow Coveo Usage Analytics logging. -->
522     <site patch:before="*[1]" name="coveoanalytics" virtualFolder="/coveo/rest/v6/analytics" enableTracking="true"
database="web" domain="extranet" />
523     <site patch:before="*[1]" name="coveo_website" virtualFolder="/sitecore modules/Web/Coveo"
physicalFolder="/sitecore modules/Web/Coveo" rootPath="/sitecore/content" startItem="/home" language="en"
database="web" domain="extranet" allowDebug="true" cacheHtml="true" loginPage="/sitecore/login" />
524     </sites>
525 </sitecore>
526 </configuration>
```

Ln 523, Col 1 (301 selected) Spaces: 2 UTF-8 CRLF XML ☺

# Coveo.SearchProvider.Custom.config - Common

- pipelines
  - examples
  - HtmlContentInBodyWithRequestsProcessor (if configured at installation)
- coveo
  - defaultIndexConfiguration
    - Field configuration examples
    - External fields examples
    - Computed Index Fields examples
    - Sitecore credentials
    - Index permissions flag
- log4net example configuration

# Coveo.SearchProvider.Custom.config - Cloud

- coveo
  - defaultIndexConfiguration
    - Cloud indexCommunicationFactory
- scheduling
  - Delete CoveoSynchronizeTopResults agent

## Coveo.SearchProvider.Custom.config x

```
1  <!--  
2   This file patches the Coveo.SearchProvider.config file default configuration.  
3   It is automatically modified by Coveo for Sitecore configuration wizards.  
4   This is where you should add your custom configuration.  
5   See the parent file for all the configuration elements and additional descriptions.  
6  -->  
7  <configuration xmlns:patch="http://www.sitecore.net/xmlconfig/">  
8    <sitecore>  
9      <pipelines>  
10     <!-- ...  
14     -->  
15     <coveoQueryFieldPipeline> ...  
24     </coveoQueryFieldPipeline>  
25     <coveoInboundFilterPipeline>  
26       <processor type="Coveo.SearchProvider.InboundFilters.ApplySitecoreInboundFilterProcessor,  
27           Coveo.SearchProviderBase"> ...  
40       </processor>  
41     </coveoInboundFilterPipeline>  
42     <coveoPostItemProcessingPipeline>  
43       <processor type="Coveo.SearchProvider.Processors.HtmlContentInBodyWithRequestsProcessor,  
44           Coveo.SearchProviderBase" />  
44     </coveoPostItemProcessingPipeline>  
45   </pipelines>  
46   <coveo> ...  
124   </coveo>  
125   <!-- Example of Coveo for Sitecore log4net configuration. ...  
141   -->  
142   <!-- ...  
161   -->
```

## Coveo.SearchProvider.Custom.config

```
4      THIS IS WHERE YOU SHOULD ADD YOUR CUSTOM CONFIGURATION.  
5      See the parent file for all the configuration elements and additional descriptions.  
6  -->  
7  <configuration xmlns:patch="http://www.sitecore.net/xmlconfig/">  
8      <sitecore>  
9          <pipelines> ...  
10         </pipelines>  
11     <coveo>  
12         <defaultIndexConfiguration>  
13             <fieldMap>  
14                 <fieldNames hint="raw:AddFieldByFieldName"> ...  
15             </fieldNames>  
16             <externalFields hint="raw:AddExternalField"> ...  
17                 </externalFields>  
18             </fieldMap>  
19             <documentOptions>  
20                 <fields hint="raw:AddComputedIndexField"> ...  
21                     </fields>  
22             </documentOptions>  
23             <indexCommunicationFactory>  
24                 <patch:attribute name="ref">coveo/cloudPlatformCommunicationFactory</patch:attribute>  
25             </indexCommunicationFactory>  
26             <sitecorePassword>lnPFJ8S/ZVgITsX0zpC6WQ==</sitecorePassword>  
27             <sitecoreUsername>sitecore\admin</sitecoreUsername>  
28             <securityConfiguration>  
29                 <indexPermissions>True</indexPermissions>  
30             </securityConfiguration>  
31         </defaultIndexConfiguration>  
32     </coveo>
```





## Coveo.SearchProvider.Custom.config

```
7   <configuration xmlns:patch="http://www.sitecore.net/xmlconfig/">
8     <sitecore>
9       <pipelines> ...
10      </pipelines>
11      <coveo> ...
12    </coveo>
13    <!-- Example of Coveo for Sitecore log4net configuration. ...
14    -->
15    <!--
16      <log4net>
17        <appender name="CoveoLogger" type="log4net.Appender.RollingFileAppender, Sitecore.Logging">
18          <file value="$(dataFolder)/logs/Coveo.Search" />
19          <appendToFile value="true" />
20          <rollingStyle value="Composite" />
21          <datePattern value=".yyyyMMdd'.txt'" />
22          <staticLogFileName value="false" />
23          <maxSizeRollBackups value="20" />
24          <maximumFileSize value="250MB" />
25          <layout type="log4net.Layout.PatternLayout">
26            <conversionPattern value="%4t %d{ABSOLUTE} %1 %-5p %m%n" />
27          </layout>
28        </appender>
29        <logger name="Coveo" additivity="false">
30          <level value="INFO" />
31          <appender-ref ref="CoveoLogger" />
32        </logger>
33      </log4net>
34    <!-->
```



## Coveo.SearchProvider.Custom.config

```
1  <!--  
2   This file patches the Coveo.SearchProvider.config file default configuration.  
3   It is automatically modified by Coveo for Sitecore configuration wizards.  
4   This is where you should add your custom configuration.  
5   See the parent file for all the configuration elements and additional descriptions.  
6  -->  
7  <configuration xmlns:patch="http://www.sitecore.net/xmlconfig/">  
8    <sitecore>  
9      <pipelines> ...  
10     </pipelines>  
11     <coveo> ...  
12     </coveo>  
13     <!-- Example of Coveo for Sitecore log4net configuration. ...  
14     -->  
15     <!-- ...  
16     -->  
17     <scheduling>  
18       <agent name="CoveoSynchronizeTopResults">  
19         <patch:delete />  
20       </agent>  
21     </scheduling>  
22     </sitecore>  
23   </configuration>
```



# Coveo.SearchProvider.Custom.config - On-Premises

- Pipelines
  - getSearchUserIdentities
    - Delete the AddAnonymousEmailUserIdentity processor
- coveo
  - defaultIndexConfiguration
    - CES indexCommunicationFactory
    - adminServiceConfiguration
    - searchCertificatePath
    - Queue configuration

# Coveo.SearchProvider.Rest.config

- pipelines
- coveo
  - restEndpointConfiguration

## Coveo.SearchProvider.Rest.config x

```
1  <!--  
2   Do not modify this configuration file.  
3   Instead, you should patch this file using Coveo.SearchProvider.Rest.Custom.config.  
4  -->  
5  <configuration xmlns:patch="http://www.sitecore.net/xmlconfig/">  
6    <sitecore>  
7      <pipelines> ...  
67     </pipelines>  
68     <coveo> ...  
83     </coveo>  
84   </sitecore>  
85 </configuration>
```



## Coveo.SearchProvider.Rest.config x

```
1  <!--  
2   Do not modify this configuration file.  
3   Instead, you should patch this file using Coveo.SearchProvider.Rest.Custom.config.  
4  -->  
5  <configuration xmlns:patch="http://www.sitecore.net/xmlconfig/">  
6    <sitecore>  
7      <pipelines>  
8        <!-- Pipeline used to customize the JSON response of the search REST endpoint. -->  
9        <coveoProcessRestJsonResponse> ...  
10       </coveoProcessRestJsonResponse>  
11       <!-- Pipeline used to customize the information returned by the REST endpoint. -->  
12       <coveoProcessParsedRestResponse> ...  
13       </coveoProcessParsedRestResponse>  
14       <getSearchUserIdentities> ...  
15       </getSearchUserIdentities>  
16       <getUserMappings> ...  
17       </getUserMappings>  
18       <!-- The "getUserMembers" is exposed by Sitecore2's web service. Therefore, the "getUserMappings" pipeline  
19       | | | should be used instead to control to which identities a Sitecore user identity is expanded. -->  
20       <getUserMembers> ...  
21       </getUserMembers>  
22       <httpRequestBegin> ...  
23       </httpRequestBegin>  
24       <httpRetry.SearchExceptionsHandler> ...  
25       </httpRetry.SearchExceptionsHandler>  
26     </pipelines>  
27     <coveo> ...  
28     </coveo>  
29   </sitecore>
```

## Coveo.SearchProvider.Rest.config x

```
1  <!--  
2   Do not modify this configuration file.  
3   Instead, you should patch this file using Coveo.SearchProvider.Rest.Custom.config.  
4 -->  
5 <configuration xmlns:patch="http://www.sitecore.net/xmlconfig/">  
6   <sitecore>  
7     <pipelines>  
8       <!-- Pipeline used to customize the JSON response of the search REST endpoint. -->  
9       <coveoProcessRestJsonResponse>  
10         <processor  
11           type="Coveo.SearchProvider.Rest.Processors.CoveoProcessRestJsonResponse.ParseJsonRestSearchResponseProcessor,  
12             Coveo.SearchProvider.Rest" />  
13       </coveoProcessRestJsonResponse>  
14       <!-- Pipeline used to customize the information returned by the REST endpoint. -->  
15       <coveoProcessParsedRestResponse>  
16         <!-- This processor resolves the REST results clickUri field in the context of the current site using the  
17           Sitecore LinkManager. -->  
18         <processor  
19           type="Coveo.SearchProvider.Rest.Processors.CoveoProcessParsedRestResponse.ResolveResultClickableUriProcessor,  
20             Coveo.SearchProvider.Rest" />  
21       </coveoProcessParsedRestResponse>  
22       <getSearchUserIdentities>...  
23     </getSearchUserIdentities>  
24     <getUserMappings>...  
25   </getUserMappings>  
26   <!-- The "getUserMembers" is exposed by Sitecore2's web service. Therefore, the "getUserMappings" pipeline  
27   |   | should be used instead to control to which identities a Sitecore user identity is expanded. -->  
28     <getUserMembers>...  
29   </getUserMembers>
```

## Coveo.SearchProvider.Rest.config x

```
1  <!--  
2   Do not modify this configuration file.  
3   Instead, you should patch this file using Coveo.SearchProvider.Rest.Custom.config.  
4  -->  
5  <configuration xmlns:patch="http://www.sitecore.net/xmlconfig/">  
6    <sitecore>  
7      <pipelines>...  
8    </pipelines>  
9    <coveo>  
10      <!-- Coveo Rest endpoint search settings -->  
11      <restEndpointConfiguration type="Coveo.Framework.Configuration.RestEndpointConfiguration, Coveo.Framework">  
12          <!-- The secret key must be configured in the search api service as well. This way, the search api service  
13          can trust the application. -->  
14          <applicationSecret>replace this by your own application secret</applicationSecret>  
15          <!-- The timeout period for requests going through the search api proxy.-->  
16          <requestTimeout>00:00:30</requestTimeout>  
17          <!-- Defines the location of the REST endpoint in the website. -->  
18          <restEndpointPath>/coveo/rest</restEndpointPath>  
19          <!-- The period during which the search token is valid. -->  
20          <searchTokenExpiration>01:00:00</searchTokenExpiration>  
21          <!-- The URI to connect to the search api service. -->  
22          <searchApiUri>http://localhost:8080</searchApiUri>  
23          <analyticsUri>https://usageanalytics.coveo.com</analyticsUri>  
24      </restEndpointConfiguration>  
25    </coveo>  
26  </sitecore>  
27 </configuration>
```

# Coveo.SearchProvider.Rest.Custom.config - Common

- coveo
  - restEndpointConfiguration
    - searchApiUri
    - apiKey
    - organizationId
    - analyticsUri

## Coveo.SearchProvider.Rest.Custom.config ✘

```
1 <configuration xmlns:patch="http://www.sitecore.net/xmlconfig/">
2   <sitecore>
3     <coveo>
4       <restEndpointConfiguration>
5         <searchApiUri>
6           <patch:attribute name="ref">coveo/cloudPlatformConfiguration/cloudPlatformUri</patch:attribute>
7         </searchApiUri>
8         <apiKey>
9           <patch:attribute name="ref">coveo/cloudPlatformConfiguration/apiKey</patch:attribute>
10        </apiKey>
11        <organizationId>
12          <patch:attribute name="ref">coveo/cloudPlatformConfiguration/organizationId</patch:attribute>
13        </organizationId>
14        <analyticsUri>https://usageanalytics.coveo.com</analyticsUri>
15      </restEndpointConfiguration>
16    </coveo>
17  </sitecore>
18 </configuration>
```

# Coveo.SearchProvider.Rest.Custom.config - On-Premises

- coveo
  - restEndpointConfiguration
    - applicationSecret

# Sitecore Data Folder

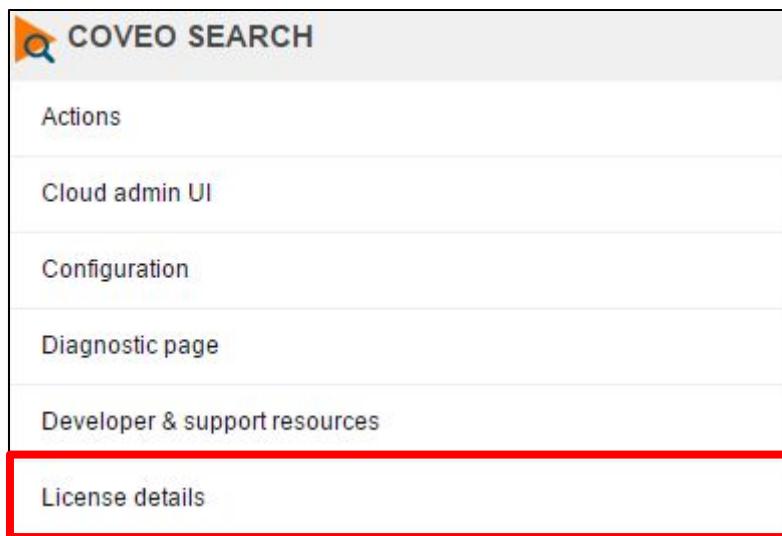
\Data

- \Coveo
  - \ConfigurationEncryptionKeys (Empty folder. Used to contain the Coveo for Sitecore encryption key for configuration sensitive information. It is stored in the core database property store since the December 2016 release.)
- coveolicense.xml (Cached Coveo for Sitecore license code)

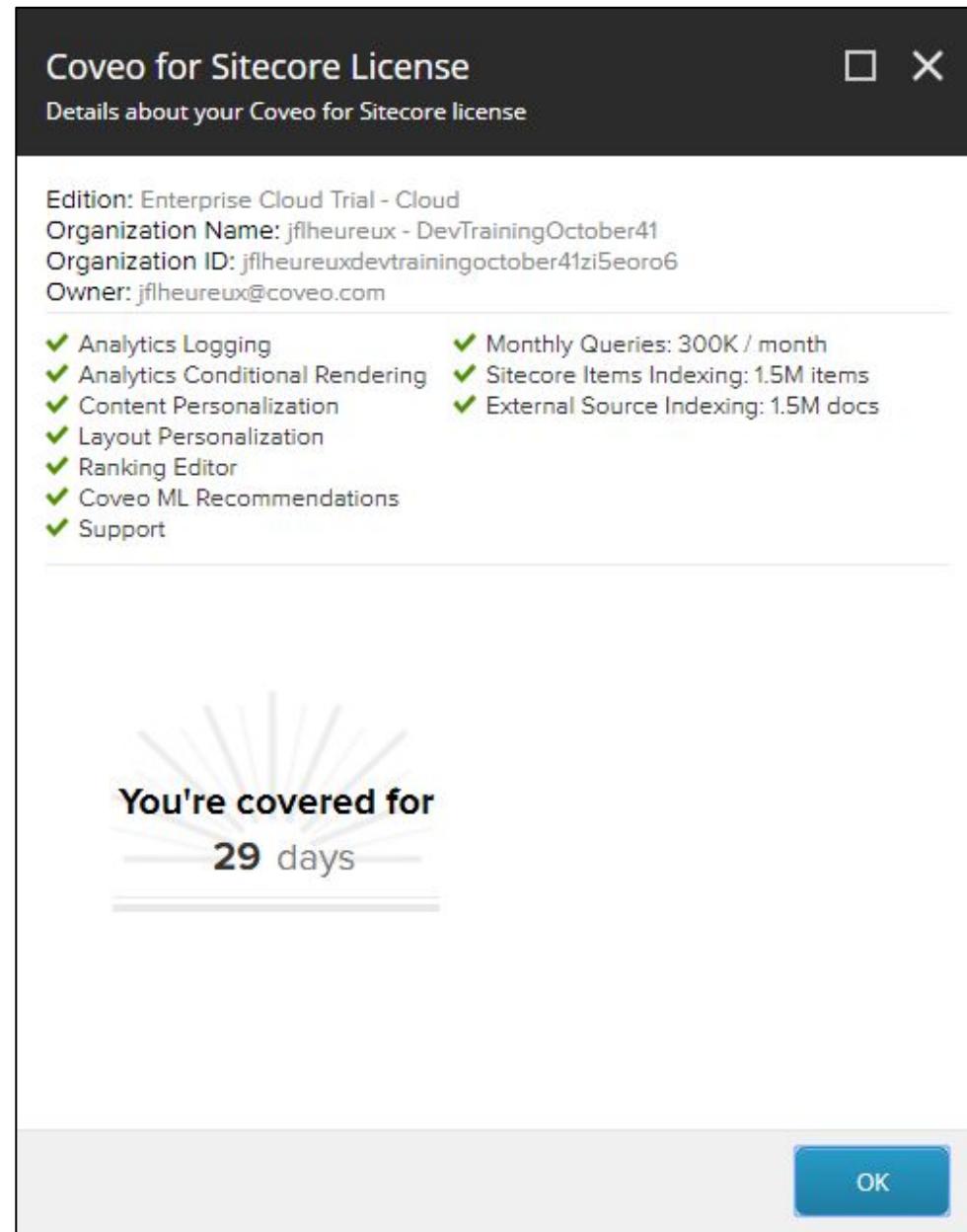
# Managing the Coveo for Sitecore License

## 1. Open the License Details dialog

- a. Sitecore Control Panel > Coveo Search > License details



The Coveo for Sitecore license is automatically fetched from the Coveo Cloud Platform at each application pool recycle. It can be manually fetched by synchronizing the Coveo configuration.



# 1. Administration

Q&A



## 2. Customization

## 2. Customization

### 1. Creating a Search Interface

- a. Best Practices
- b. Coveo for Sitecore Hive UI Components
- c. Data Sources
- d. Statically Loaded Components
- e. Global Searchbox
- f. Result Filtering, Ranking, and Boosting
- g. Result Templates
- h. Styling

### 2. Coveo JavaScript Search Framework

### 3. Configuration

- a. Indexing Scope
- b. Fields & Computed Index Fields

### 4. Query Syntax

### 5. Free-Text Search

### 6. External Content



# UI Architecture

# Expected Result - Search Box

A screenshot of the Sitecore website's header and navigation bar. The header has a dark purple background. On the left is the Sitecore logo with the tagline "Own the experience". On the right are three buttons: "login", "register", and "sites ▾". Below the header is a dark grey navigation bar with links: "About Launch Sitecore ▾", "Team", "Glossary", "Search" (which is highlighted in white), and a magnifying glass icon.

Home &gt; Search Results

## Type



- Article 1
- Article Group 1
- Team Member 1
- Team Section 1

Results 1-4 of 4

RELEVANCY DATE ▾



## Casey Piket

Senior Regional Sales Engineer, SouthEast

As a Sales Engineer, Casey has helped educate customers and design solutions to solve significant business challenges for organizations of all sizes. Casey has a long track record working with software solutions that improve the delivery and discovery of content.



## Minimizing Downtime During Deployment

2/18/2014

Workflow ... CM - Content Management Server CD1 - Content Delivery Server 1 CD2 - Content Delivery Server 2 We have the 3 standard Sitecore Databases and an extra web database called the Live ...

## Details

Created By sitecorechc

Created 10/28/2011

Updated By sitecoreadmin

Language english

URI </about-launch-sitecore/our-strategy/minimizing-downtime-during-deployment>

Tags CMS, How-to



## Our Strategy

8/4/2015

About Launch Sitecore ... Workflow ... Building a large, multi-national, multilingual site ... The separation of authoring and delivery instances allow us to perform these migrations in steps.

## Details ▾



## Team

7/9/2015

Home ... Casey has a long track record working with software solutions that improve the delivery and discovery of content. ... If we are to undertake a task, we must do it with excellence.

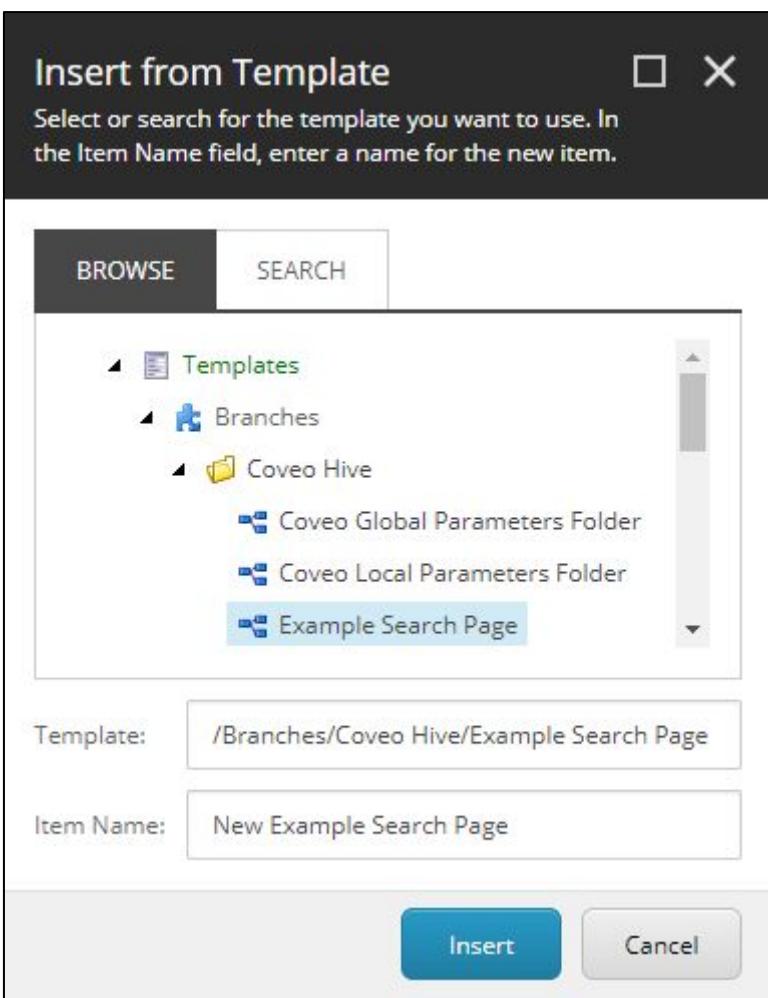
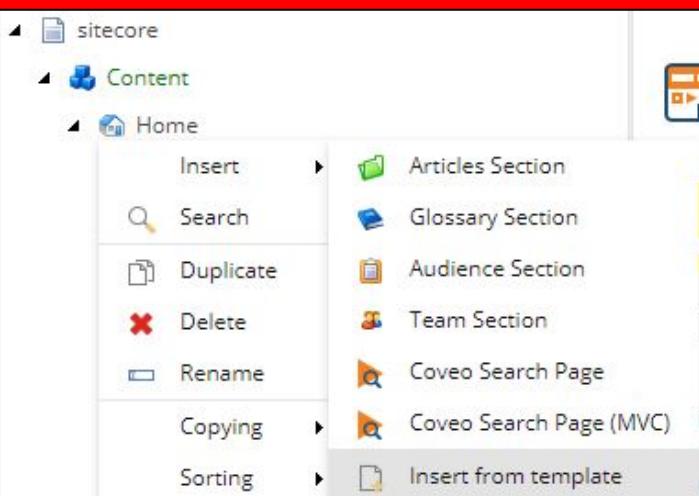
## Details ▾

# Creating an Example Search Page (1/2)

1. From the Sitecore Content Editor, select the Home content item
  - a. /sitecore/Content/Home

2. Insert a new Example Search Page item  
(Right-click > Insert > Insert from template)

3. In the Insert from Template dialog
  - a. Template: /Branches/Coveo Hive/Example Search Page
  - b. Item Name: New Example Search Page
  - c. Choose Insert



# Creating an Example Search Page (2/2)

## 4. Validate that your page is working (Publish > Preview)

- You should see 61 or 62 search results

The screenshot shows the Sitecore Content Editor interface. On the left, the navigation tree displays a folder structure under 'Content' for 'Home' and 'Audience'. A specific item, 'New-Example-Search-Page', is selected and highlighted in blue. The main content area shows a search interface with a search bar, a 'Created Date' filter set to 'Dec 31, 2007 - Dec 31, 2017', and a search results table. The table has two columns: 'Template Name' and a count. The results are as follows:

Template Name	Count
Article	21
Term	16
Team Member	7
Article Group	4
Audience Segment	4
Search	1

The 'Details' pane on the right shows the properties for the selected item: 'Created By' is 'sitecoreadmin', 'Created' is 'Today, 11:35 AM', 'Updated By' is 'sitecoreadmin', 'Language' is 'english', and the 'URI' is '<http://devtrainingoctober41.local/new-example-search-page>'. The status bar at the bottom right indicates the date '4/16/2016'.

# Lab

1. Create an example Coveo search page under the Home item
2. Use the page in preview mode

# How it Works Behind the Scenes

## Coveo Search Interface

### Coveo UI

**Modular Frame / Modular Frame - 3 columns / Default Frame**

**Header Section**

**UI Header**

**Main Section**

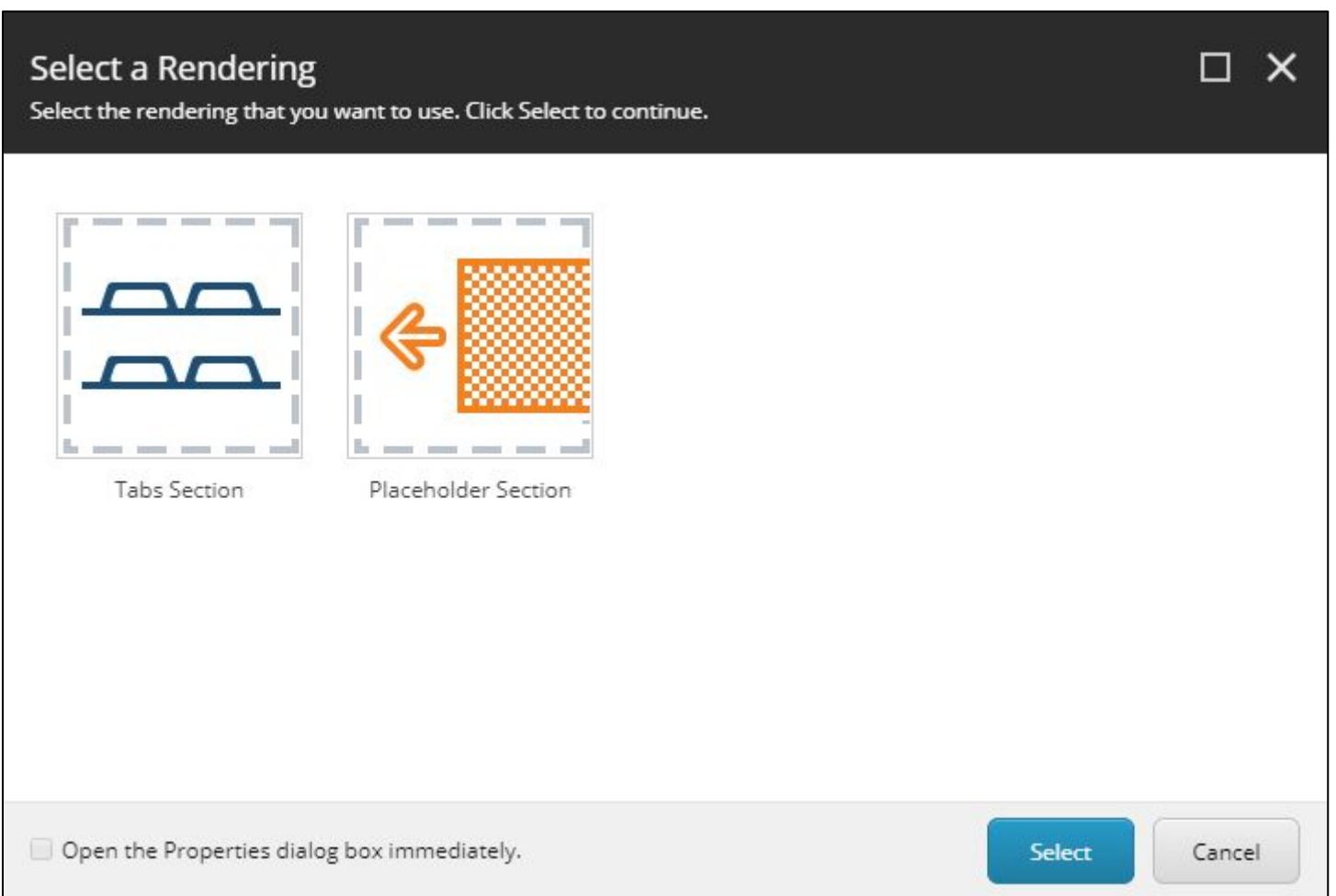
**UI Footer**

**Footer Section**

# Coveo for Sitecore Hive - Frames

Rendering	Description
Modular Frame	<p>5 placeholders for a maximum of flexibility:</p> <ul style="list-style-type: none"><li>● Header Section</li><li>● UI Header</li><li>● Main Section</li><li>● UI Footer</li><li>● Footer Section</li></ul>
Modular Frame - 3 columns	Same as Modular Frame, but with an additional CSS class to force a 3 column layout in the Main Section
Default Frame	<p>More placeholders in a predefined layout:</p> <ul style="list-style-type: none"><li>● Searchbox Settings, Searchbox, Searchbox Options</li><li>● UI Header</li><li>● Facets</li><li>● Tabs</li><li>● Results Summary Section, Results Layout</li><li>● Sorts</li><li>● UI Results Header, Results List, UI Results Footer</li><li>● UI Footer</li></ul>

# Header Section



# UI Header

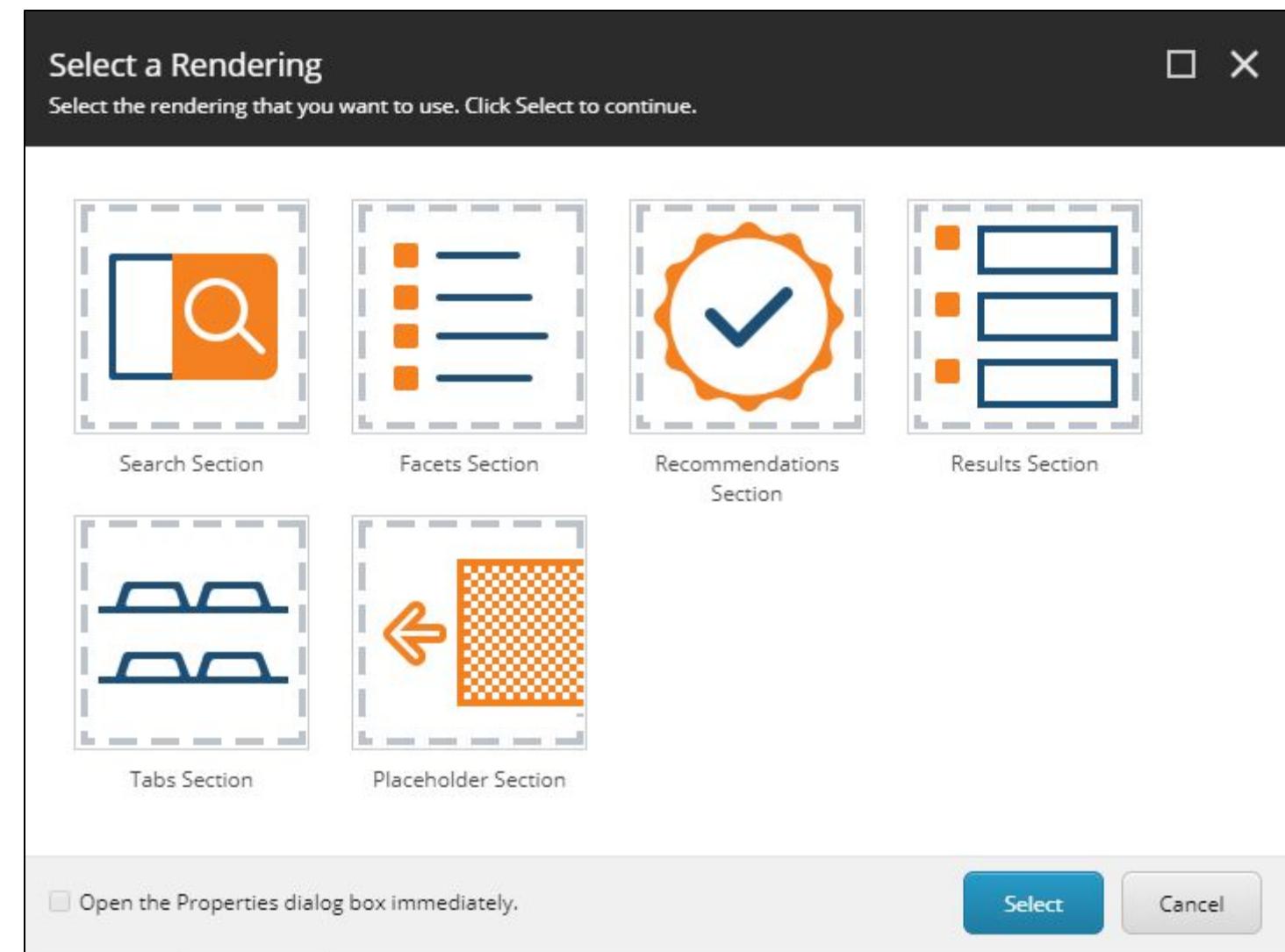
**Select a Rendering**  
Select the rendering that you want to use. Click Select to continue.

The dialog box contains 16 items arranged in two rows of eight:

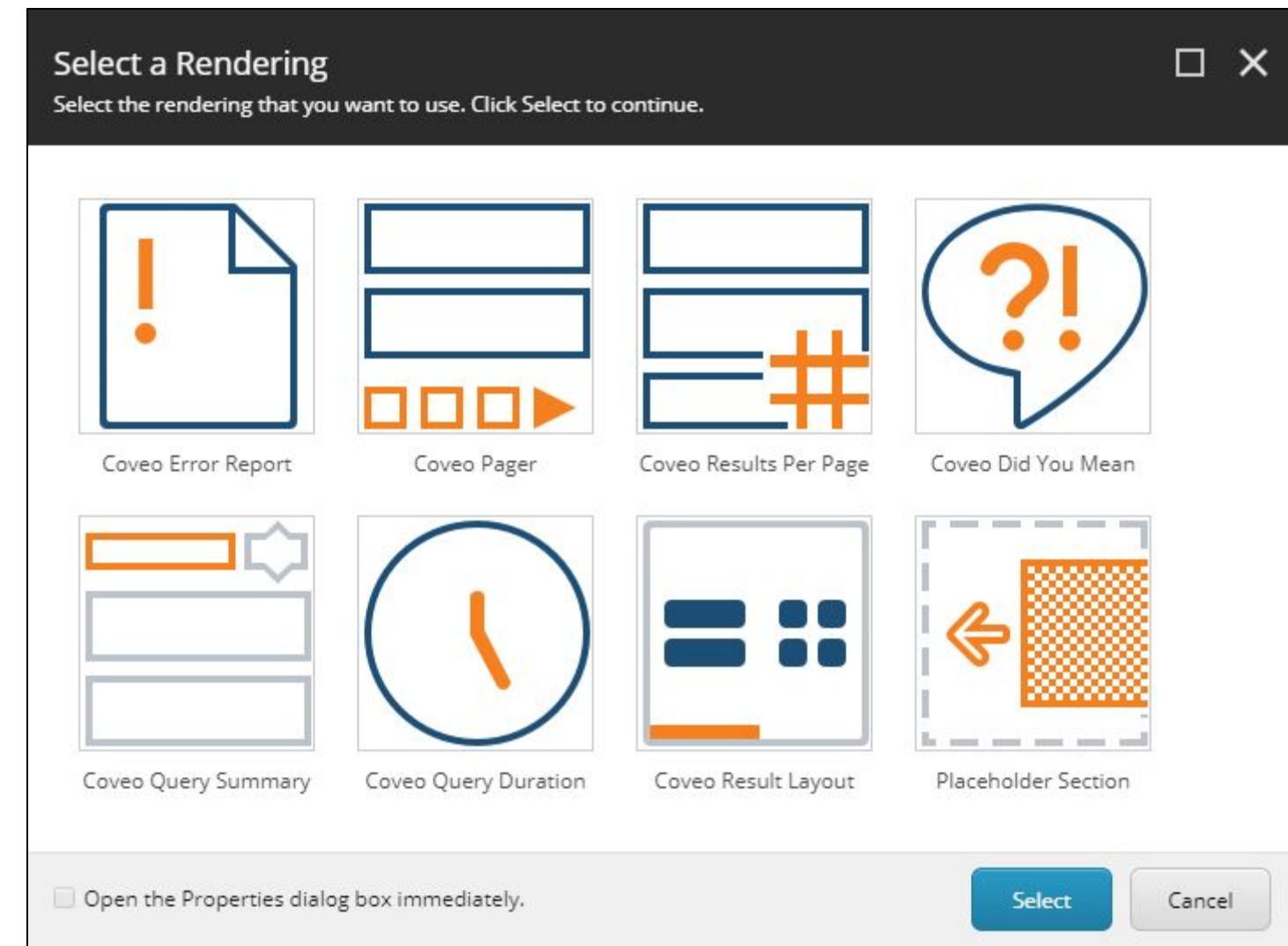
- Coveo Error Report
- Coveo Pager
- Coveo Results Per Page
- Coveo Did You Mean
- Coveo Breadcrumbs
- Coveo Query Summary
- Coveo Query Duration
- Coveo Send Analytics To Sitecore
- Tabs Section
- Query Filter
- Query Ranking
- External Content
- Automatic Boosting
- Coveo Distance Resources
- Coveo For Sitecore Analytics
- Placeholder Section

At the bottom left is a checkbox labeled "Open the Properties dialog box immediately." At the bottom right are "Select" and "Cancel" buttons.

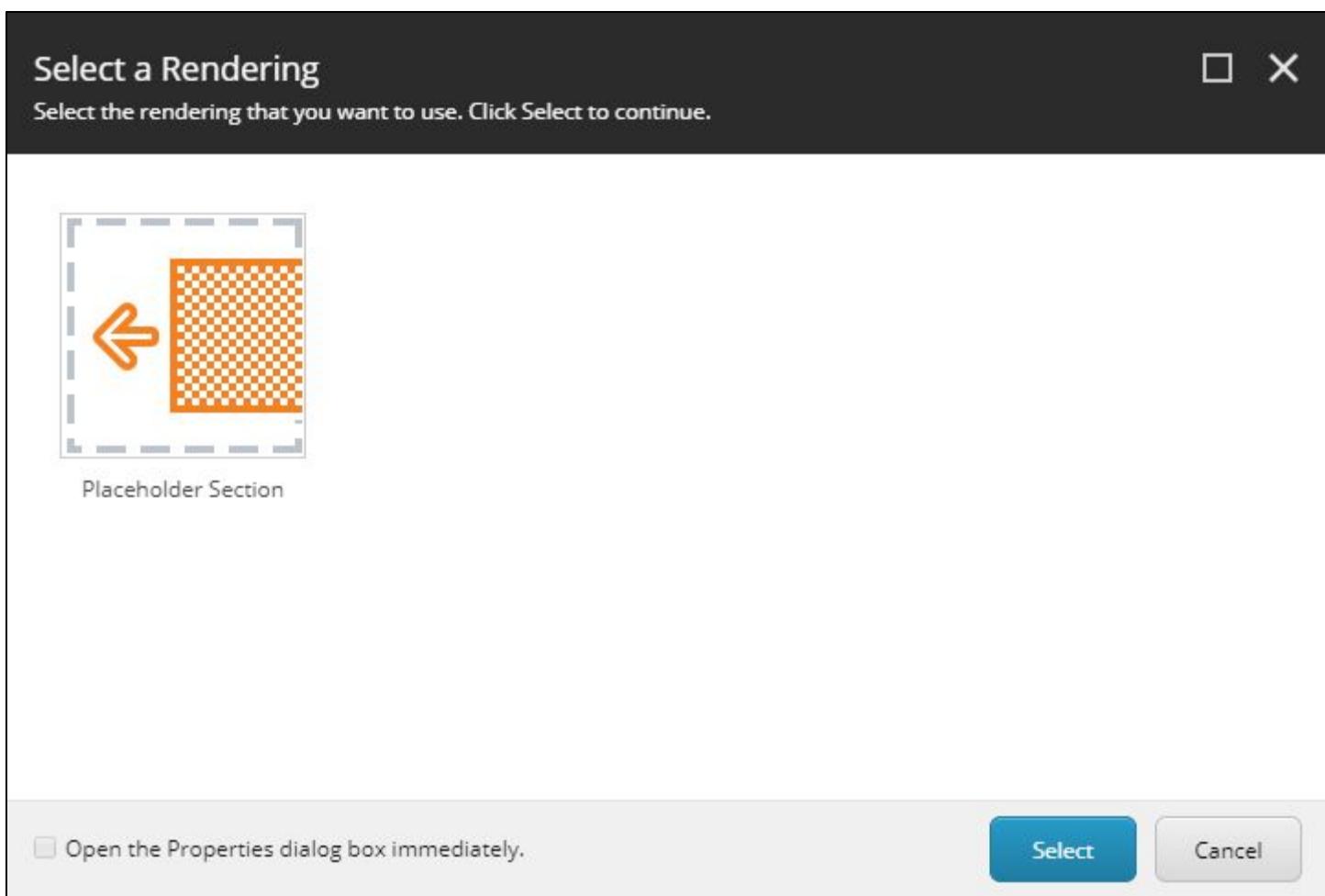
# Main Section



# UI Footer



# Footer Section





## Search Page Structure

# Integrating Search Components in an Existing Layout

A standard Sitecore site will have one or a few (usually 2 or 3 maximum) global page layout(s). In order to preserve the look and feel of the site, using the Coveo Basic Search Interface Layout is not recommended. It is provided as an example only.

Instead, you would add Coveo renderings in an existing layout.

# Creating Search Pages in an Existing Layout

Sitecore does not support rendering template inheritance. For that reason, sites have many rendering templates including the same general renderings for header, footer, navigation, etc.

If you need only one search page:

1. Create a search page item based on a generic rendering template
2. Add the Coveo components in its placeholders

# Creating Search Pages in an Existing Layout

If you need many search pages:

1. Duplicate a generic page rendering template for your search pages
2. Remove useless components and add the Coveo components in the search page template "\_\_Standard Values" item placeholders
3. Create search page items based on your search page rendering template

# Creating Search Pages in an Existing Layout

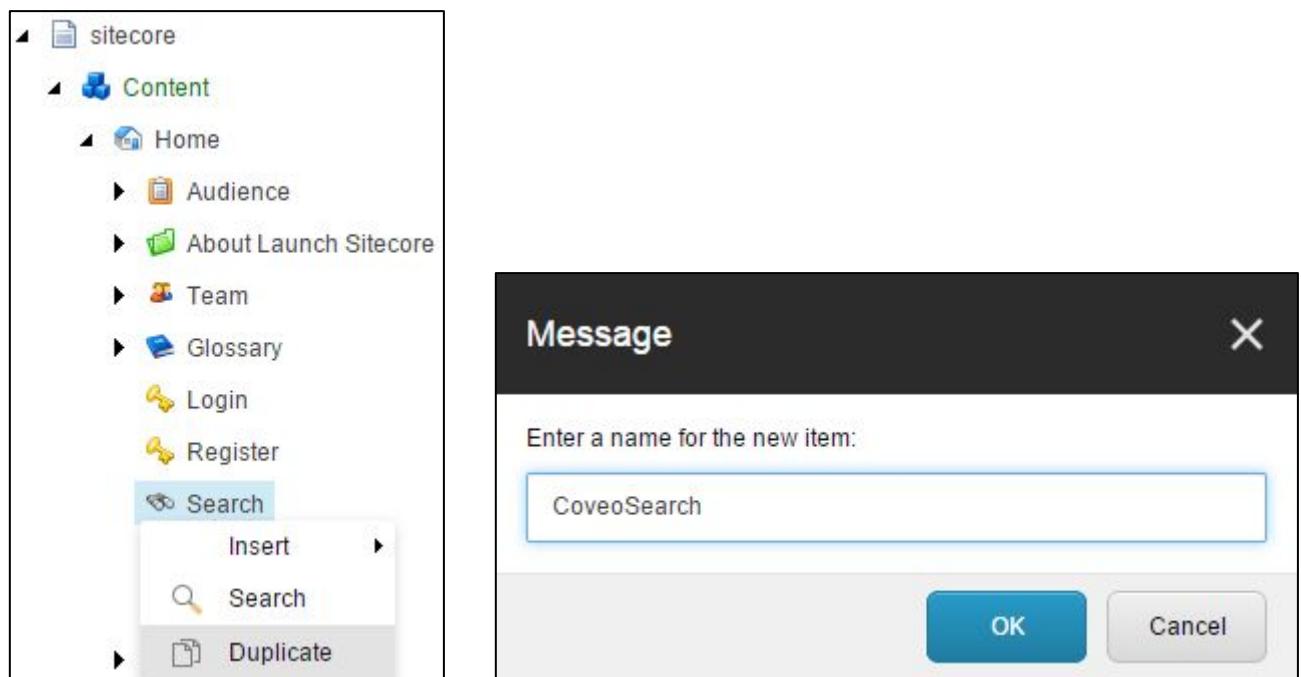
In all cases, the minimal Coveo components to add to display results are:

- Coveo Search Resources
- Coveo Search Interface
- A frame
- Results Section
- Coveo Results List

# Creating a Search Page in Launch Sitecore (1/5)

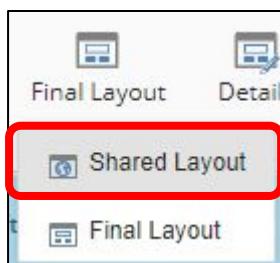
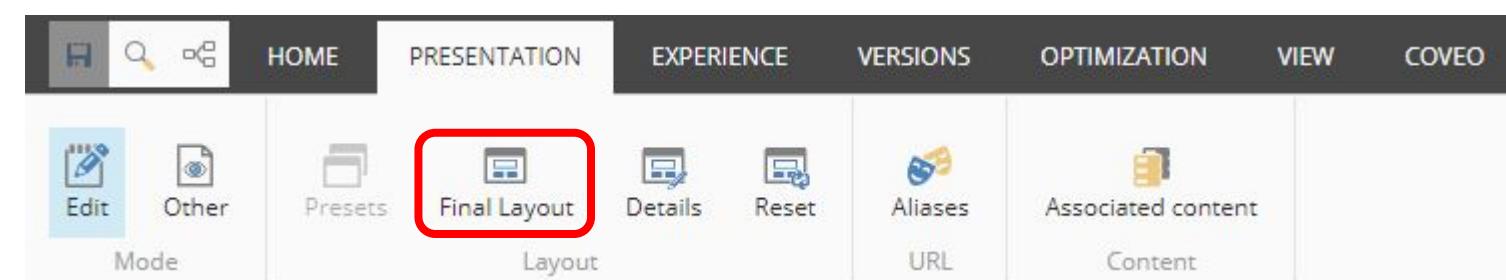
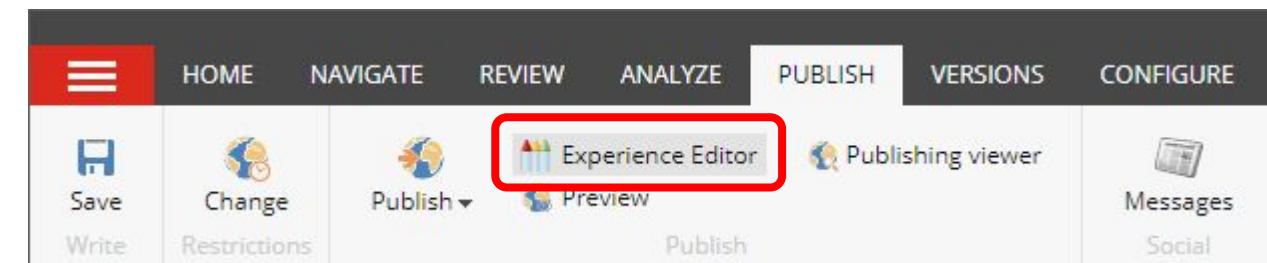
We just need one search page. Launch Sitecore have a "General Page" template. The existing "Search" item is based on that template. Let's duplicate this item and configure our copy.

1. From the Sitecore Content Editor, duplicate the Search content item  
(Right click > Duplicate)
  - a. /sitecore/Content/Home/Search
  - b. New name: CoveoSearch



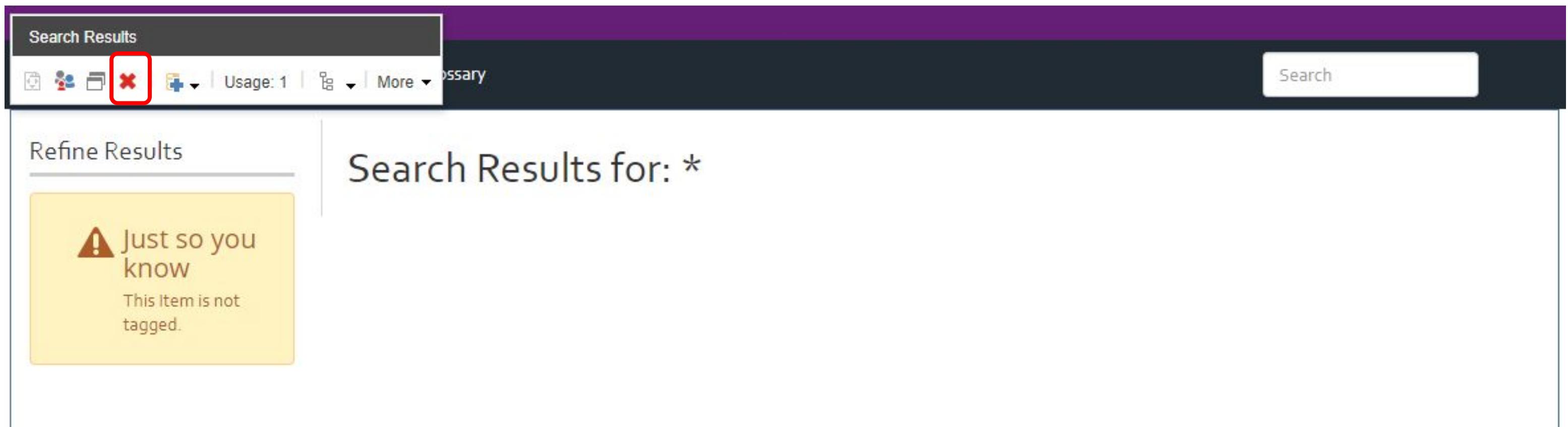
# Creating a Search Page in Launch Sitecore (2/5)

2. Access the Experience Editor (Publish > Experience Editor)
3. Navigate to the Presentation ribbon tab
4. Switch editing to the shared layout
  - a. In the Layout section, click the Final Layout button
  - b. Choose Shared Layout



# Creating a Search Page in Launch Sitecore (3/5)

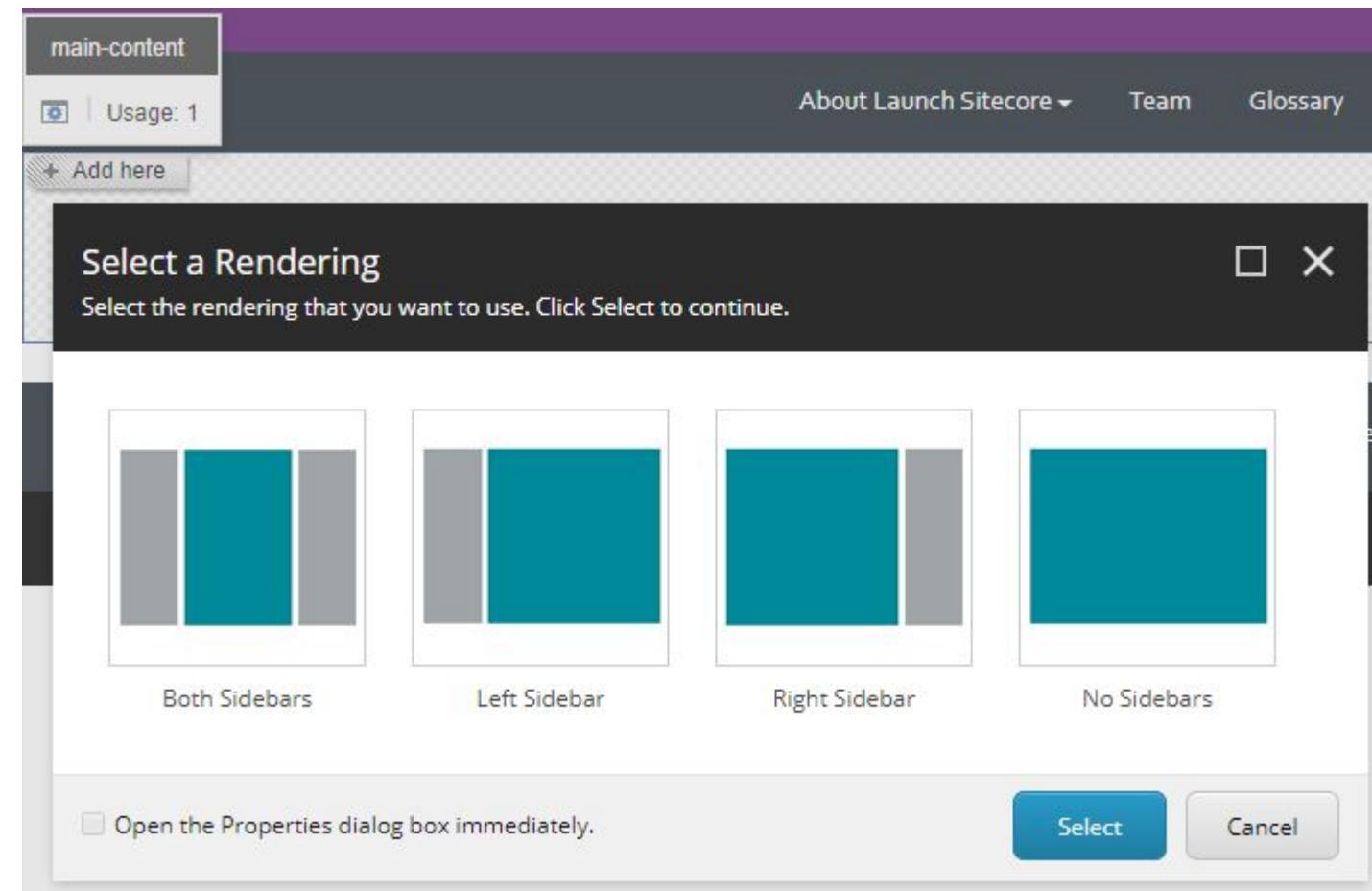
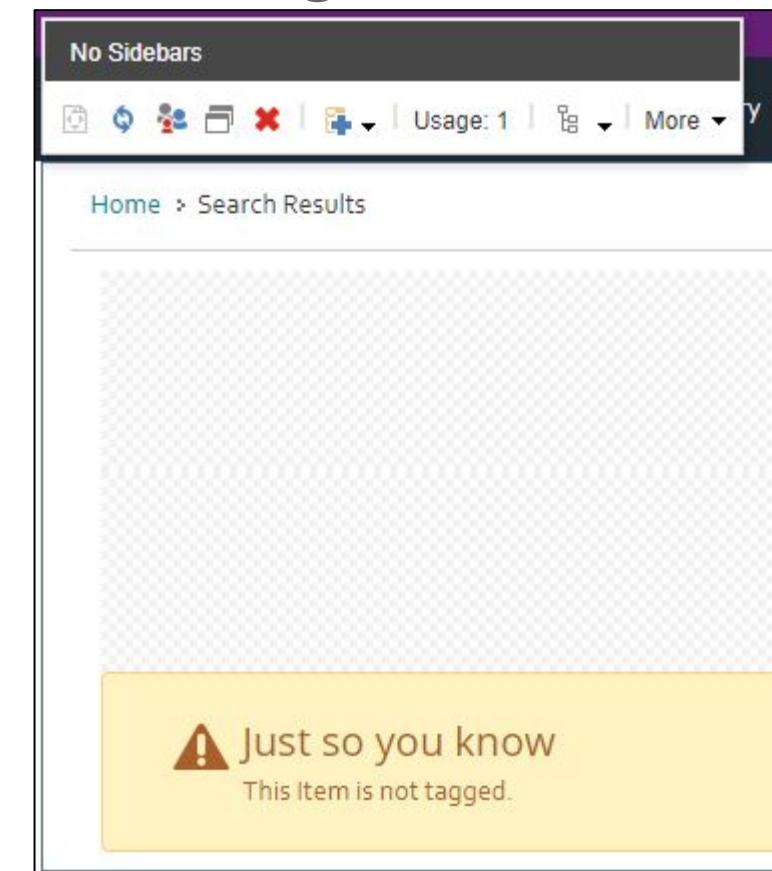
5. Remove the original Search Results component by clicking the Remove component experience button in the floating toolbar



# Creating a Search Page in Launch Sitecore (4/5)

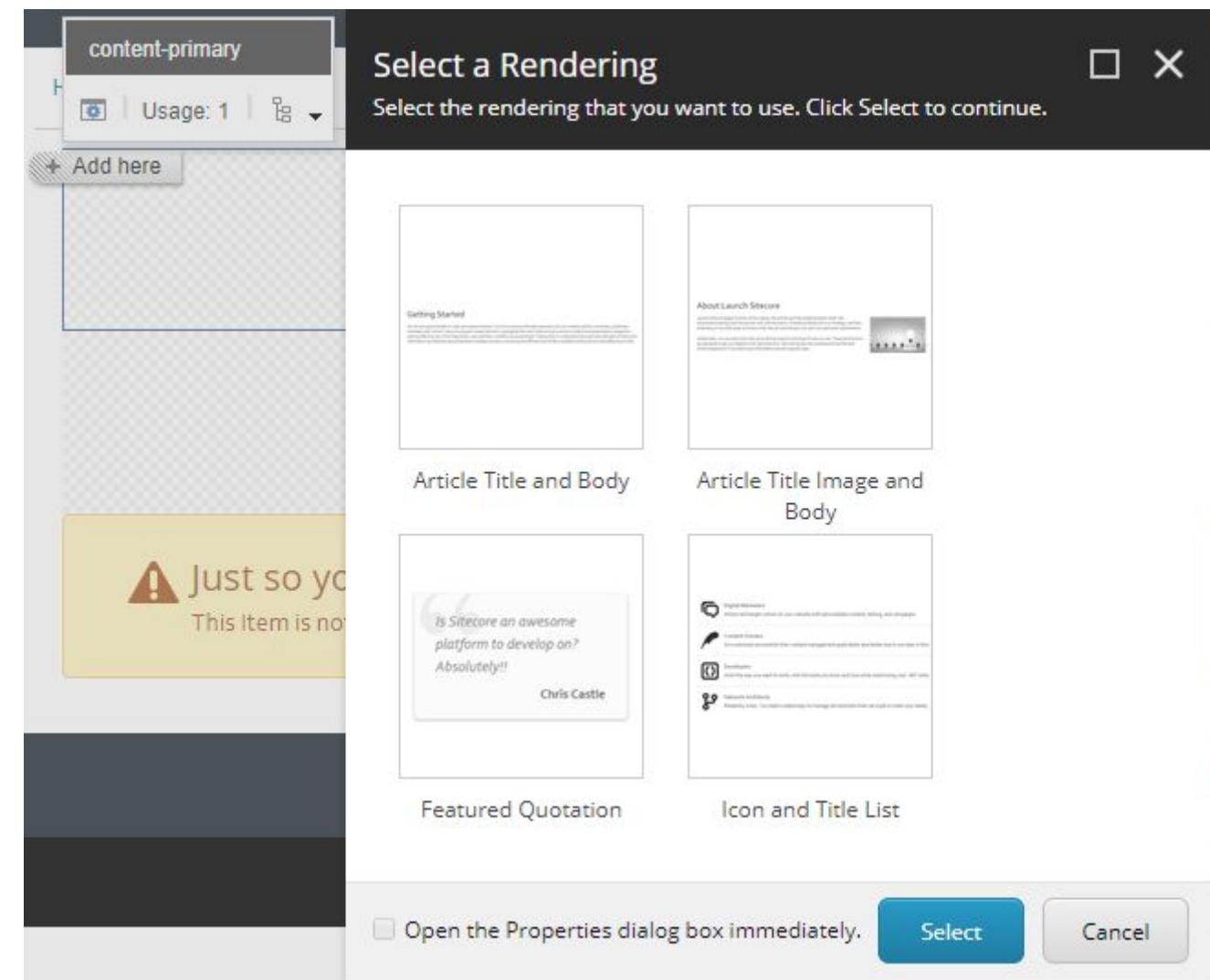
6. Select the main-content placeholder
7. Click the "Add here" button
8. In the Select a Rendering dialog
  - a. Select the No Sidebars rendering
  - b. Choose Select

You now have 2 more placeholders



# Creating a Search Page in Launch Sitecore (5/5)

9. Select the content-primary placeholder
10. Click the "Add here" button
  - a. The insert options on the placeholder prevents inserting a Coveo Search Interface component
11. Choose Cancel
12. Save the item
13. Keep the Experience Editor opened



# Lab

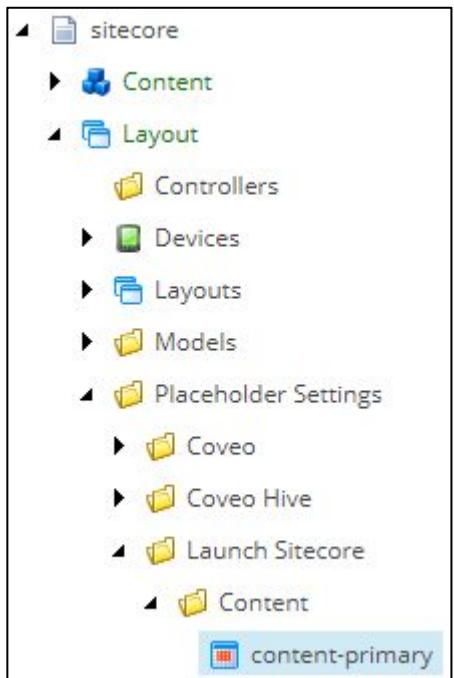
1. Duplicate the Search item
2. Switch the Experience Editor to edit the shared layout
3. Delete the Search Results component
4. Add a No Sidebars component
5. Save the item

# Allowing a Coveo Search Interface in a Placeholder (1/2)

1. From the Sitecore Content Editor, select the content-primary placeholder settings item
  - a. /sitecore/Layout/Placeholder Settings/Launch Sitecore/Content/content-primary
2. In the Allowed Controls field, click the Edit link

The screenshot shows the Sitecore Content Editor's 'Placeholder Settings' interface for the 'content-primary' placeholder. At the top, there's a note about applying settings to matching placeholders with the same key. Below is a text input field containing 'content-primary'. Under 'Allowed Controls [shared]', there's a link to edit them. A red box highlights the 'Edit' link. A list of allowed controls follows, each with a small icon and a URL:

- Article Title and Body [/sitecore/layout/Renderings/LaunchSitecore/Single Item/Article Title and Body]
- Article Title Image and Body [/sitecore/layout/Renderings/LaunchSitecore/Single Item/Article Title Image and Body]
- Featured Quotation [/sitecore/layout/Renderings/LaunchSitecore/Single Item/Featured Quotation]
- Icon and Title List [/sitecore/layout/Renderings/LaunchSitecore/Lists/Icon and Title List]



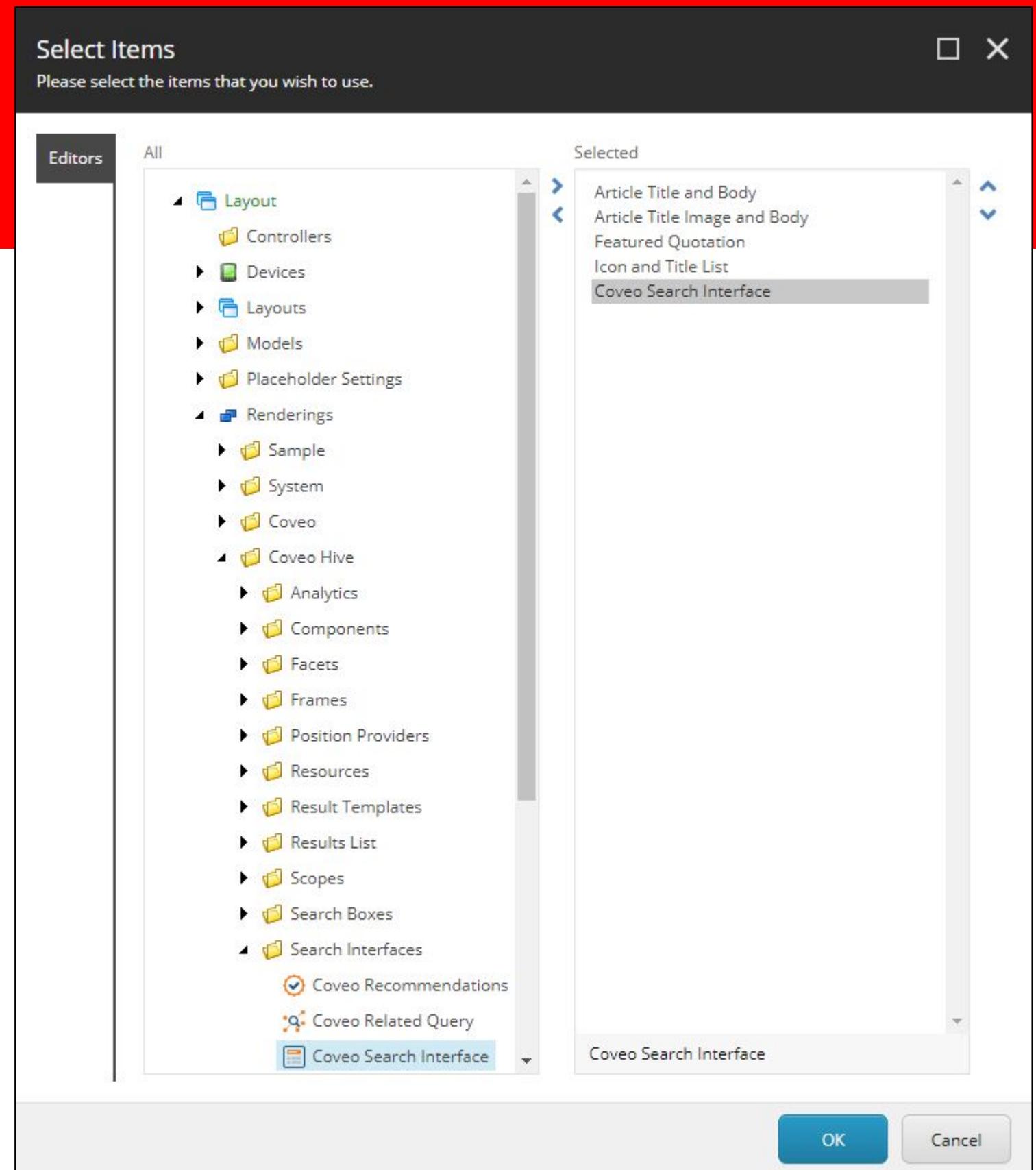
# Allowing a Coveo Search Interface in a Placeholder (2/2)

## 3. In the Select Items dialog

- In the left side (All), double-click the Coveo Search Interface rendering to move it to the right side (Selected)
  - /Layout/Renderings/Coveo Hive/Search Interfaces/Coveo Search Interface

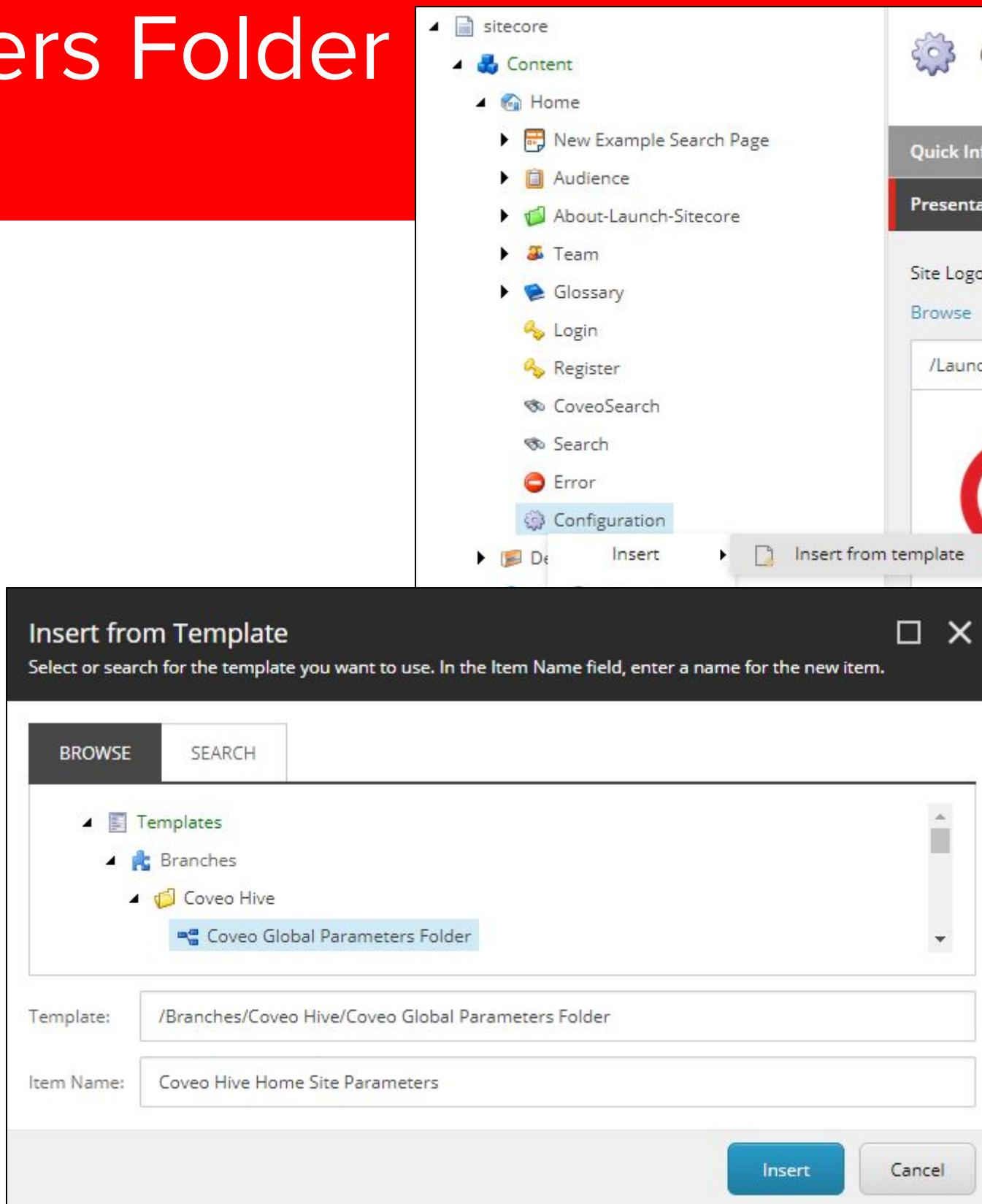
- Choose OK

## 4. Save the item



# Creating a Coveo Global Parameters Folder

1. From the Sitecore Content Editor, select the Configuration content item
  - a. /sitecore/Content/Home/Configuration
2. Insert a new Coveo Global Parameters Folder (right-click > Insert > Insert from template)
3. In the Insert from Template dialog
  - a. Template: /Branches/Coveo Hive/Coveo Global Parameters Folder
  - b. Item Name: Coveo Hive Home Site Parameters
  - c. Choose Insert

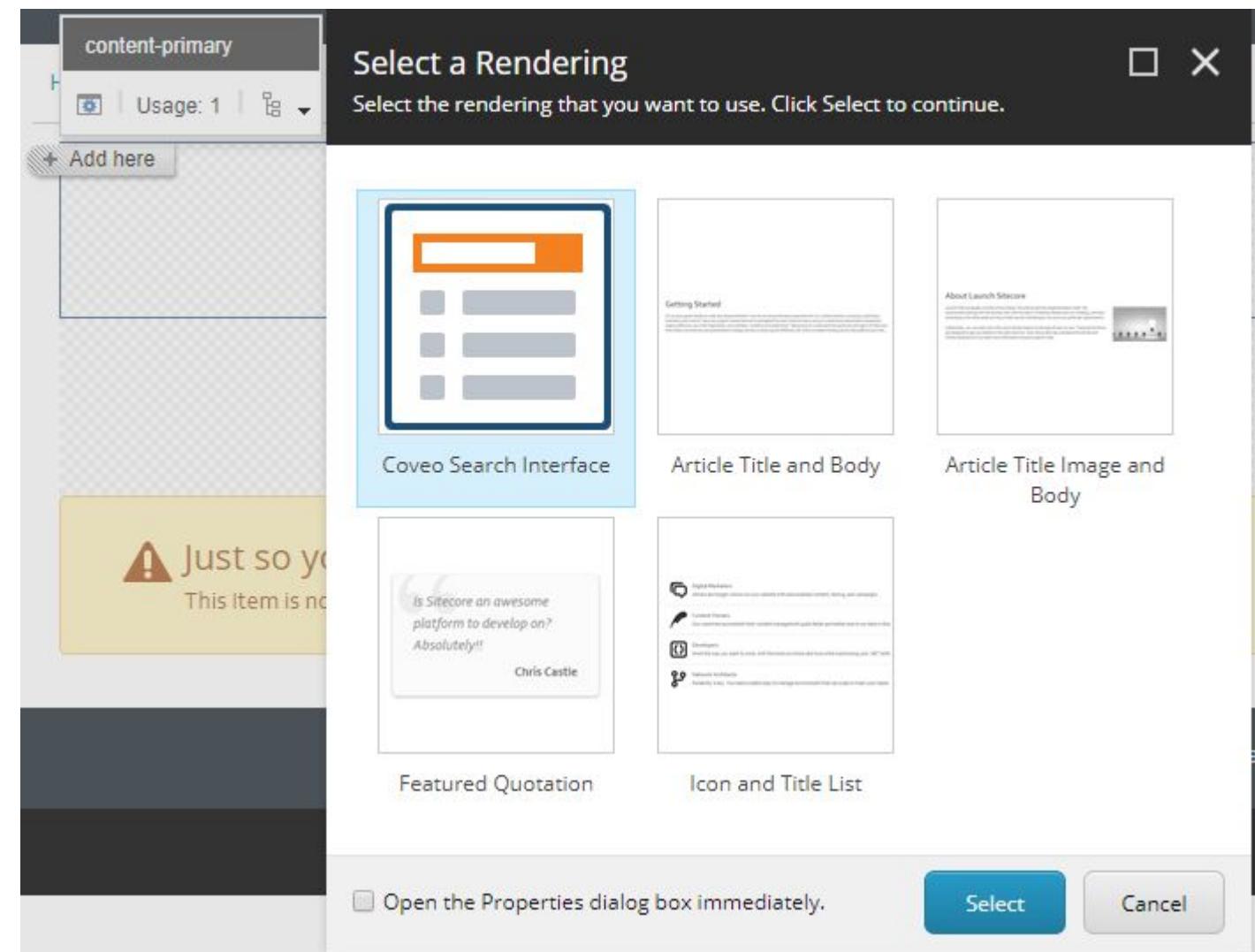


# Lab

1. Allow to insert a Coveo Search Interface in the content-primary placeholder
2. Create a Global Parameters Folder

# Adding Basic Components to a Search Page (1/6)

1. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
2. Access the Experience Editor (Publish > Experience Editor)
3. Select the content-primary placeholder
4. Click the "Add here" button
5. In the Select a Rendering dialog
  - a. Select Coveo Search Interface
  - b. Choose Select

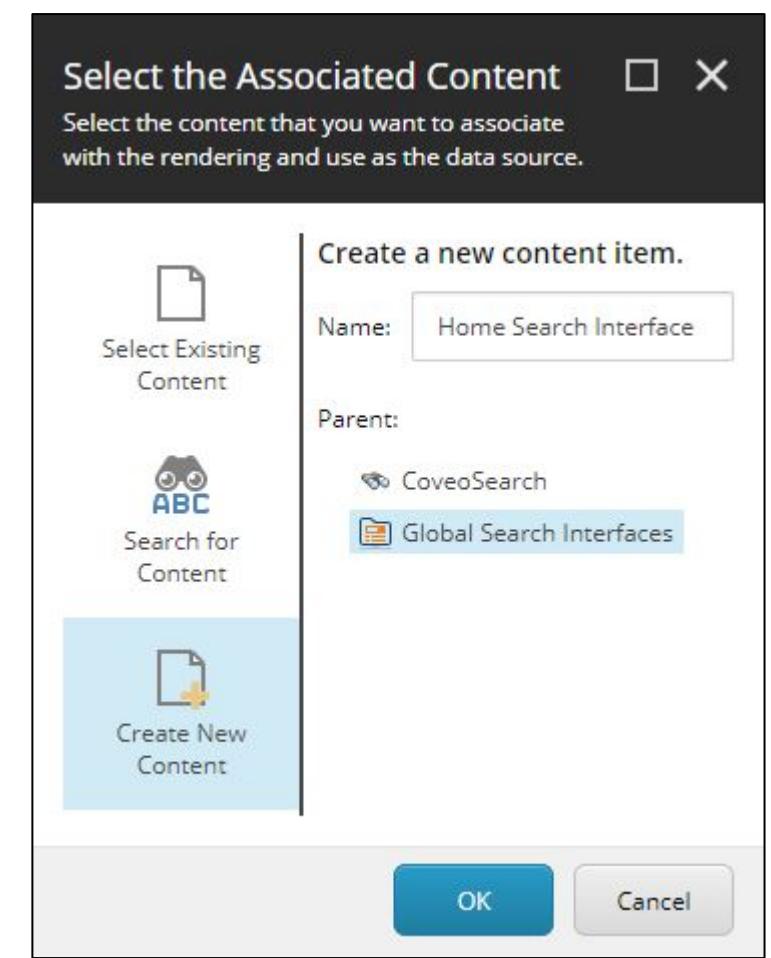


# Adding Basic Components to a Search Page (2/6)

## 6. In the Select the Associated Content dialog

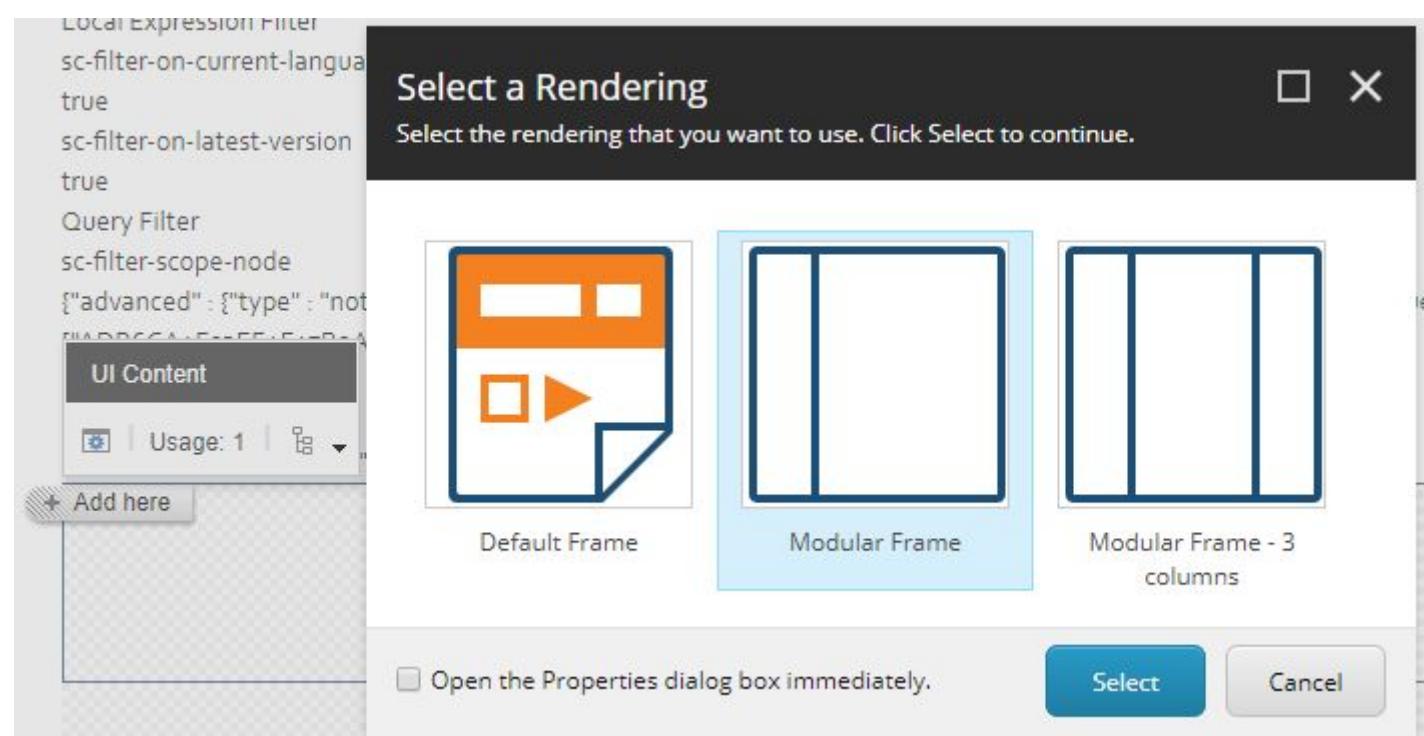
- In the left column, select Create New Content
- Name: Home Search Interface
- Parent: Global Search Interfaces
- Choose OK

## 7. Save the item



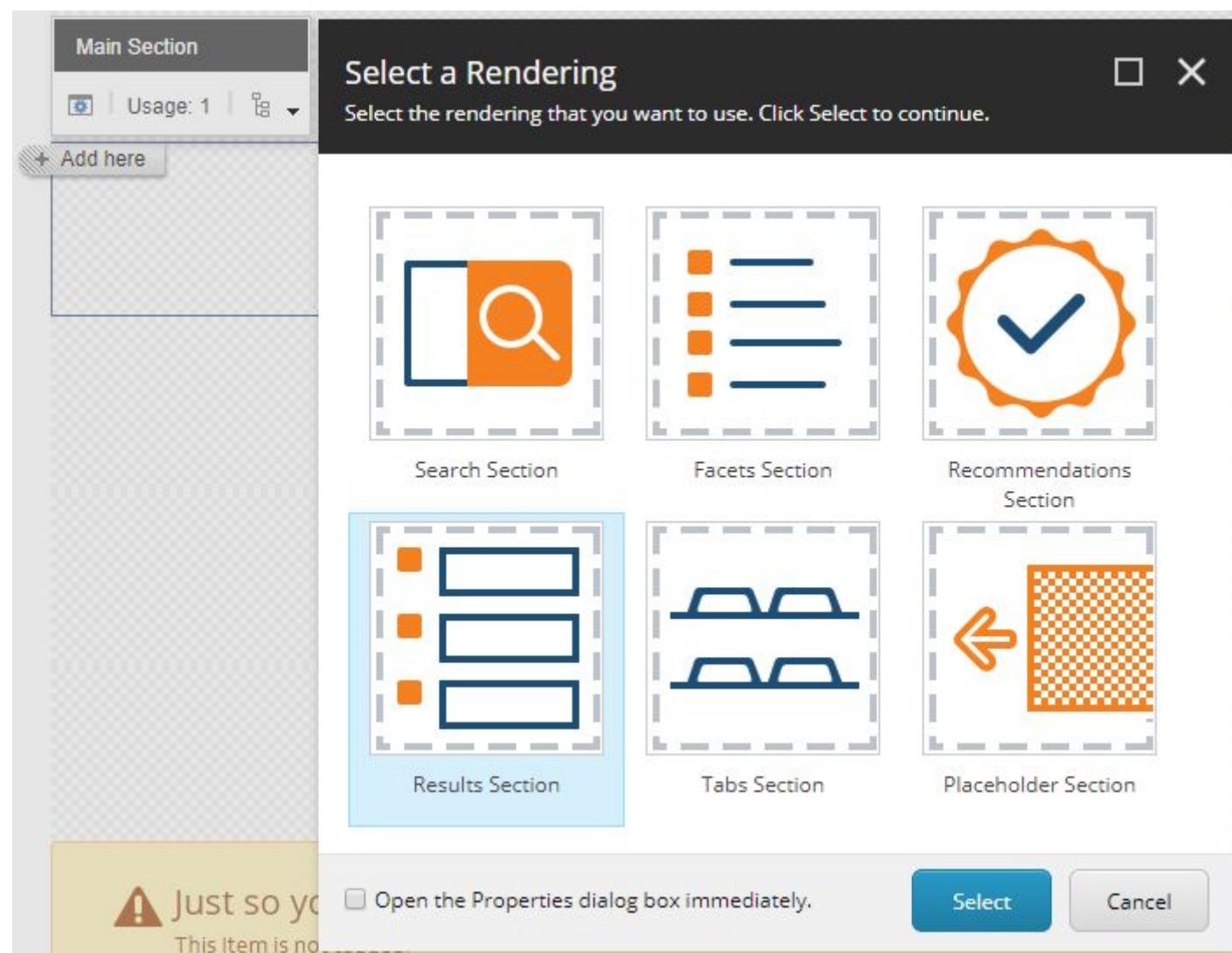
# Adding Basic Components to a Search Page (3/6)

8. Select the UI Content placeholder
9. Click the "Add here" button
10. In the Select a Rendering dialog
  - a. Select Modular Frame
  - b. Choose Select



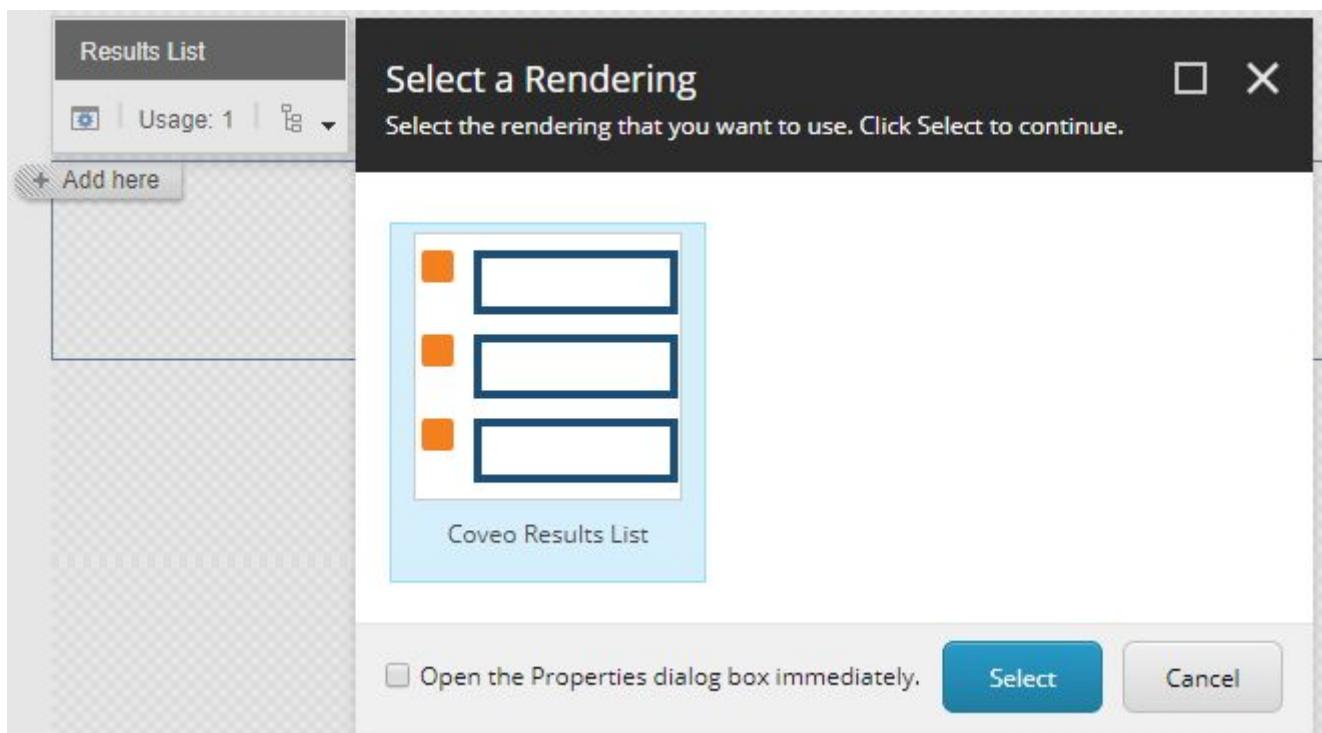
# Adding Basic Components to a Search Page (4/6)

11. Select the Main Section placeholder
12. Click the "Add here" button
13. In the Select a Rendering dialog
  - a. Select Results Section
  - b. Choose Select



# Adding Basic Components to a Search Page (5/6)

14. Select the Results List placeholder
15. Click the "Add here" button
16. In the Select a Rendering dialog
  - a. Select Coveo Results List
  - b. Choose Select

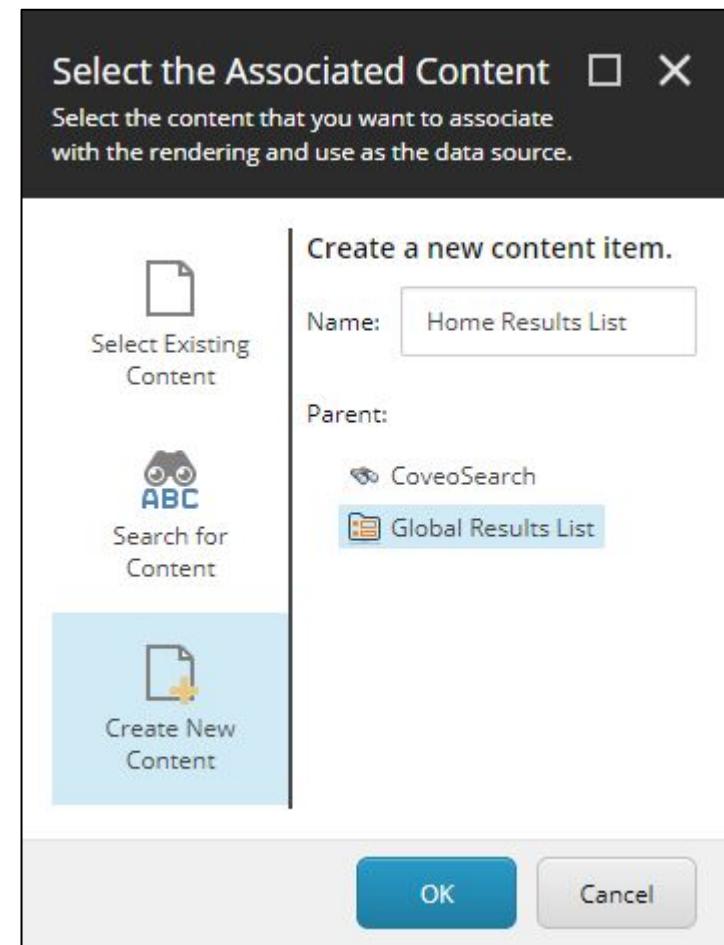
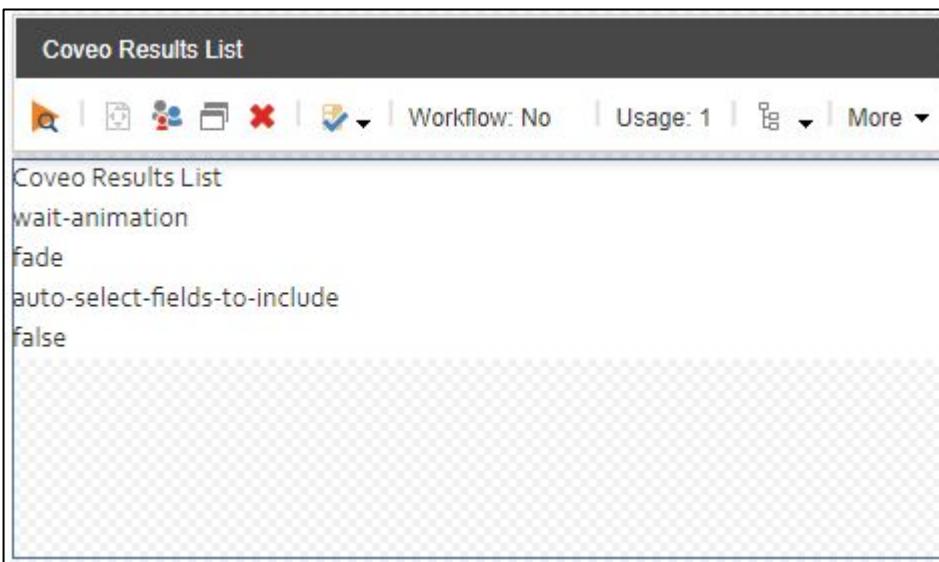


# Adding Basic Components to a Search Page (6/6)

## 17. In the Select the Associated Content dialog

- a. In the left column, select Create New Content
- b. Name: Home Results List
- c. Parent: Global Results List
- d. Choose OK

## 18. Save the item



# The Search Interface is not Working

Why?

# Adding Coveo Search Resources Statically

1. From the file system, open the Launch Sitecore Main layout file
  - a. <your sitecore instance folder>\Website\Views\Shared\Main.cshtml
2. At the end of the file, before </body>, insert the static rendering

```
<script>window.sr = new scrollReveal();</script>
<!-- Coveo Search Resources -->
@Html.Sitecore().Rendering("{41C7C6DD-C132-44BD-890A-B035056BC054}")
</body>
</html>
```

3. Save the file
4. Refresh the Experience Editor
  - a. The search interface is now initializing

Mandatory components (Edit the Search Interface properties to configure them)

The Coveo Analytics are not enabled for this Search Interface. Insert a Coveo Analytics component to record Coveo Usage Analytics data.

Automatic Boosting

Local Expression Filter

Query Filter

Query Ranking

Coveo Results List

[Search Results](#)

[Coveo Hive Placeholder Extenders](#)

[New Example Search Page](#)

## Search Results

[Coveo Hive Placeholder Extenders](#)

[New Example Search Page](#)

## Content

[Basic Presentation Explanation - About Launch Sitecore | Launch Sitecore](#)

You will see a dialog similar to the screenshot below: ... Home ... Audience Landing Pages ... About Launch Sitecore ... Team ... Glossary ... Login ... Copyright © 1999-2015 Sitecore

[Create a Goal - About Launch Sitecore | Launch Sitecore](#)

Using the API to define that a Goal has been achieved ... 1). ... In a real-world scenario, this could be a "Thank You" page where a visitor is sent to ... Home ... Team ... Glossary ... Login

[The Sitecore UIs - About Launch Sitecore | Launch Sitecore](#)

Getting Started ... Submit ... There are a few Page Editor modes, the most common are Edit and Preview. ... Adding page Components ... Inserting new content ... Publishing ... Logging off

[Create a custom validator - About Launch Sitecore | Launch Sitecore](#)

Field Type Level. ... These validators will apply to every field of a certain Sitecore field type (i.e., ... The class itself inherits from the StandardValidator class and includes an Evaluate ...

## Visit Details

[Visit Details](#)

# Component Parameters

The Coveo Hive components have many parameters to affect their behavior.

Parameters are stored in associated data sources. They can be modified from the Content Editor, or from the Coveo experience button in the Experience Editor floating toolbar.



# Lab

- 1.** Add the following components to the search page:
  - a. Coveo Search Interface
    - i. Modular Frame
      1. Results Section
        - a. Coveo Results List
  - 2.** Add the Coveo Search Resources to the main layout
  - 3.** Validate your search page is initializing



# Essential Search Page Components

# Usage Analytics

[Home](#) > [Search Results](#)

[Add to Favorites](#)

Mandatory components (Edit the Search Interface properties to configure them)

The Coveo Analytics are not enabled for this Search Interface. Insert a Coveo Analytics component to record Coveo Usage Analytics data.

Automatic Boosting

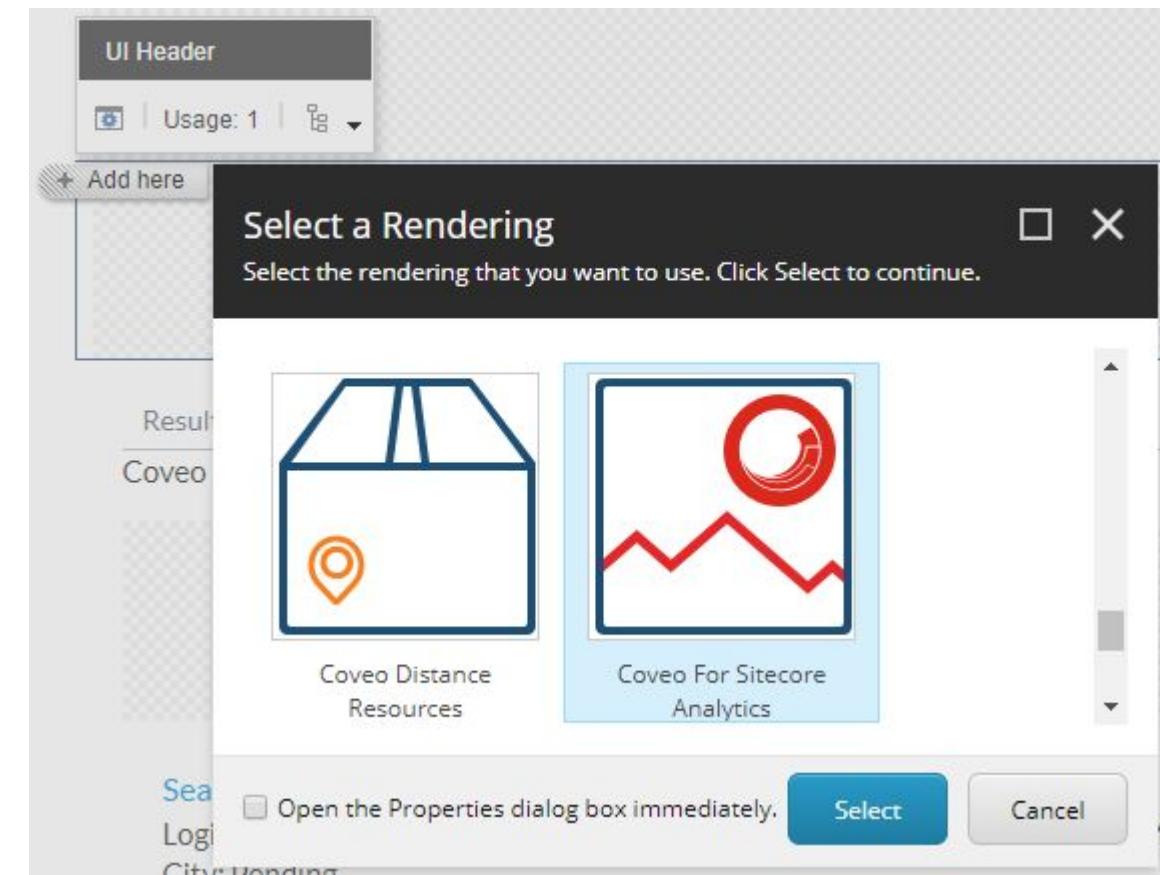
Usage analytics is one of the most important features of a Coveo solution

Without them, it is impossible to continuously and automatically improve the search results relevance and provide intelligent query suggestions through machine learning

It is very important to insert a Coveo Analytics or Coveo for Sitecore Analytics component to your search interfaces

# Adding Analytics to a Search Page (1/2)

1. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
2. Access the Experience Editor (Publish > Experience Editor)
3. Select the UI Header placeholder
4. Click the "Add here" button
5. In the Select a Rendering dialog
  - a. Select Coveo For Sitecore Analytics
  - b. Choose Select

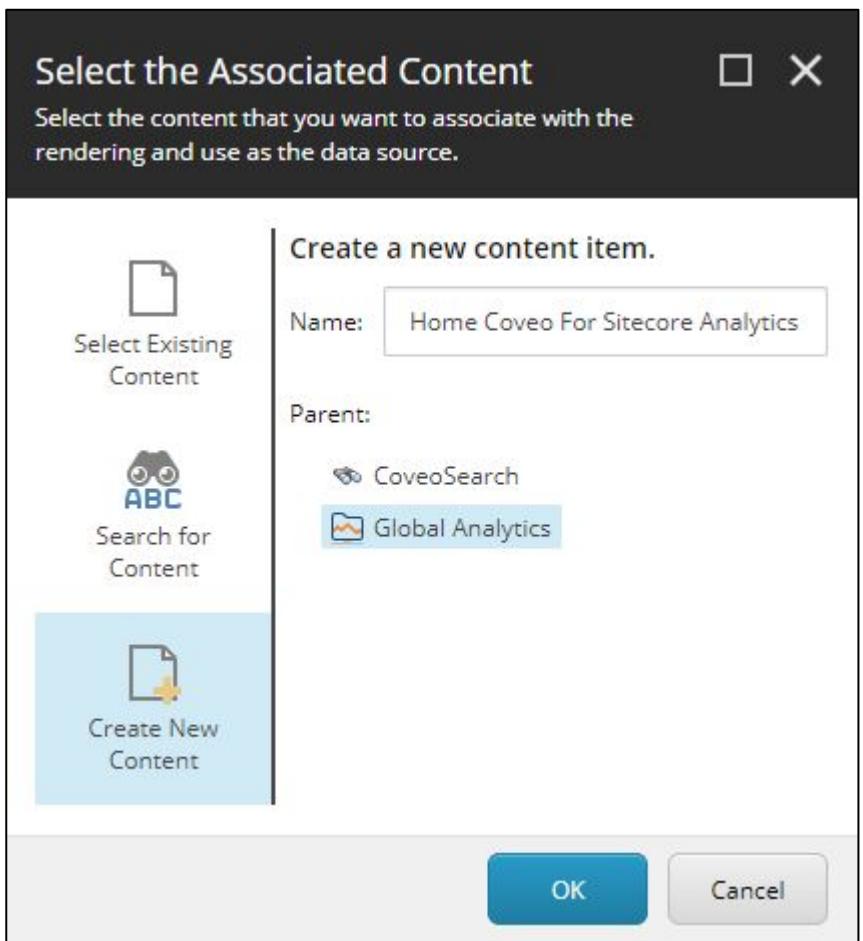
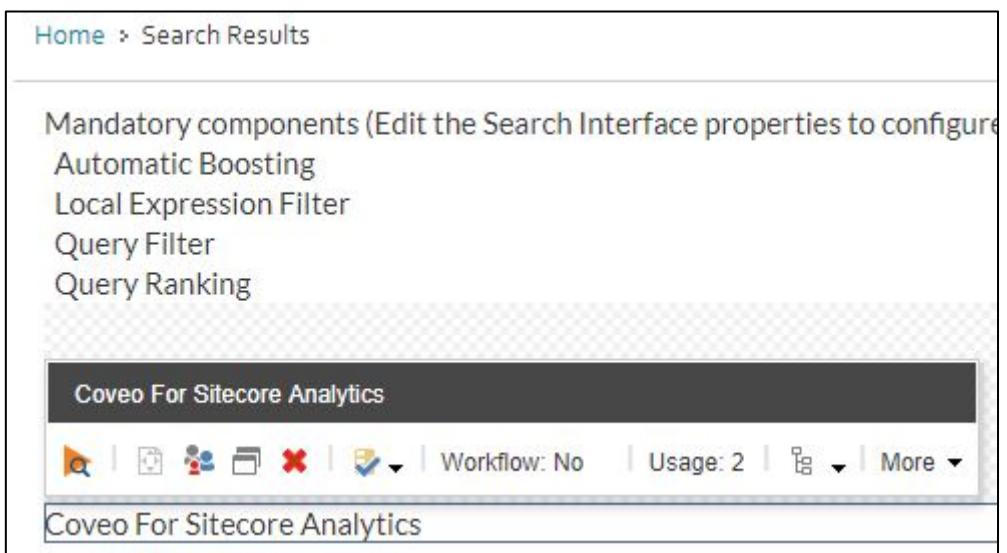


# Adding Analytics to a Search Page (2/2)

## 6. In the Select the Associated Content dialog

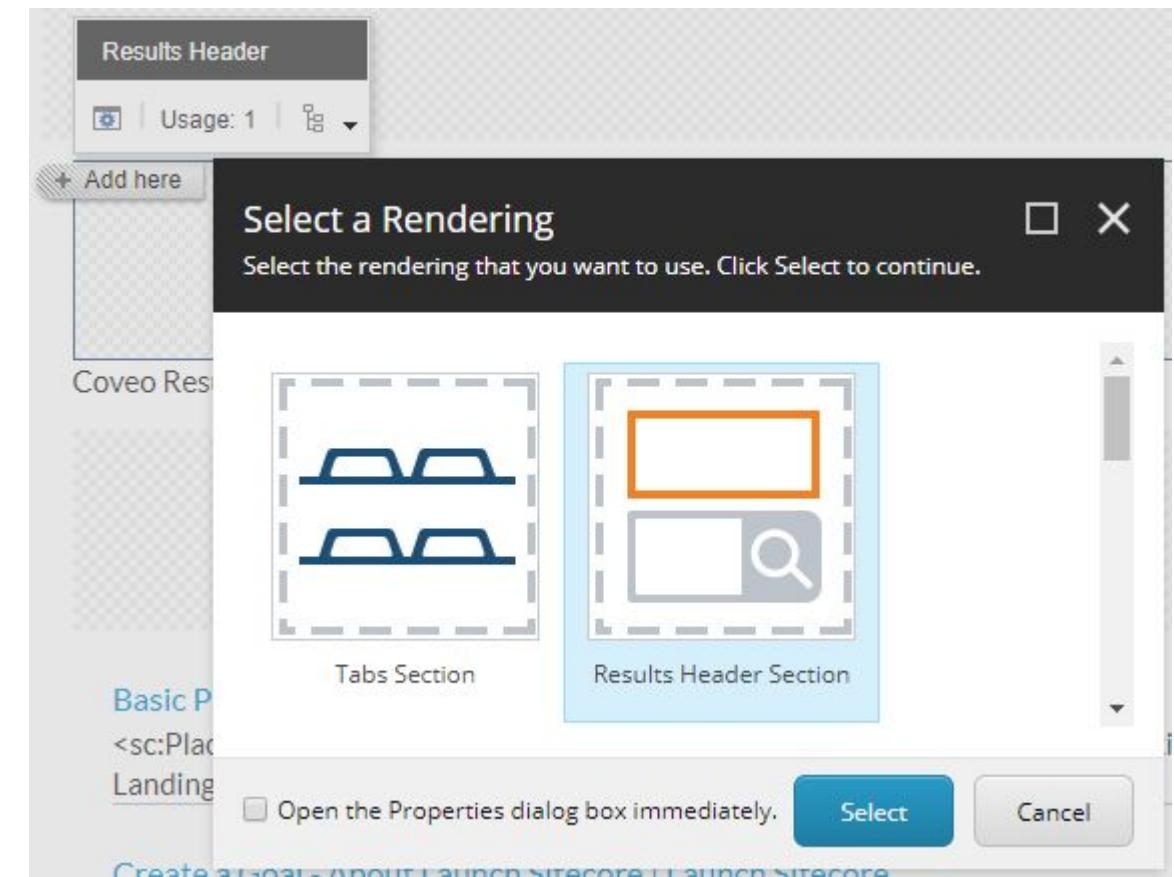
- a. In the left column, select Create New Content
- b. Name: Home Coveo For Sitecore Analytics
- c. Parent: Global Analytics
- d. Choose OK

## 7. Save the item



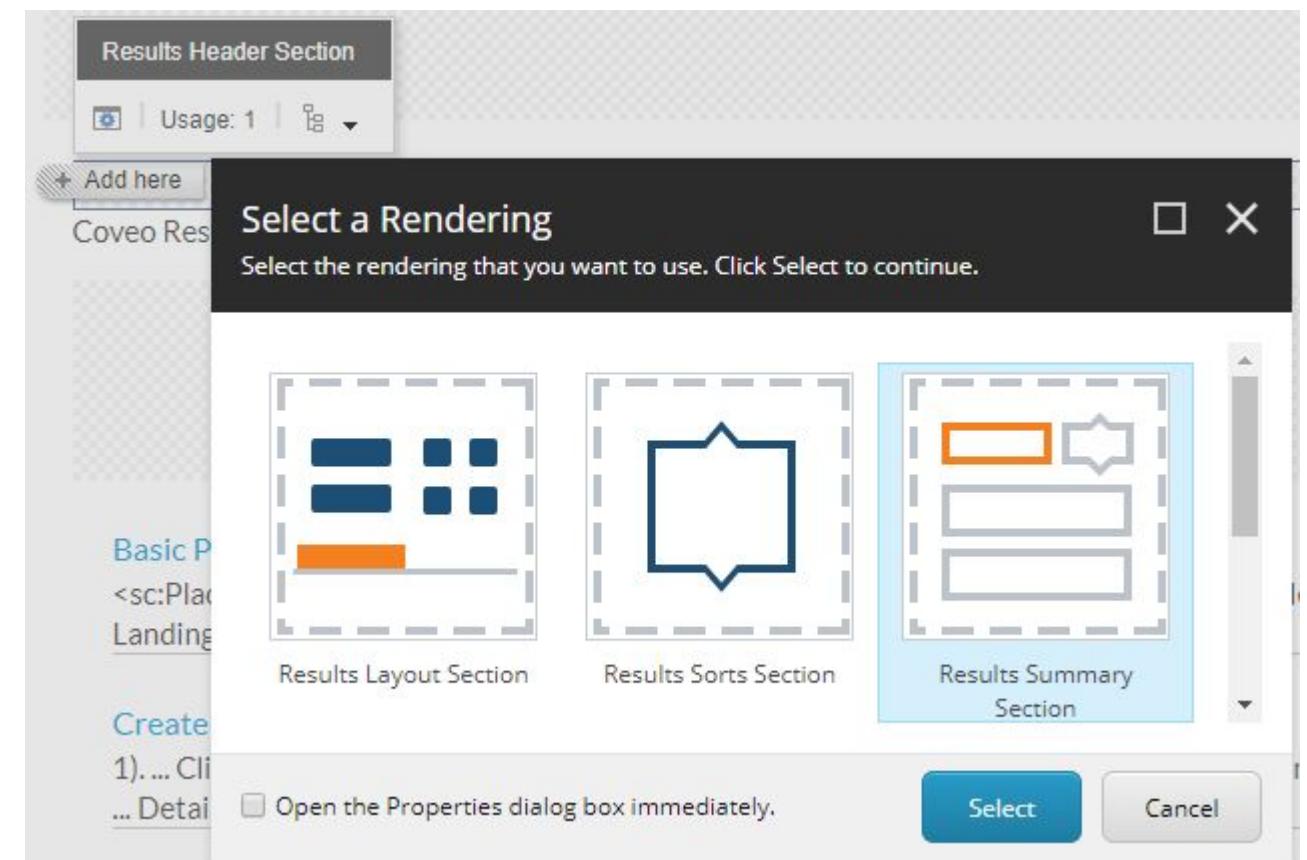
# Adding a Query Summary to a Search Page (1/4)

1. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
2. Access the Experience Editor (Publish > Experience Editor)
3. Select the Results Header placeholder
4. Click the "Add here" button
5. In the Select a Rendering dialog
  - a. Select Results Header Section
  - b. Choose Select



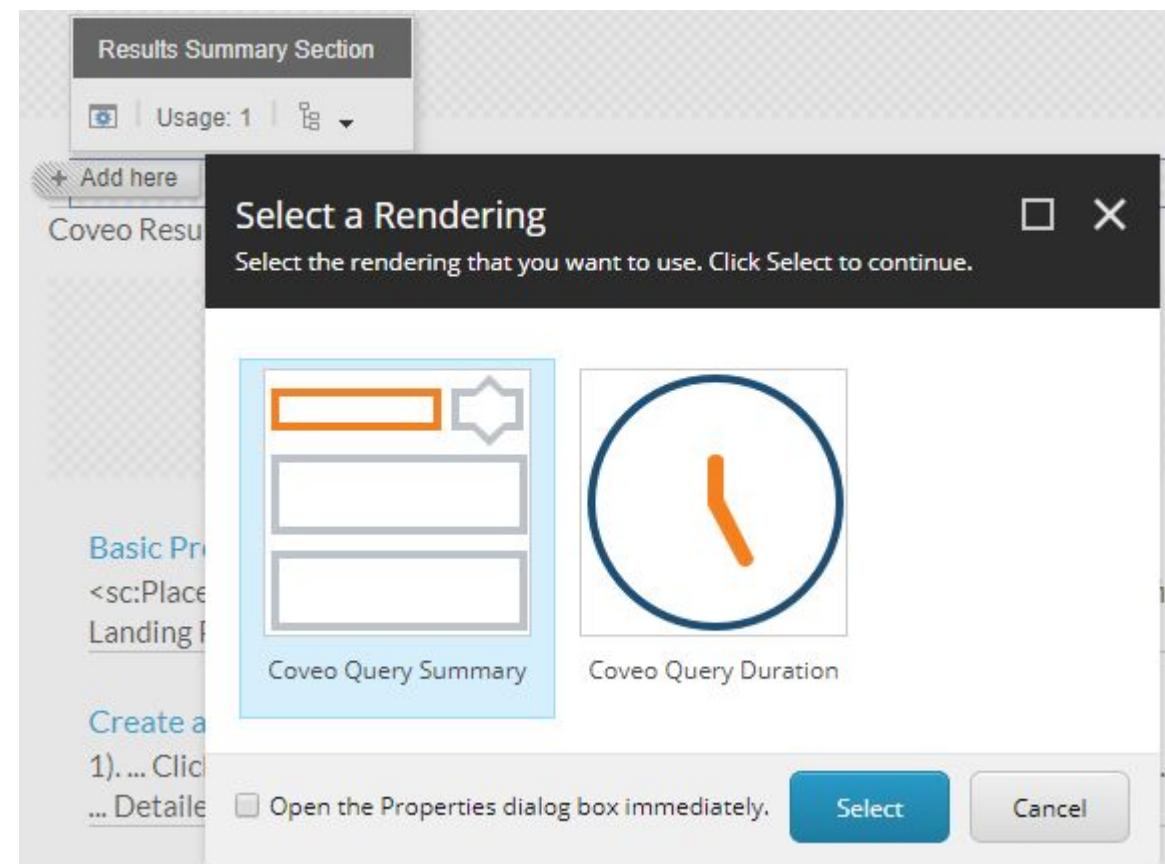
# Adding a Query Summary to a Search Page (2/4)

6. Select the Results Header Section placeholder
7. Click the "Add here" button
8. In the Select a Rendering dialog
  - a. Select Results Summary Section
  - b. Choose Select



# Adding a Query Summary to a Search Page (3/4)

9. Select the Results Summary Section placeholder
10. Click the "Add here" button
11. In the Select a Rendering dialog
  - a. Select Coveo Query Summary
  - b. Choose Select

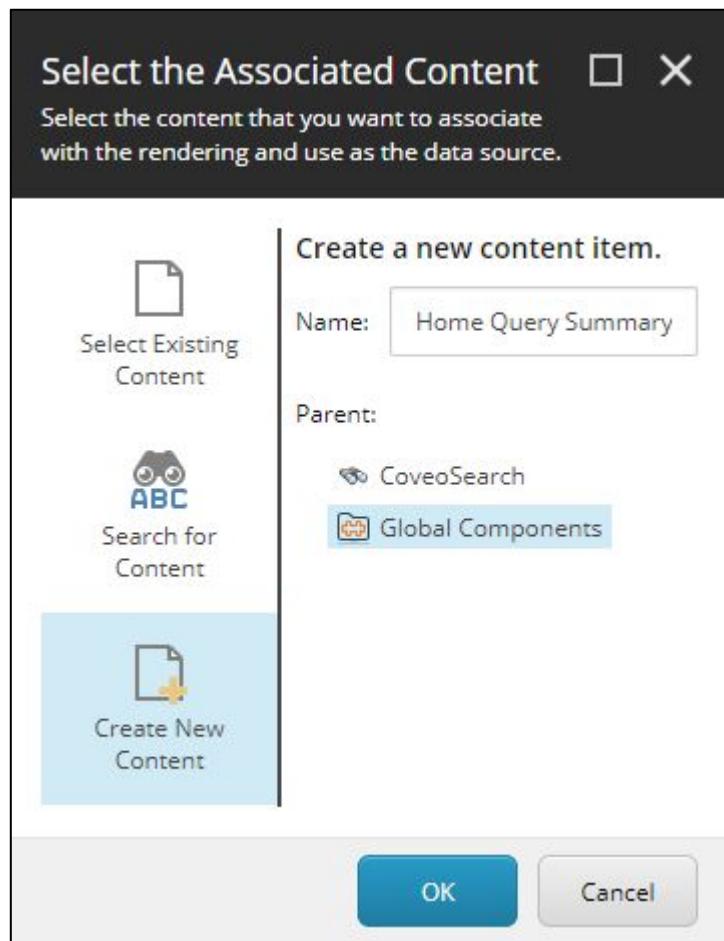
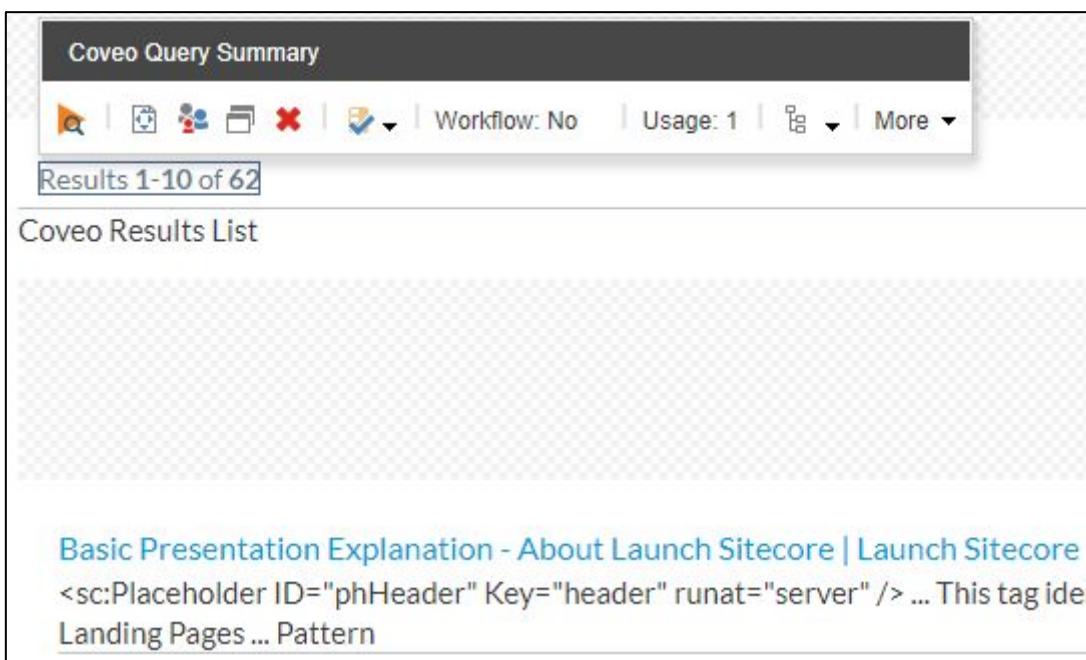


# Adding a Query Summary to a Search Page (4/4)

## 12. In the Select the Associated Content dialog

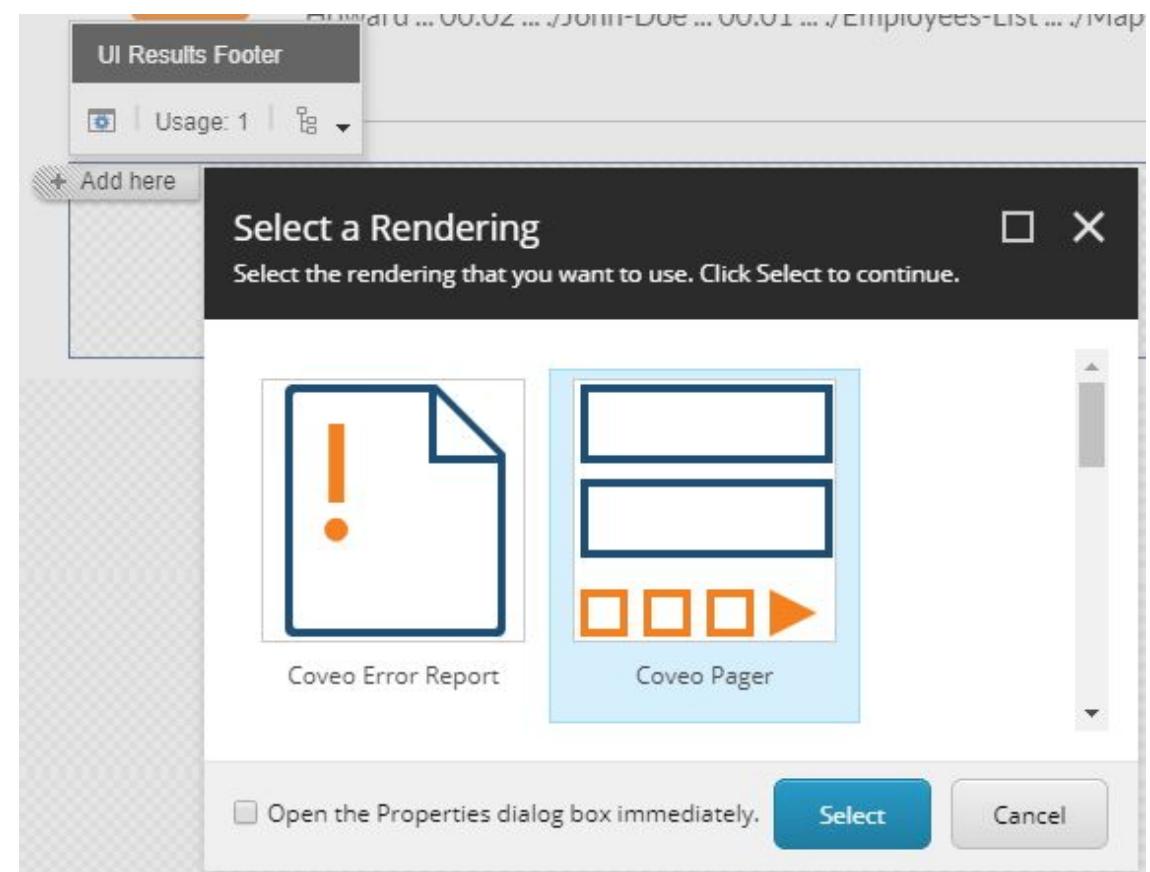
- In the left column, select Create New Content
- Name: Home Query Summary
- Parent: Global Components
- Choose OK

## 13. Save the item



# Adding a Pager to a Search Page (1/2)

1. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
2. Access the Experience Editor (Publish > Experience Editor)
3. Select the UI Results Footer placeholder
4. Click the "Add here" button
5. In the Select a Rendering dialog
  - a. Select Coveo Pager
  - b. Choose Select

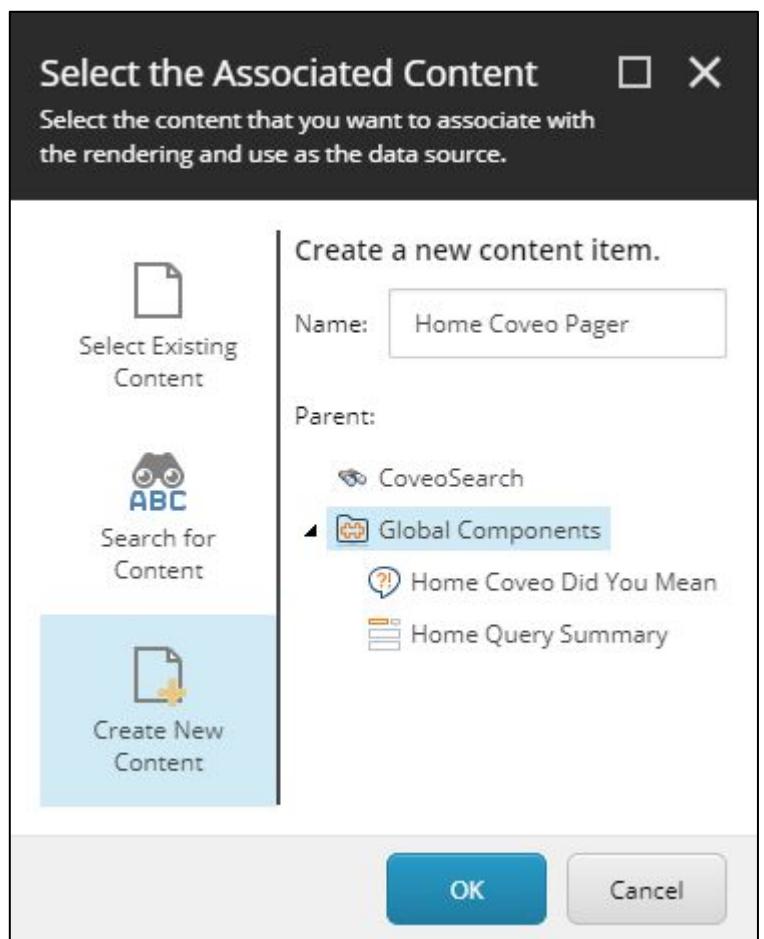
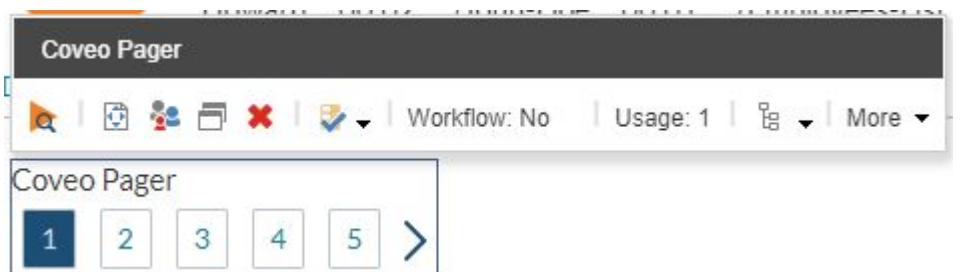


# Adding a Pager to a Search Page (2/2)

## 6. In the Select the Associated Content dialog

- In the left column, select Create New Content
- Name: Home Coveo Pager
- Parent: Global Components
- Choose OK

## 7. Save the item



# Lab

1. Add the following components to the search page:
  - a. Coveo For Sitecore Analytics
  - b. Results Header Section
    - i. Results Summary Section
      1. Coveo Query Summary
  - c. Coveo Pager

# Publishing the Search Page

- 1.** From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
- 2.** Submit the item for review (Review > Submit)
  - a. In the dialog, click OK button
- 3.** Approve the item (Review > Approve without Test)
  - a. In the dialog, click OK button

# Validating Changes

1. In another browser or an incognito window, open your published search page
  - a. <http://YourInstanceName/CoveoSearch>
2. Validate that the published result page is behaving the same way as the preview result page
  - a. You should see 101 search results

# The Published Search Page is not Working

Why?

# Publishing the Site

1. From the Sitecore Content Editor, smart publish the site  
(Publish tab > Publish dropdown button > Publish Site option > Smart publish radiobutton > Publish button)
2. In another browser or an incognito window, open your published search page
  - a. <http://YourInstanceName/CoveoSearch>
3. Validate that the published result page is behaving the same way as the preview result page
  - a. You should see 101 search results

[Home](#) > Search Results

Results 1-10 of 101

[Search Results | Launch Sitecore](#)

About Launch Sitecore ... Team ... Glossary ... Login ... Copyright © 1999-2015 Sitecore ... Visit Details 10 ... Pattern ... No Match ... Unknown ... Visit is not associated with a campaign.

[Home | Launch Sitecore](#)

Reliability is key. ... Learn more » ... Getting Started with Launch Sitecore ... Get Started ... Home ... Audience Landing Pages ... About Launch Sitecore ... Team ... Glossary ... Login

[Content](#)[Basic Presentation Explanation - About Launch Sitecore | Launch Sitecore](#)

<sc:Placeholder ID="phHeader" Key="header" runat="server" /> ... This tag identifies the Header Placeholder region in the wireframe above. ... Home ... Audience Landing Pages ... Pattern

[Create a Goal - About Launch Sitecore | Launch Sitecore](#)

1) ... Clicking on the Goals button will bring up a checklist that shows all currently ... A screenshot of one of the Wizard steps from WFFM is below: ... Team ... Pattern ... Detailed

[The Sitecore UIs - About Launch Sitecore | Launch Sitecore](#)

Adding page Components ... Logging off ... Language: View the page content in another language/locale ... Of course, all these views are independent, so you can choose to view your page in a ...

[Create a custom validator - About Launch Sitecore | Launch Sitecore](#)

Field Type Level. ... The screenshot below shows the Required validator and the field that calls out the ... Create an item to represent the validator (as the Required validator does) ... Team

[Visit Details](#)[Visit Details](#)[Documentation](#)

# Anatomy of a Basic Coveo for Sitecore Query

## Advanced Query:

```
(  
  Item Templates Exclusion from the Properties Editor Options  
)  
(  
  Automatic Source Filter  
)
```

## Constant Query:

Various Filters from the Properties Editor Options

# Lab

1. Submit and approve the item
2. Smart publish your site
3. Test the page in published mode
4. Inspect the queries sent by the Coveo JavaScript Search Framework

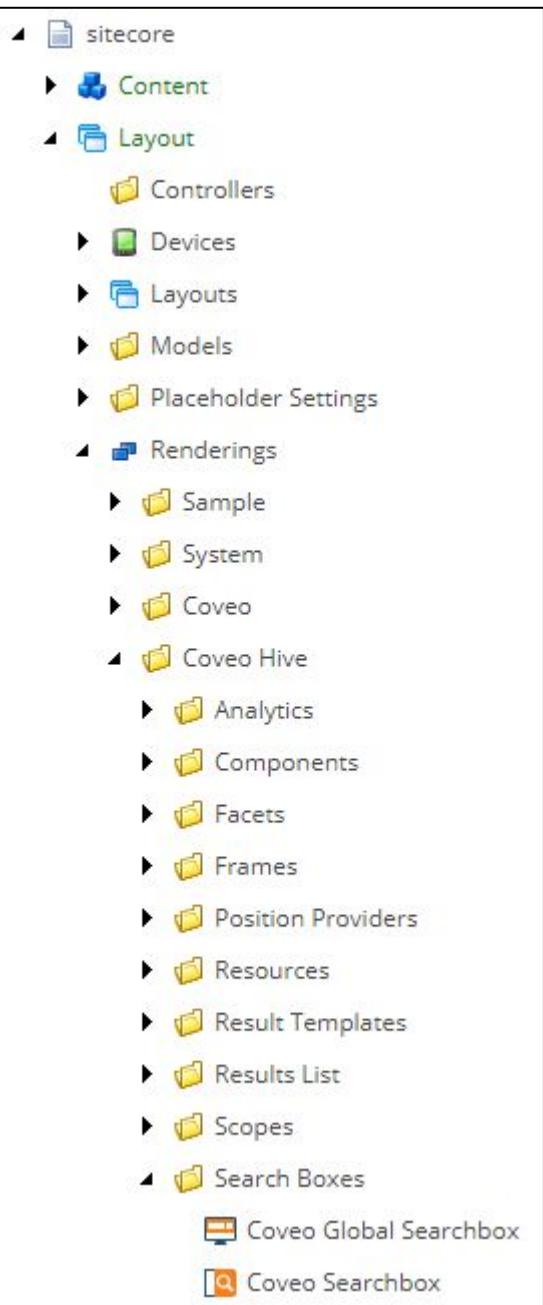


## Global Searchbox

# Searchbox Component

There are 2 searchbox components:

- Coveo Searchbox
  - Only for the search page
- Coveo Global Searchbox
  - Redirects to a search page
  - For all pages, including the search page
  - Requires the Coveo Search Resources rendering



# Adding a Placeholder for a Global Searchbox

Launch Sitecore has its searchbox statically defined in its header view rendering. We need to replace it by a placeholder for the Coveo searchbox.

- 1.** From the file system, open the header view
  - a. <your sitecore instance folder>\Website\Views\Navigation\Header.cshtml
- 2.** Look for the search box at the end of the file

```
@Html.Sitecore().Rendering("{4E4C8D75-5D59-4591-8B57-8402DBD28DF0}") <!-- Search Box -->
```

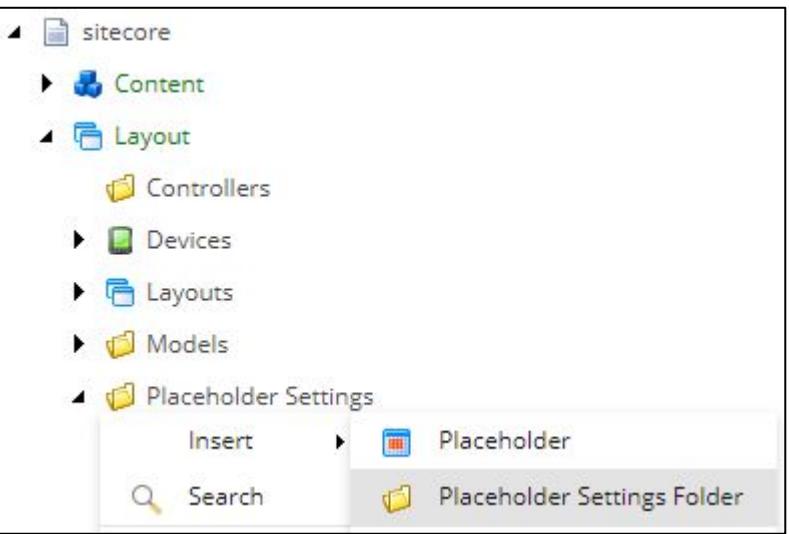
- 3.** Replace it by a placeholder

```
<!-- Search Box -->
<div class="navbar-right" style="width: 400px;">
    @Html.Sitecore().Placeholder("search-box")
</div>
```

- 4.** Save the file

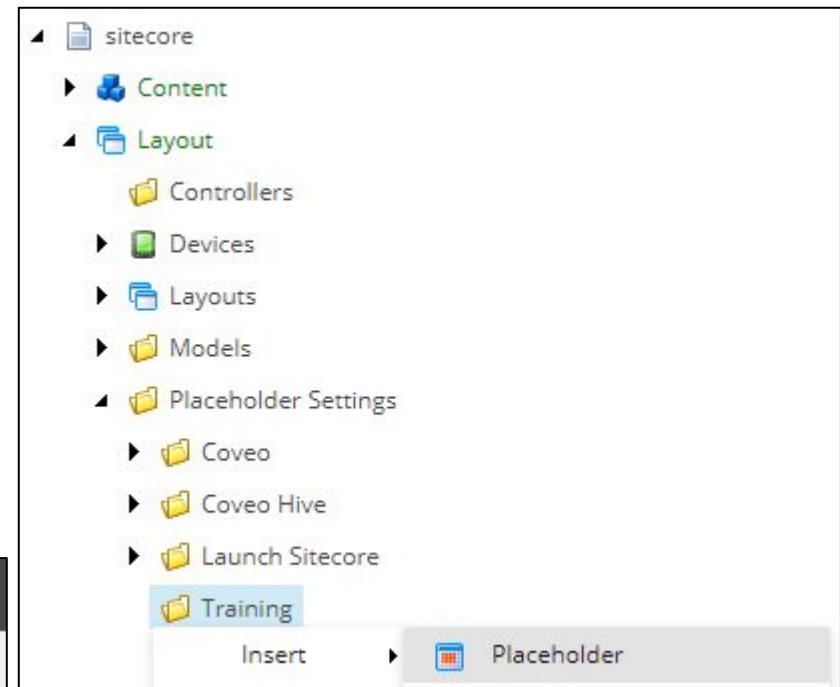
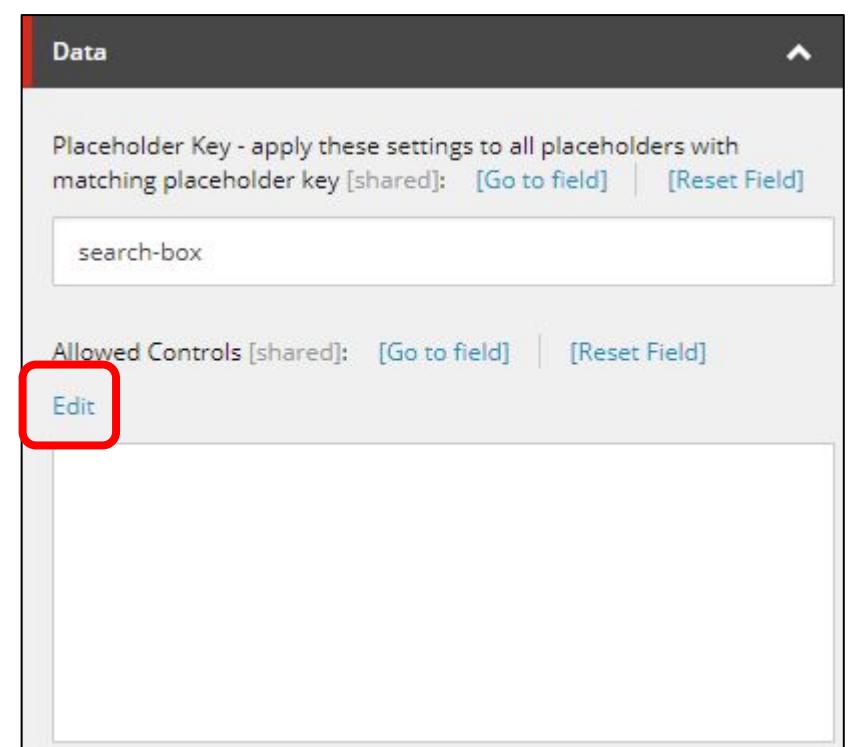
# Creating Placeholder Settings (1/3)

1. From the Sitecore Content Editor, select the Placeholder Settings folder
  - a. /sitecore/Layout/Placeholder Settings
2. Insert a new Placeholder Settings Folder  
(right-click > Insert > Placeholder Settings Folder)
  - a. Name: Training



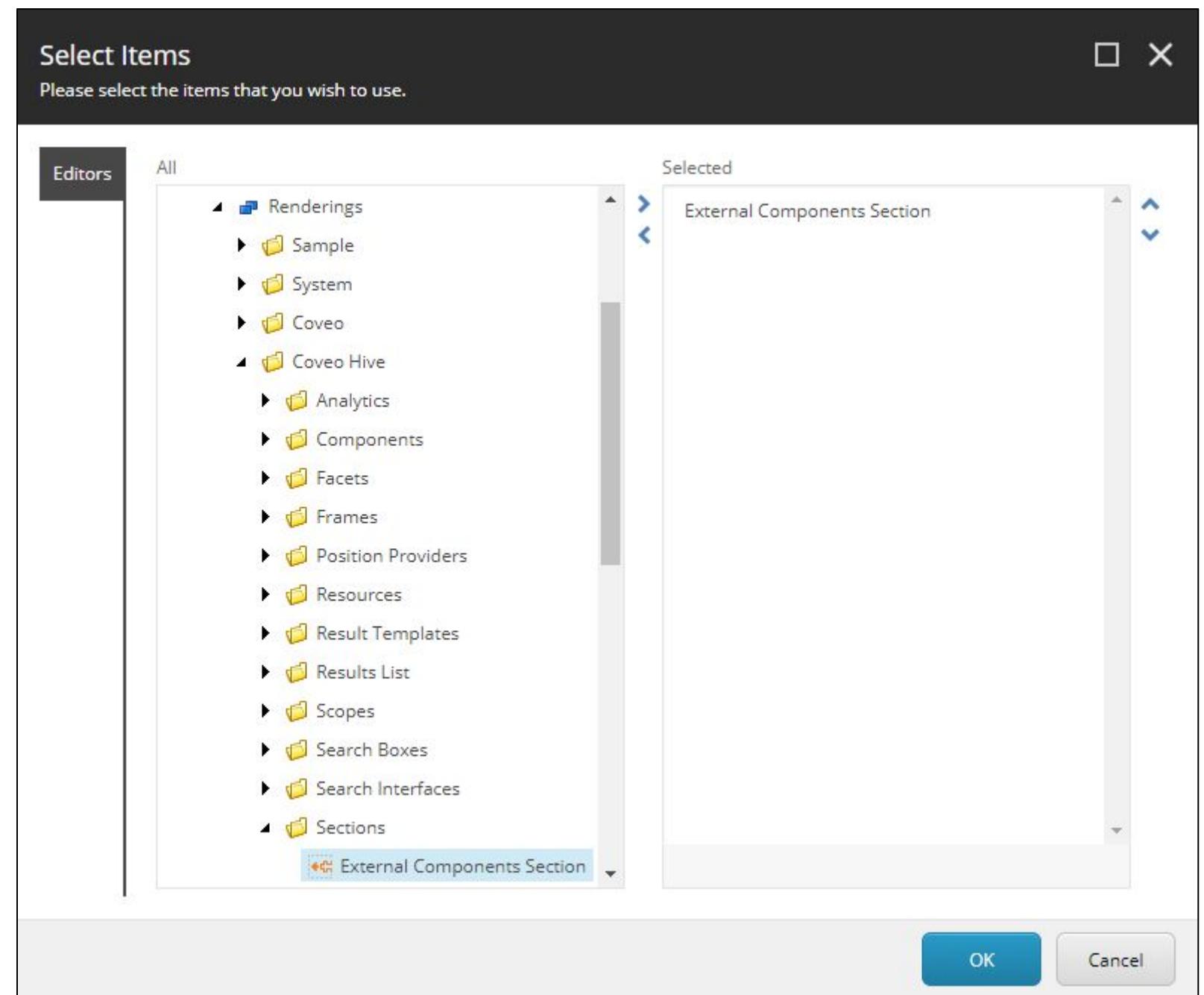
# Creating Placeholder Settings (2/3)

3. In this new folder, insert a new Placeholder item (right-click > Insert > Placeholder)
  - a. Name: search-box
4. In the Allowed Controls field, click the Edit link



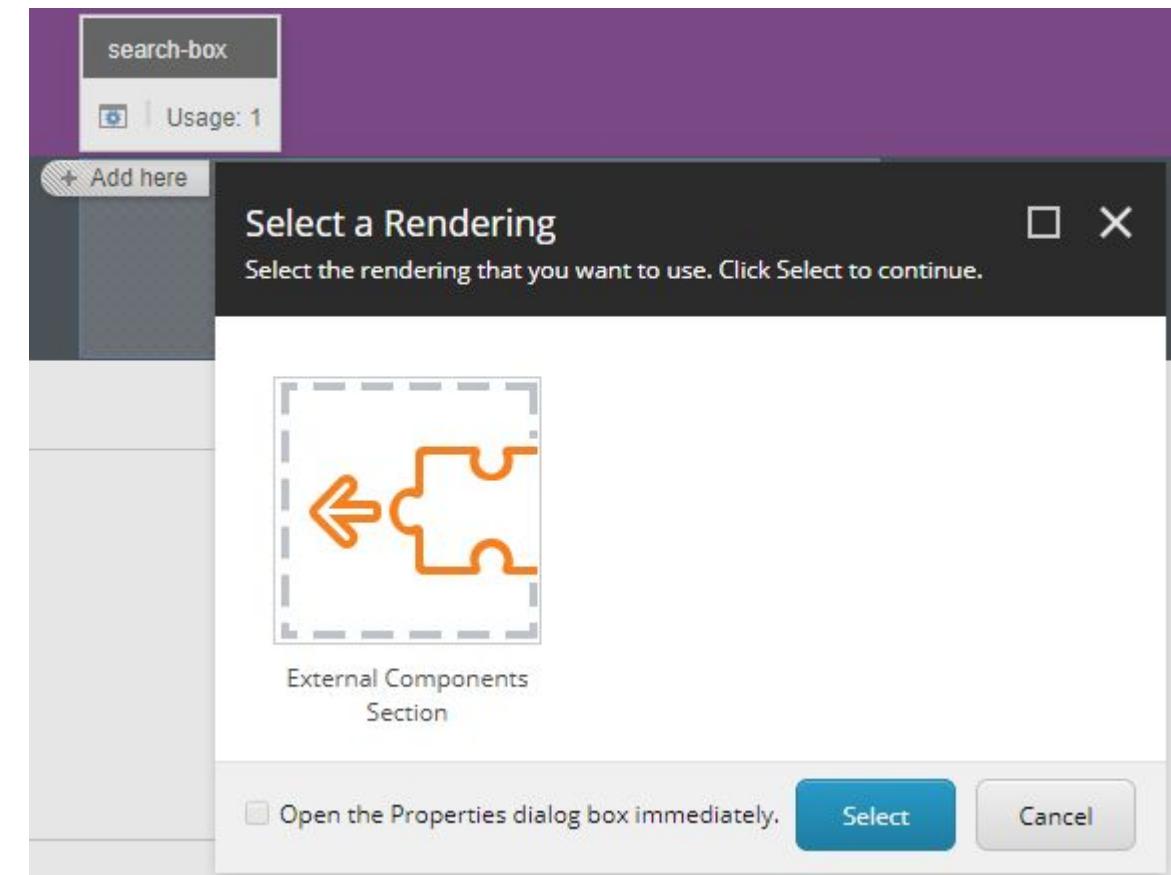
# Creating Placeholder Settings (3/3)

5. In the Select Items dialog
  - a. Select the External Components Section rendering
    - i. /Layout/Renderings/Coveo Hive/Sections/External Components Section
  - b. Choose OK
6. Save the item



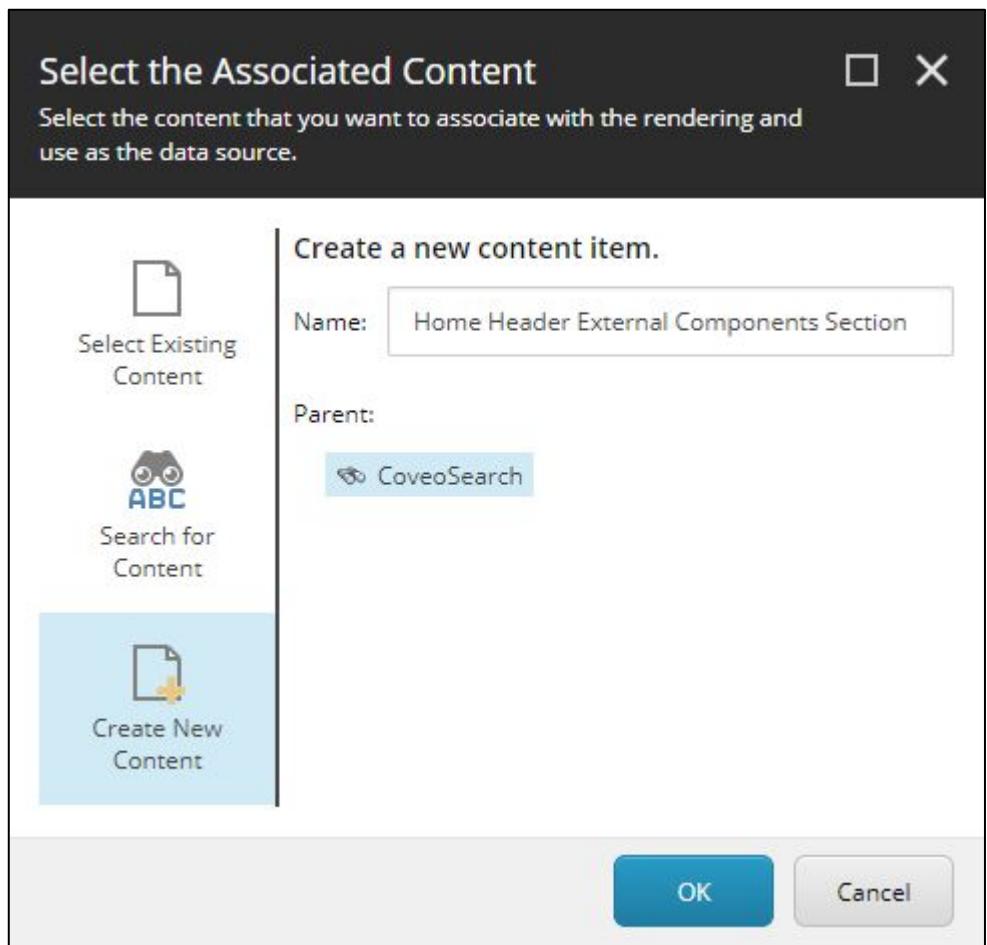
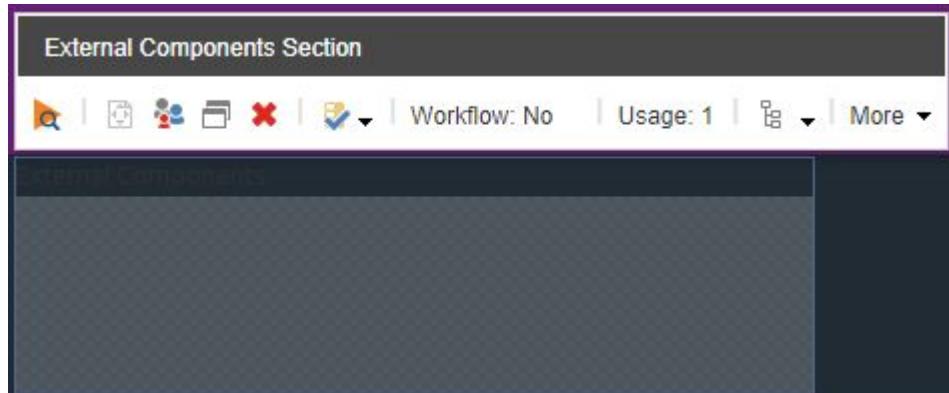
# Adding a Global Searchbox to a Search Page (1/5)

1. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
2. Access the Experience Editor (Publish > Experience Editor)
3. Select the search-box placeholder
4. Click the "Add here" button
5. In the Select a Rendering dialog
  - a. Select External Components Section
  - b. Choose Select



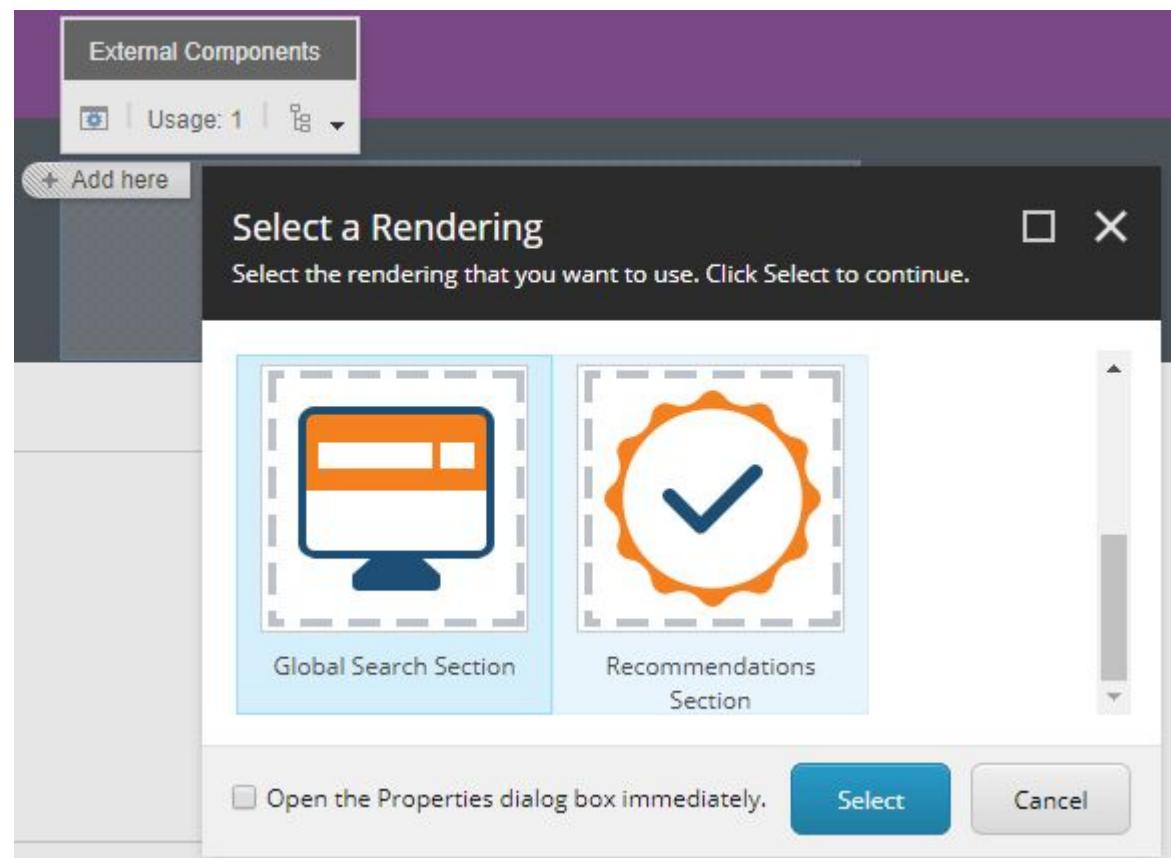
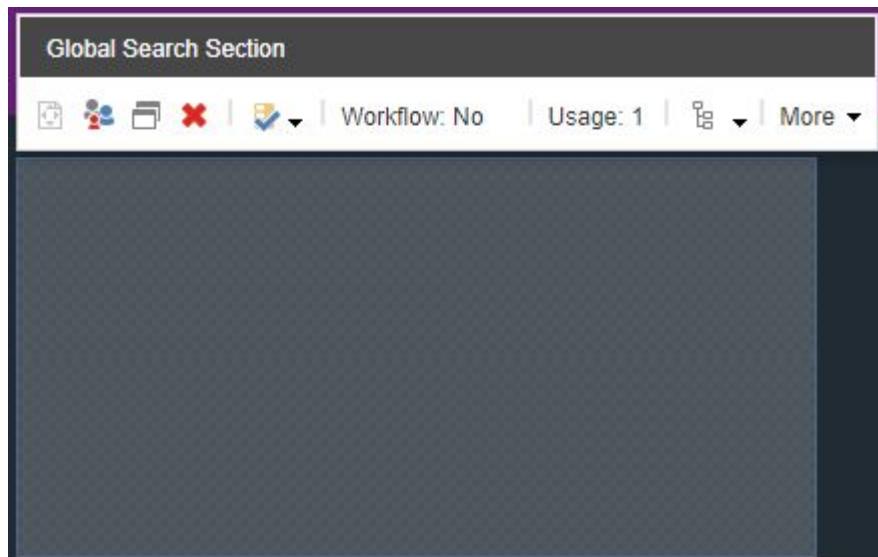
# Adding a Global Searchbox to a Search Page (2/5)

6. In the Select the Associated Content dialog
  - a. In the left column, select Create New Content
  - b. Name: Home Header External Components Section
  - c. Parent: CoveoSearch
  - d. Choose OK



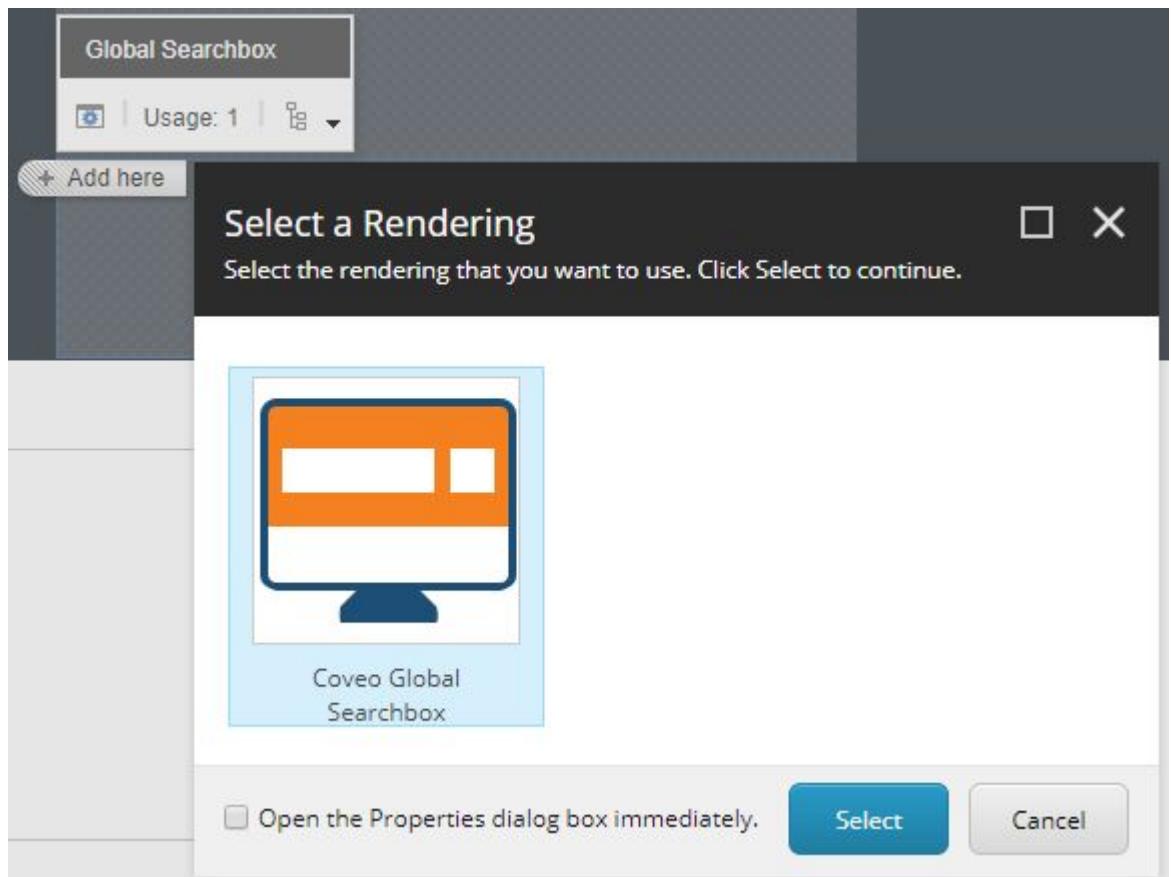
# Adding a Global Searchbox to a Search Page (3/5)

7. Select the External Components placeholder
8. Click the "Add here" button
9. In the Select a Rendering dialog
  - a. Select Global Search Section
  - b. Choose Select



# Adding a Global Searchbox to a Search Page (4/5)

10. Select the Global Searchbox placeholder
11. Click the "Add here" button
12. In the Select a Rendering dialog
  - a. Select Coveo Global Searchbox
  - b. Choose Select

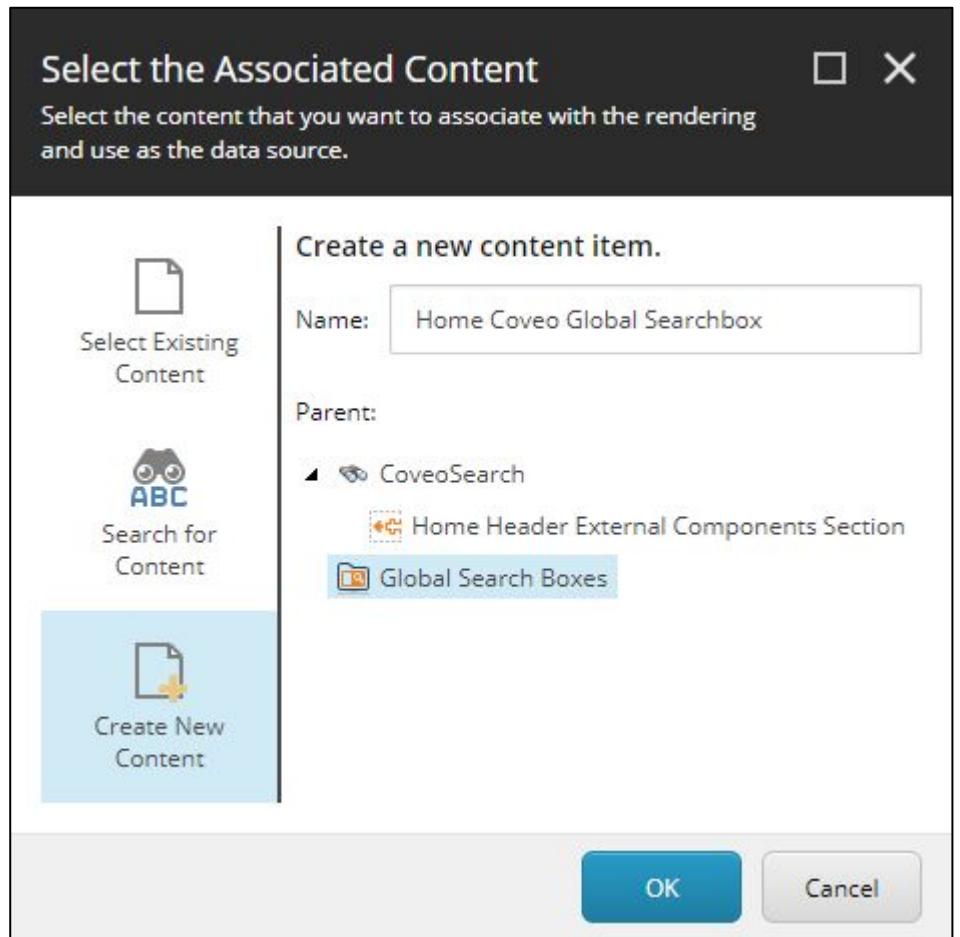


# Adding a Global Searchbox to a Search Page (5/5)

## 13. In the Select the Associated Content dialog

- a. In the left column, select Create New Content
- b. Name: Home Coveo Global Searchbox
- c. Parent: Global Search Boxes
- d. Choose OK

## 14. Save the item



# The Global Searchbox is Not Displayed

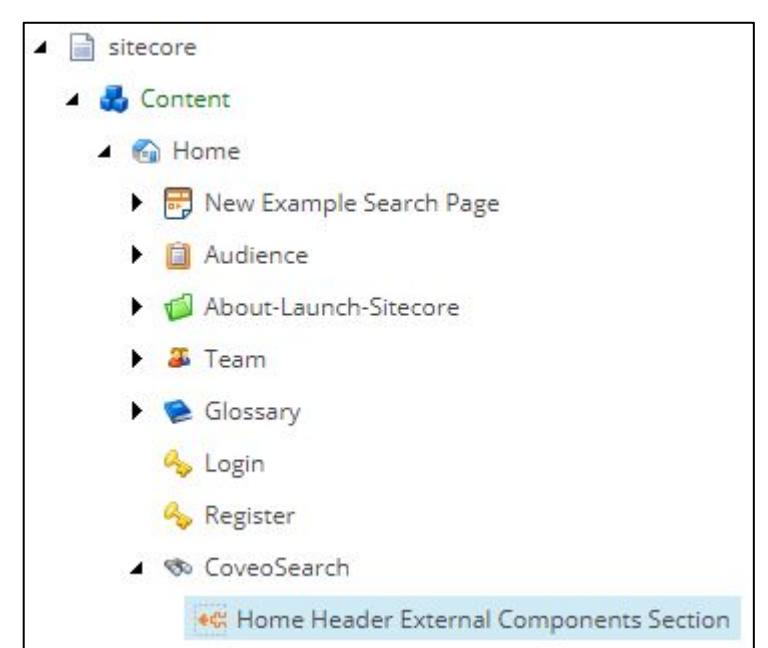
Why?



# Configuring the External Components Section (1/2)

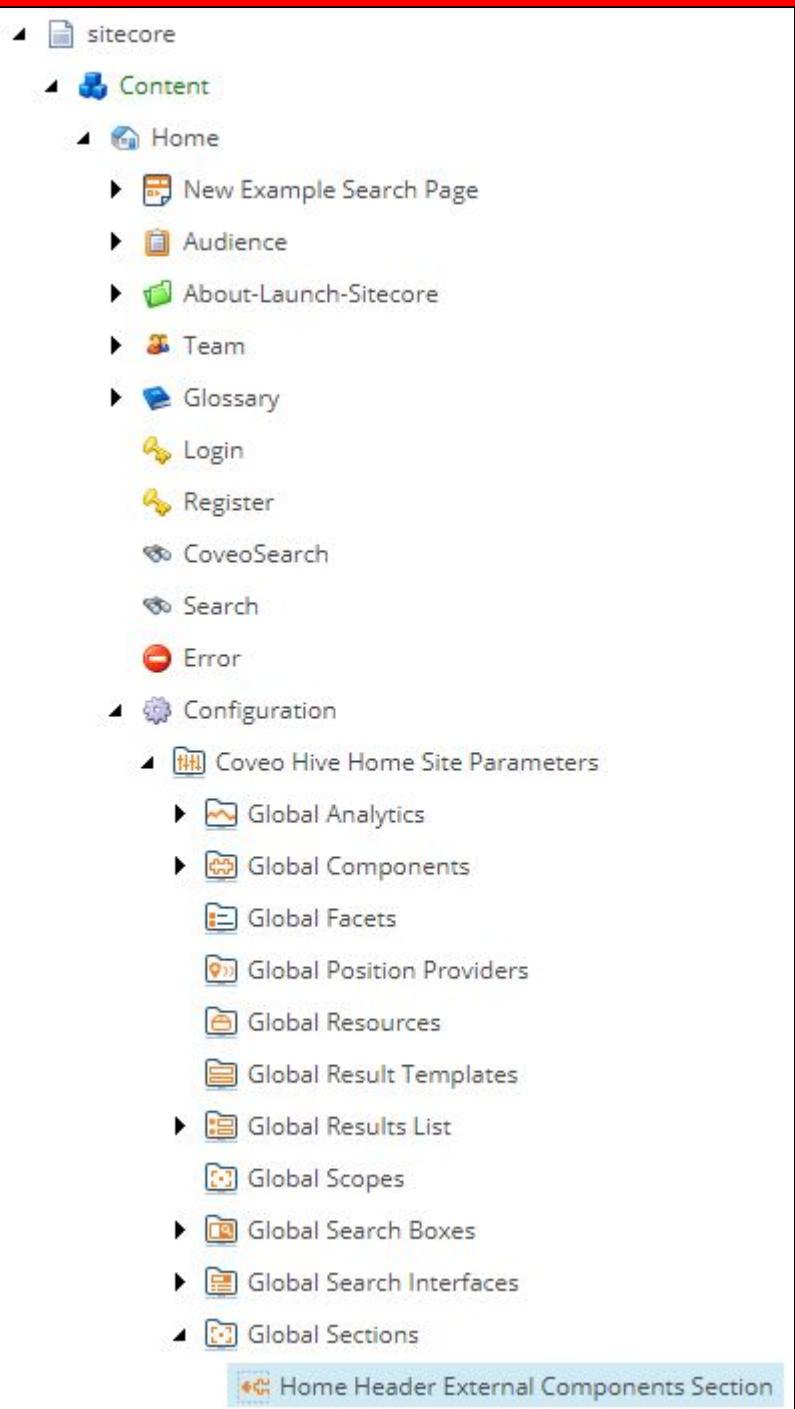
1. From the Sitecore Content Editor, select your Home Header External Components Section item

- a. /sitecore/Content/Home/CoveoSearch/ Home Header External Components Section



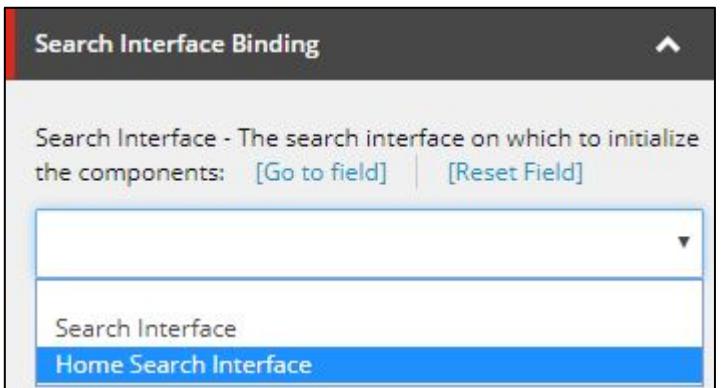
2. Move it to the Global Section folder with the other data source items

- a. /sitecore/Content/Home/Configuration/Coveo Hive Home Site Parameters/Global Sections/Home Header External Components Section



# Configuring the External Components Section (2/2)

3. In the Search Interface field (Search Interface Binding)
  - a. Select your Home Search Interface
4. Save the item
5. In the Experience Editor
  - a. Refresh the page

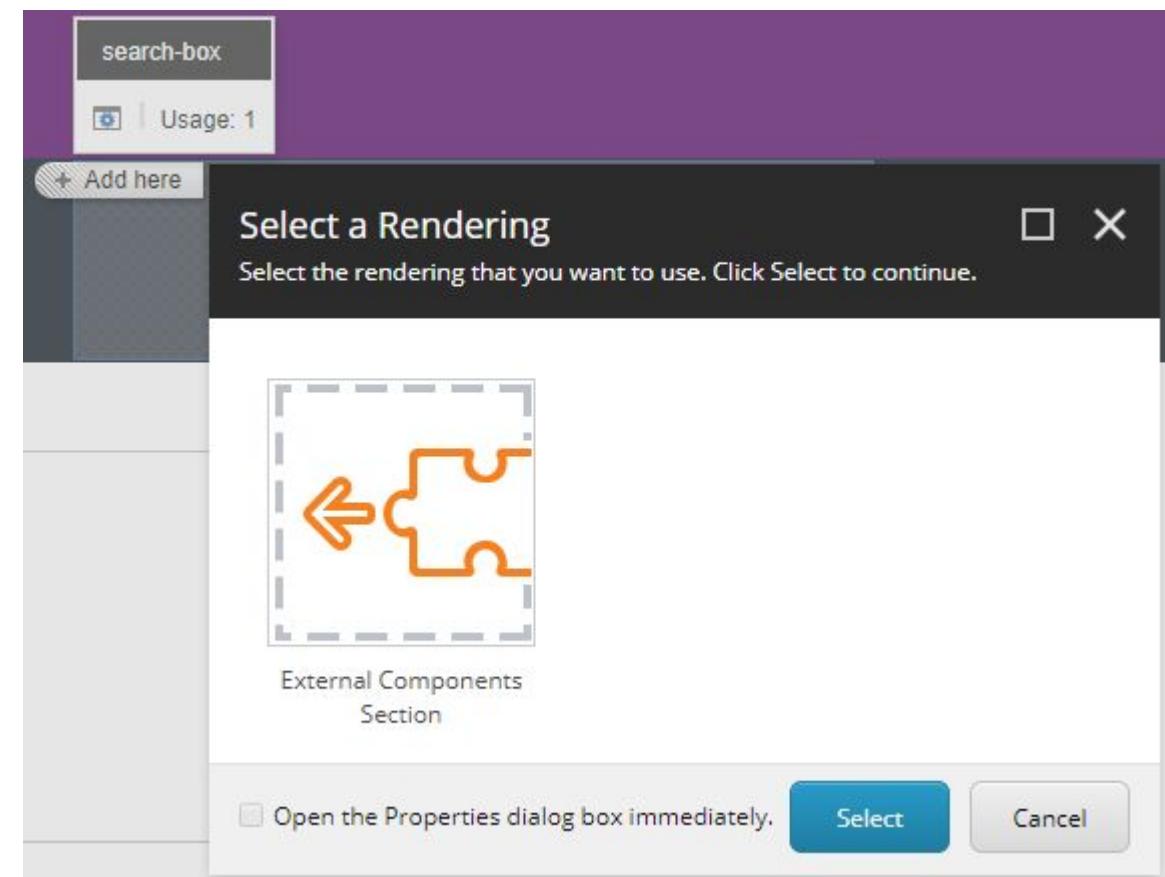


# Lab

1. Replace the Launch Sitecore header search box by a placeholder
2. Add a Placeholder Settings item for this new placeholder
3. Add the renderings to the placeholder:
  - a. External Components Section
    - i. Global Search Section
      1. Coveo Global Searchbox
4. Configure the External Components Section
5. Validate the search box is displayed and works

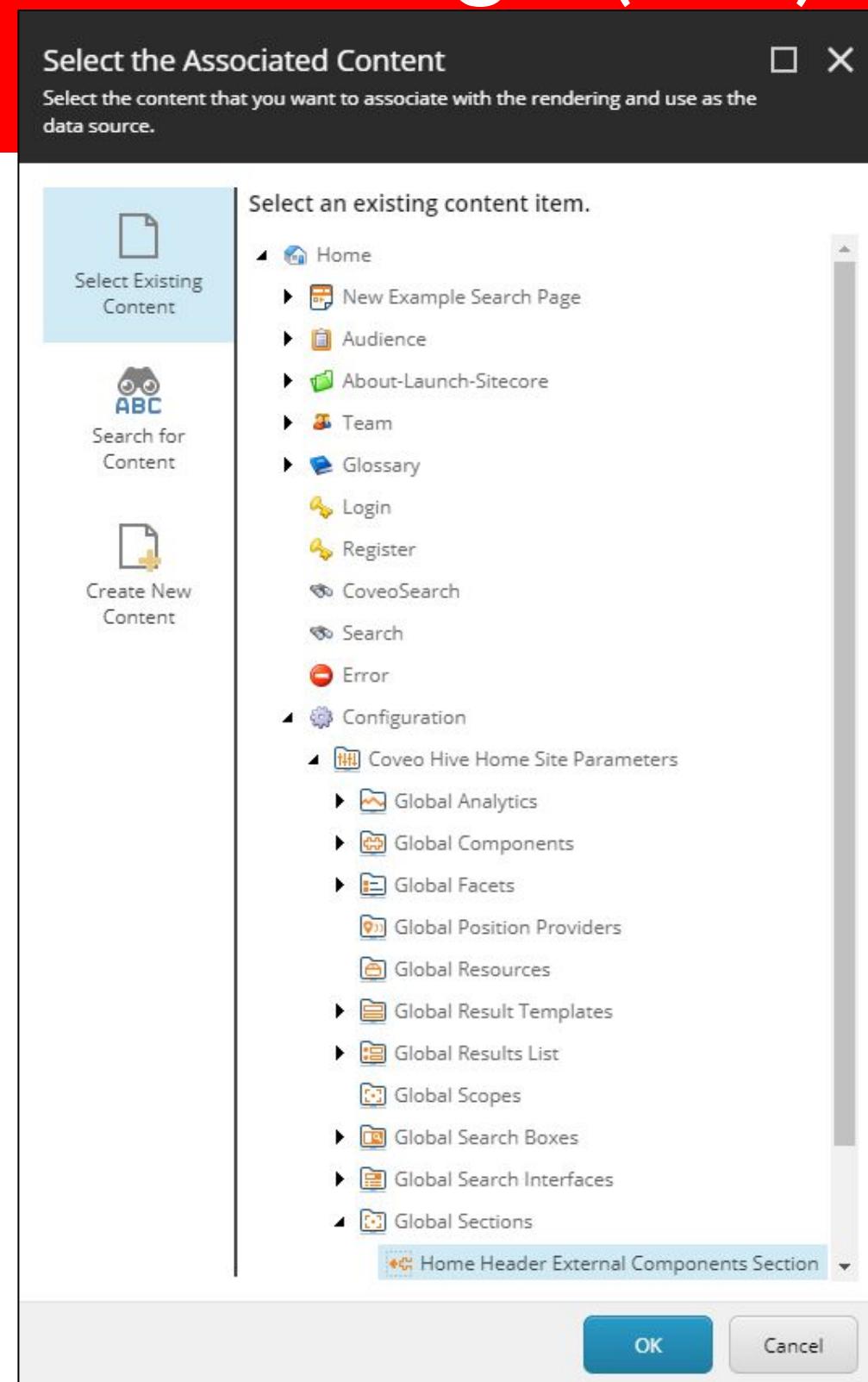
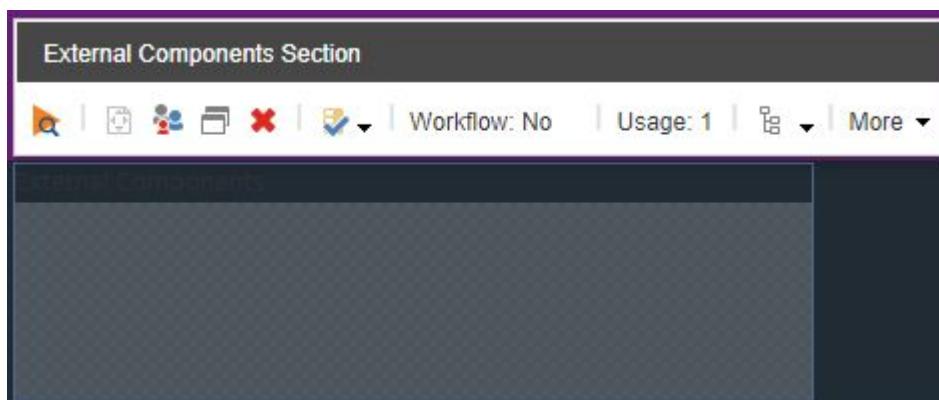
# Adding a Global Searchbox to the Home Page (1/5)

1. From the Sitecore Content Editor, select the Home content item
  - a. /sitecore/Content/Home
2. Access the Experience Editor (Publish > Experience Editor)
3. Select the search-box placeholder
4. Click the "Add here" button
5. In the Select a Rendering dialog
  - a. Select External Components Section
  - b. Choose Select



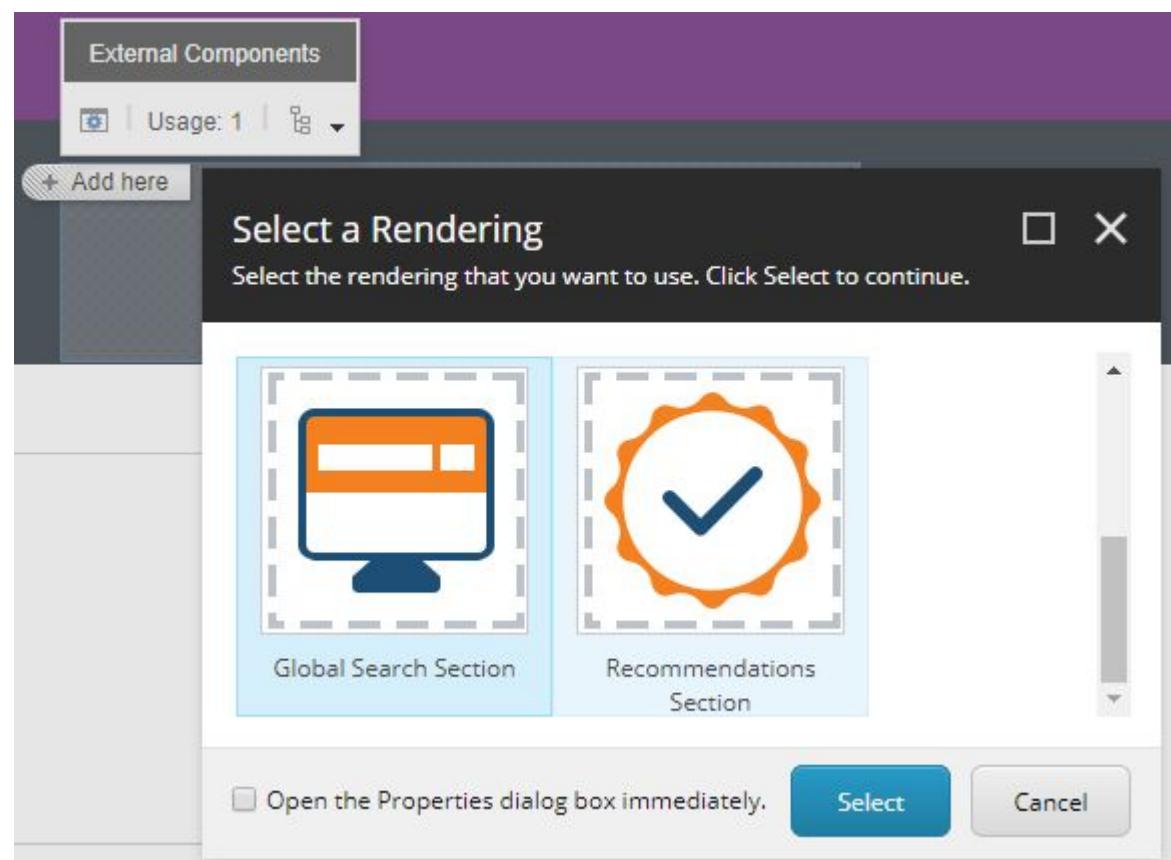
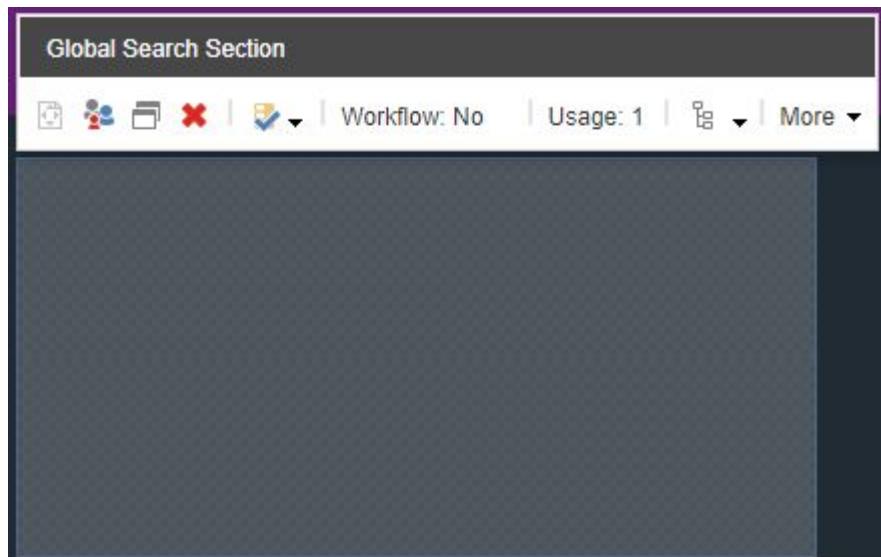
# Adding a Global Searchbox to the Home Page (2/5)

6. In the Select the Associated Content dialog
  - a. Select the existing Home Header External Components Section item
    - i. /Home/Configuration/Coveo Hive Home Site Parameters/Global Sections/Home Header External Components Section
  - b. Choose OK



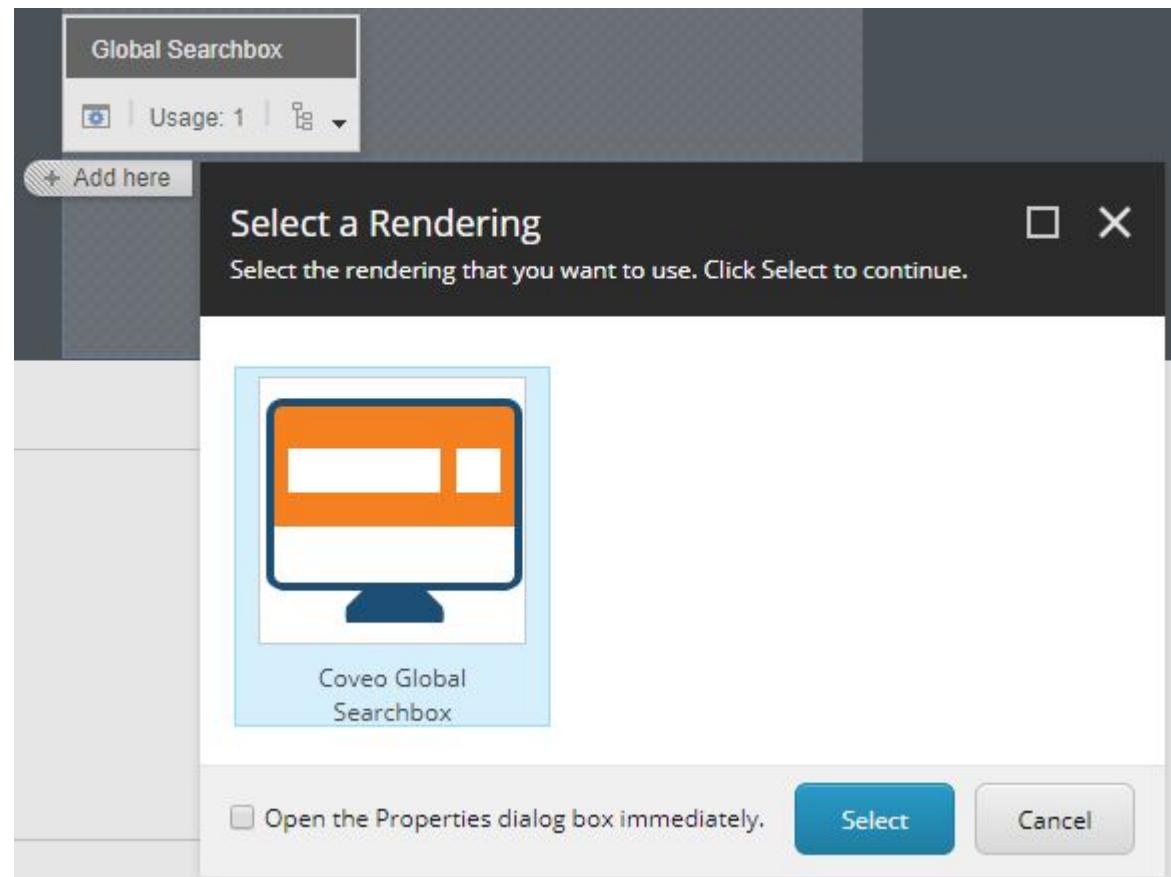
# Adding a Global Searchbox to the Home Page (3/5)

7. Select the External Components placeholder
8. Click the "Add here" button
9. In the Select a Rendering dialog
  - a. Select Global Search Section
  - b. Choose Select



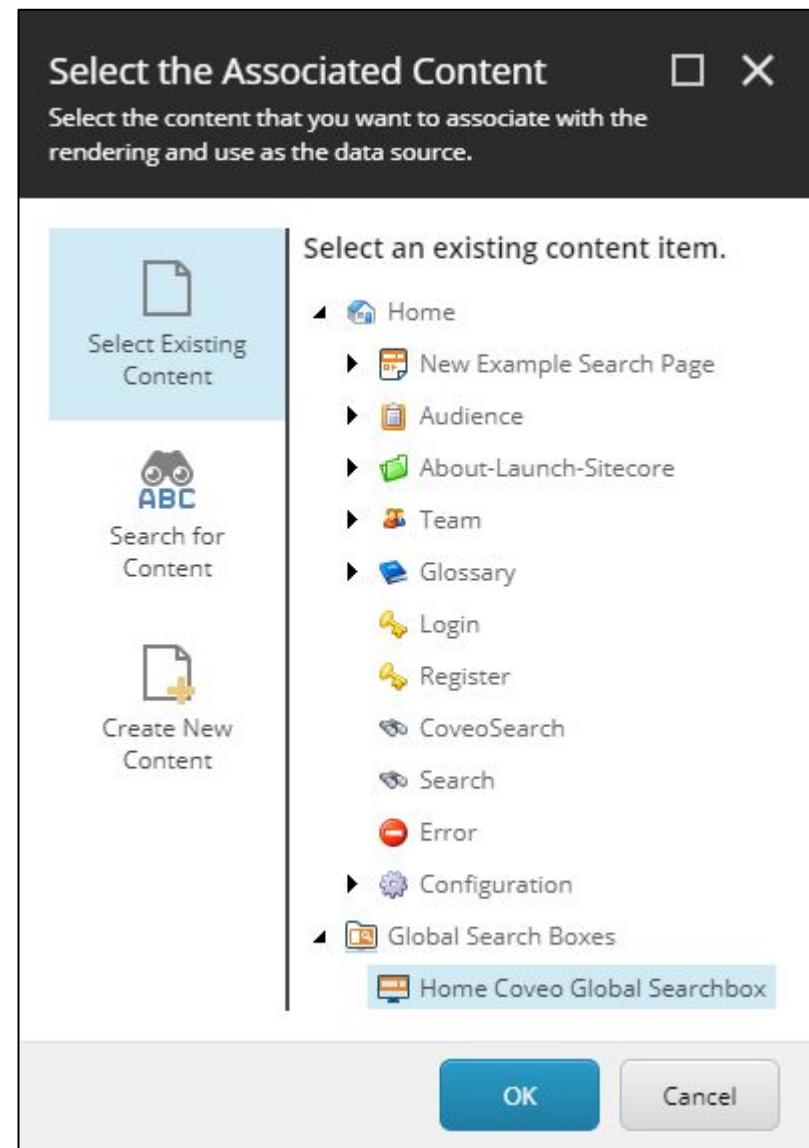
# Adding a Global Searchbox to the Home Page (4/5)

10. Select the Global Searchbox placeholder
11. Click the "Add here" button
12. In the Select a Rendering dialog
  - a. Select Coveo Global Searchbox
  - b. Choose Select



# Adding a Global Searchbox to the Home Page (5/5)

- 13.** In the Select the Associated Content dialog
  - a. Select the existing Home Coveo Global Searchbox item
    - i. /Global Search Boxes/Home Coveo Global Searchbox
  - b. Choose OK
- 14.** Save the item



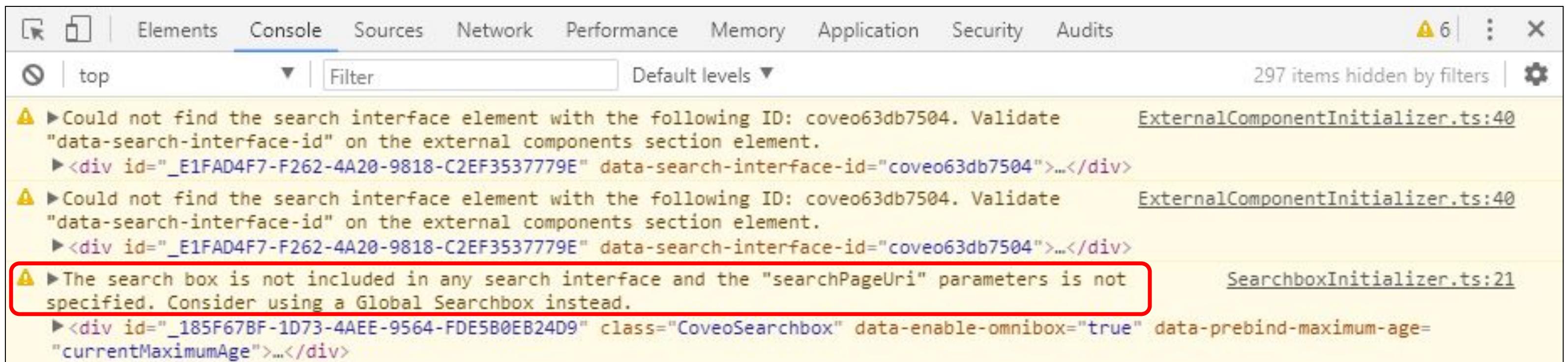
# The Global Searchbox is Not Displayed

Why?



# The Global Searchbox is Not Displayed

It is not initialized by the Coveo Search Interface anymore.  
We need to configure the search page to redirect to when using it.

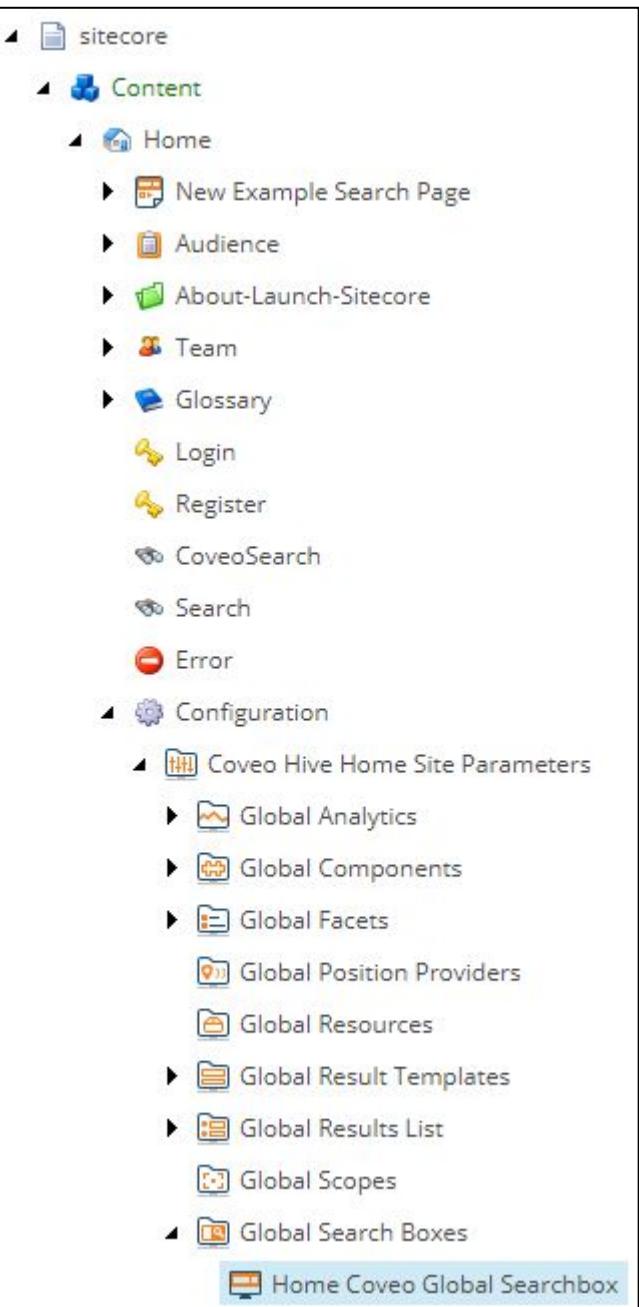


The screenshot shows the Chrome DevTools console tab with the following log entries:

- ▶ Could not find the search interface element with the following ID: coveo63db7504. Validate "data-search-interface-id" on the external components section element.  
▶ <div id="\_E1FAD4F7-F262-4A20-9818-C2EF3537779E" data-search-interface-id="coveo63db7504">...</div>
- ▶ Could not find the search interface element with the following ID: coveo63db7504. Validate "data-search-interface-id" on the external components section element.  
▶ <div id="\_E1FAD4F7-F262-4A20-9818-C2EF3537779E" data-search-interface-id="coveo63db7504">...</div>
- ▶ The search box is not included in any search interface and the "searchPageUri" parameters is not specified. Consider using a Global Searchbox instead.  
▶ <div id="\_185F67BF-1D73-4AEE-9564-FDE5B0EB24D9" class="CoveoSearchbox" data-enable-omnibox="true" data-prebind-maximum-age="currentMaximumAge">...</div>

# Configuring the Global Searchbox (1/3)

1. From the Sitecore Content Editor, select your Home Coveo Global Searchbox item
  - a. /sitecore/Content/Home/Configuration/Coveo Hive Home Site Parameters/Global Search Boxes/Home Coveo Global Searchbox

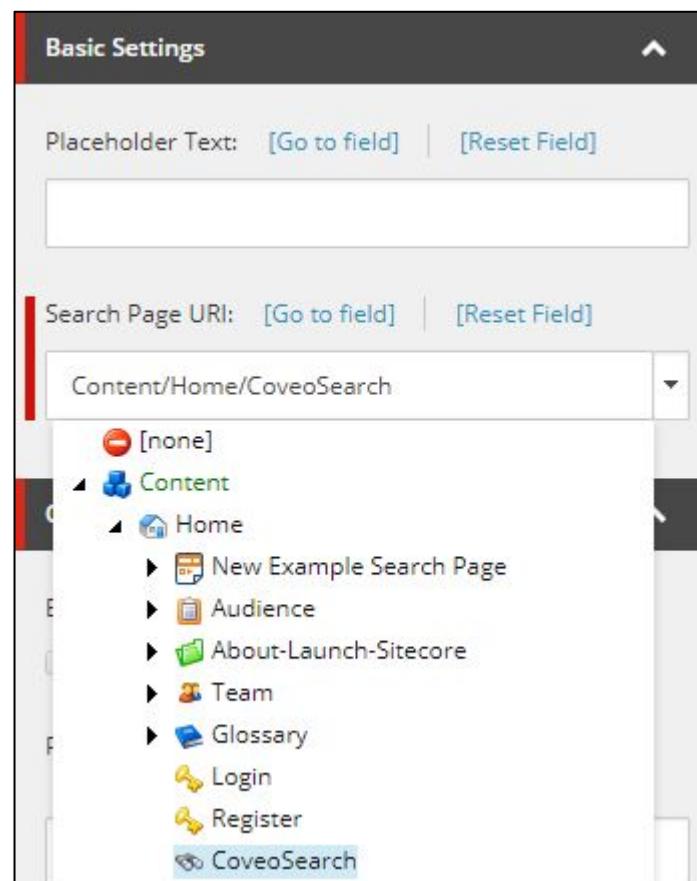


# Configuring the Global Searchbox (2/3)

## 2. In the Search Page URI field (Basic Settings)

- a. Select your CoveoSearch item
  - i. /Content/Home/CoveoSearch

## 3. Save the item



# Configuring the Global Searchbox (3/3)

4. In the Experience Editor
  - a. Refresh the page

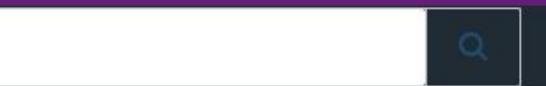


# Lab

1. Add a Coveo Global Searchbox to the Home item
2. Validate the searchbox is displayed and works in preview mode

# Publishing the Site

1. From the Sitecore Content Editor, smart publish the site  
(Publish tab > Publish dropdown button > Publish Site option > Smart publish radiobutton > Publish button)
2. In another browser or an incognito window, open your published site
  - a. <http://YourInstanceName>
3. Validate that the home page is displaying the Coveo Searchbox and redirecting the user to the result page when a search is made
4. Validate that the result page is displaying the Coveo Searchbox and executing the query

[Home](#) > Search Results

Results 1-10 of 101

[Search Results | Launch Sitecore](#)

About Launch Sitecore ... Team ... Glossary ... Login ... Copyright © 1999-2015 Sitecore ... Visit Details 10 ... Pattern ... No Match ... Unknown ... Visit is not associated with a campaign.

[Home | Launch Sitecore](#)

Reliability is key. ... Learn more » ... Getting Started with Launch Sitecore ... Get Started ... Home ... Audience Landing Pages ... About Launch Sitecore ... Team ... Glossary ... Login

[Content](#)[Basic Presentation Explanation - About Launch Sitecore | Launch Sitecore](#)

<sc:Placeholder ID="phHeader" Key="header" runat="server" /> ... This tag identifies the Header Placeholder region in the wireframe above. ... Home ... Audience Landing Pages ... Pattern

[Create a Goal - About Launch Sitecore | Launch Sitecore](#)

1) ... Clicking on the Goals button will bring up a checklist that shows all currently ... A screenshot of one of the Wizard steps from WFFM is below: ... Team ... Pattern ... Detailed

[The Sitecore UIs - About Launch Sitecore | Launch Sitecore](#)

Adding page Components ... Logging off ... Language: View the page content in another language/locale ... Of course, all these views are independent, so you can choose to view your page in a ...

[Create a custom validator - About Launch Sitecore | Launch Sitecore](#)

Field Type Level. ... The screenshot below shows the Required validator and the field that calls out the ... Create an item to represent the validator (as the Required validator does) ... Team

[Visit Details](#)[Visit Details](#)[Documentation](#)

# Lab

1. Smart publish your site
2. Test the home and search pages in published mode



## Indexing Scope

# Filtering

Filtering can be done at 2 moments:

- Indexing time
  - Called indexing scope
  - Restrict the number of indexed Sitecore items to avoid indexing items you know you don't want the users to search for
  - Examples: Index crawler root, Inbound filter, Excluded template...
  - Done by developers
- Query time
  - Restrict the search results returned for a search query
  - Done by developers
  - Can also be done by content authors, administrators and marketers with the Enterprise editions of Coveo for Sitecore

# Indexing Scope

It is recommended to reduce the number of indexed items over filtering the search interfaces results to improve indexing time and query performance.

You can do this with any combination of:

1. Using multiple crawlers and setting their root item
2. Using Coveo inbound filter pipeline processors
3. Including/Excluding templates

# Inbound Filters - Default Sitecore Pipeline

Sitecore comes with an inbound filter pipeline:

<indexing.filterIndex.inbound>

This pipeline is run for all the Sitecore indexes

Its usage is not recommended as it affects search inside Sitecore itself

# Inbound Filters - Coveo Specific Pipeline

It is recommended to use the Coveo inbound filter pipeline instead:

```
<coveoInboundFilterPipeline>
```

This pipeline is run only for Coveo indexes and allow you to choose for which indexes to run each processor.

```
<coveoInboundFilterPipeline>
  <processor type="Coveo.SearchProvider.InboundFilters.ApplySitecoreInboundFilterProcessor, Coveo.SearchProviderBase" />

  <!-- This processor ensures that only the latest version of a Sitecore item is indexed. -->
  <processor type="Coveo.SearchProvider.CoveoInboundFilters.IndexLatestVersionInboundFilter, Coveo.SearchProviderBase" />

  <!-- This processor executes the processors located in the coveoSitecoreInboundFilterPipeline pipeline below. -->
  <processor type="Coveo.SearchProvider.InboundFilters.InvokeSitecoreInboundFilterPipeline, Coveo.SearchProviderBase" />
</coveoInboundFilterPipeline>

<!-- Pipeline used to execute Sitecore inbound filter processors. This pipeline is run by the InvokeSitecoreInboundFilterPipeline processor in the
  coveoInboundFilterPipeline pipeline. -->
<!-- Add your existing Sitecore inbound filter processors to be run only on Coveo indexes. -->
<coveoSitecoreInboundFilterPipeline>
</coveoSitecoreInboundFilterPipeline>
```

# Changing the Indexing Scope Using Coveo Inbound Filters

1. From the file system, browse to your Training configuration folder
  - a. <your sitecore instance folder>\Website\App\_Config\Include\Training
2. Create and open a "IndexingScope.config" file
3. Initialize the file with this configuration

```
<configuration>
  <sitecore>
    <pipelines>
      <coveoInboundFilterPipeline>
        <processor type="Coveo.SearchProvider.CoveoInboundFilters.HasLayoutInboundFilter, Coveo.SearchProviderBase">
          <ApplyToPaths>/sitecore/content</ApplyToPaths>
        </processor>
      </coveoInboundFilterPipeline>
    </pipelines>
  </sitecore>
</configuration>
```

4. Save the file
5. Keep this file opened in your editor

# Changing the Indexing Scope by Excluding Templates

1. In your "IndexingScope.config" file, look for the "<pipelines>" node
2. Insert these nodes after its closing tag

```
</pipelines>
<coveo>
  <defaultIndexConfiguration>
    <documentOptions>
      <exclude hint="list:AddExcludedTemplate">
        <ErrorPage>{455B0CD7-43EE-4553-A7AD-FFFC1DA2143E}</ErrorPage>
        <CoveoExampleSearchPage>{0555887D-AB12-46E6-8EA8-93D2E6ED3CB4}</CoveoExampleSearchPage>
      </exclude>
    </documentOptions>
  </defaultIndexConfiguration>
</coveo>
</sitecore>
```

3. Save the file

# Rebuilding the Search Index to Change the Indexing Scope

1. Rebuild only your Coveo\_master\_index from the Indexing Manager
2. Validate that documents were filtered in the Coveo Cloud Administration Console Content Browser section
  - a. You should have 64 documents in the master source

The screenshot shows the 'Sources' page in the Coveo Cloud Administration Console. The page has a header with 'Sources' and a help icon, and an 'ADD SOURCE' button. Below the header is a sub-header 'Manage your sources to make content searchable.' A 'Filter' input field with a funnel icon is located on the right. The main area is a table with columns: Type, Name, Status, and Content. There is one item listed:

Type	Name	Status	Content
Sitecore Secured	Coveo_master_index - WKS-000615-DevTraining41 s4kdpqck7kd4jv7c2r37ln2cpe-jflheureuxdevtraining41d9jewksj	Ready to receive items ✓ Last rebuild completed an hour ago	64 items 1.3 MB

Results 1-10 of 64

#### [Search Results | Launch Sitecore](#)

Login ... Copyright © 1999-2015 Sitecore ... Visit Details 1 0 ... Pattern ... No Match ... Unknown ... Campaign ... Visit is not associated with a campaign. ... Location ...  
City: Pending

#### [Basic Presentation Explanation - About Launch Sitecore | Launch Sitecore](#)

<sc:Placeholder ID="phHeader" Key="header" runat="server" /> ... This tag identifies the Header Placeholder region in the wireframe above. ... Home ... Audience  
Landing Pages ... Pattern

#### [Create a Goal - About Launch Sitecore | Launch Sitecore](#)

1) ... Clicking on the Goals button will bring up a checklist that shows all currently ... A screenshot of one of the Wizard steps from WFFM is below: ... Team ... Pattern  
... Detailed

#### [The Sitecore UIs - About Launch Sitecore | Launch Sitecore](#)

Adding page Components ... Logging off ... Language: View the page content in another language/locale ... Of course, all these views are independent, so you can  
choose to view your page in a ...

#### [Create a custom validator - About Launch Sitecore | Launch Sitecore](#)

Field Type Level. ... The screenshot below shows the Required validator and the field that calls out the ... Create an item to represent the validator (as the Required  
validator does) ... Team

#### [Home | Launch Sitecore](#)

Learn more » ... Network Architects ... Reliability is key. ... You need a stable easy to manage environment that can scale to meet your needs. ... Home ... Copyright ©  
1999-2015 Sitecore

#### [Login | Launch Sitecore](#)

About Launch Sitecore ... Team ... Glossary ... Login ... Copyright © 1999-2015 Sitecore ... Visit Details 1 0 ... Pattern ... No Match ... Unknown ... Visit is not associated  
with a campaign.

#### [Search Results | Launch Sitecore](#)

Login ... Copyright © 1999-2015 Sitecore ... Visit Details 1 0 ... Pattern ... No Match ... Unknown ... Campaign ... Visit is not associated with a campaign. ... Location ...  
City: Pending

#### [Our Strategy - About Launch Sitecore | Launch Sitecore](#)

Enable multiple languages ... The separation of authoring and delivery instances allow us to perform these ... Home ... Audience Landing Pages ... About Launch  
Sitecore ... Team ... Glossary

# Indexing Scope Gotchas

- 1.** When Coveo for Sitecore replaces the default Sitecore indexes, filtering the indexed items is not recommended
  - a. Because Sitecore needs all the items to be indexed in the sitecore\_master\_index, sitecore\_web\_index and sitecore\_core\_index to run
  - b. Filter search results at query time instead
- 2.** Using the indexing.filterIndex.inbound pipeline is not recommended
  - a. Because it filters items in all the indexes at the same time
  - b. Use the coveoInboundFilterPipeline pipeline instead

# Lab

1. Change the indexing scope using a Coveo inbound filter
2. Change the indexing scope using excluded templates
3. Rebuild the master index
4. Validate the number of indexed documents



# Coveo JavaScript Search Framework

# History - .Net Front-End

The screenshot shows a search results page from the Coveo platform. The URL in the address bar is <https://demo.coveo.com/#s=4fdf33e898d2f729083446a1>. The search query is "Review Project start PNR". The results are displayed in a grid format.

Result	Date
<b>RE: Review Project start PNR350120 Finansa</b> From: Liz Smith To: Josh Davis; Bruce Lapointe Hi Josh Davis We need to discuss it during our next meeting. Regards, Liz ... Miss Smith We have changed a couple of sections in the document. Do you agree... Quick View   Search In Conversation   Attachments   Details + B. Dobs -> L. Smith... (May 25) Liz, works for me. Bob ... They have changed... Quick View L. Smith -> B. Dobs (May 23) They have changed our SOW. What do you think?... Quick View [W] V33_Finansa SOW for review.docx (Bob Dobs) - Quick View [W] V32_Finansa SOW for review.docx (Liz Smith) - Quick View [W] V31_Finansa SOW draft v2.docx (Josh Davis) - Quick View	May 26
<b>RE: Review Project start PNR35064 Finansa</b> From: Liz Smith To: Liz D Cruz; Avi Green Hi Liz D Cruz We need to discuss it during our next meeting. Regards, Liz ... Miss Smith We have changed a couple of sections in the document. Do you agree... Quick View   Search In Conversation   Attachments   Details + B. Dobs -> L. Smith; A. Green (May 25) Liz, works for me. Bob ... They have changed... Quick View L. Smith -> B. Dobs (May 24) They have changed our SOW. What do you think?... Quick View [W] V17_Finansa SOW for review.docx (Bob Dobs) - Quick View [W] V16_Finansa SOW for review.docx (Liz Smith) - Quick View [W] V15_Finansa SOW draft v2.docx (Liz D Cruz) - Quick View	May 26
<b>RE: Review Project start PNR350148 Finansa</b> From: Liz Smith To: Sean Forbes; Pat Stumuller Hi Sean Forbes We need to discuss it during our next meeting. Regards, Liz ... Miss Smith We have changed a couple of sections in the document. Do you agree...	May 25

On the left side, there are three filter panels: "From", "To", and "Type". The "From" panel lists names like Liz Smith, Bob Dobs, Arthur Song, etc. The "To" panel lists names like Liz Smith, Bruce Lapointe, Avi Green, etc. The "Type" panel has a single entry: "Email (2,000)".

# Why A New UI?

- Needed to integrate in new ecosystems
  - Salesforce
  - Sitecore
  - Customer websites
- .Net Front-End wasn't a good technology fit

# Overview

- Client side implementation of a search UI
- Written in TypeScript
- Open source on [GitHub](#)
- Available on [npm](#)
- Well documented
  - [Human written documentation](#)
  - [Generated API documentation](#)
  - [Getting started tutorial](#)
- Versions
  - V2 - Currently developed, supports lazy loading, used in Coveo for Sitecore 4.1 (Hive)
  - V1 - In maintenance mode, not developed anymore, used in Coveo for Sitecore 4.0 & 4.1

# Overview

- Only JS, CSS, HTML, images, and fonts files
  - Source maps and typings files are included
  - Debug and minified versions of JavaScript files are provided
- Requires any web server to serve the files
- Agnostic of the backend
- No client/server state management
- Designed to be extended
  - Fully skinnable & customizable
- Responsive by design
  - Optimizations for mobile

# Where?

- Can be downloaded as a single ZIP file containing everything needed
- Included in Coveo products
  - Coveo for Sitecore
  - Coveo for Salesforce
  - Coveo for Dynamics
  - On-premises Coveo Search API



sitecore®

# Coveo Search API

- Server component bridging the JS UI and the index
- Available in cloud & on-premises
- Offers query pipelines

# Coveo Search API - Query and Results

```
access_token: 52d806a2-0f64-4390-a3f2-e0f41a4a73ec
q: @sysuri
aq: @objecttype=="Board"
firstResult: 0
numberOfResults: 10
excerptLength: 200
filterFieldRange: 5
groupBy: [{"field": "@licategoryshorttitle", "maximumNumberOfValues": 6, "sortCriteria": "occurrences", "injectionDepth": 1000}, {"field": "@liboardshorttitle", "maximumNumberOfValues": 6, "sortCriteria": "occurrences", "injectionDepth": 1000}, {"field": "@sysauthor", "maximumNumberOfValues": 6, "sortCriteria": "occurrences", "injectionDepth": 1000}, {"field": "@liprofilelocation", "maximumNumberOfValues": 6, "sortCriteria": "occurrences", "injectionDepth": 1000}, {"field": "@sysconcepts", "maximumNumberOfValues": 6, "sortCriteria": "occurrences", "injectionDepth": 1000}]
enableDidYouMean: true
sortCriteria: relevancy
```

The screenshot shows a JSON viewer interface with a tree view on the left and a detailed preview on the right. The tree view shows the following structure:

- JSON
- totalCount : 23346
- duration : 75
- indexDuration : 15
- queryCorrections
- groupByResults
- results
- 0

The detailed preview for the first result (index 0) includes the following fields:

- title : "3126 cranks but will not start"
- uri : "https://forums.cat.com/community:caterpillar/category:ConstructionTrades/board:CT\_MachineOperationMaintenance/topic:16295/message:16295"
- printableUri : "https://forums.cat.com/community:caterpillar/category:ConstructionTrades/board:CT\_MachineOperationMaintenance/topic:16295/message:16295"
- clickUri : "http://forums.cat.com/t5/Machine-Operation-Maintenance/3126-crank-but-will-not-start/m-p/16295#M700"
- uniqueId : "-1\$0\$872.19805\$https://forums.cat.com/community:caterpillar/category:constructiontrades/board:ct\_machineoperationmaintenance/topic:16295/message:16295"
- excerpt : "This is on a piece of military Equipment DV100 deuce. ... We have tried everything sensors,pump,ecm,fuel regulator,etc... ... When it first stalled it was like it wa
- firstSentences : null
- summary : null
- excerptHighlights
- firstSentencesHighlights
- printableUriHighlights
- summaryHighlights
- titleHighlights
- parentResult : null
- childResults
- raw
  - systitle : "3126 cranks but will not start"
  - systopparent : "212240686856649185"
  - licategoryshorttitle : "Construction Trades"
  - lcommunityid : "caterpillar"
  - sysurihash : "Y8oB4QGmqvmCXfeE"
  - licategorytitle : "Construction Trades"
  - limessagekudoscount : 0
  - sysuri : "https://forums.cat.com/community:caterpillar/category:ConstructionTrades/board:CT\_MachineOperationMaintenance/topic:16295/message:16295"
  - limessageurl : "http://forums.cat.com/t5/Machine-Operation-Maintenance/3126-crank-but-will-not-start/m-p/16295#M700"
  - sysfolders : "00001https://forums.cat.com/b25767c8;00002https://forums.cat.com/community:caterpillar/f0807851;00003https://forums.cat.com/community:caterpillar/c
  - sysauthorloginname : "FMS"
  - litopickudoscount : 0
  - litopicsubject : "3126 cranks but will not start"

# Coveo Search API - Authentication

- The Search API is in charge of authenticating users
- This is normally done using either:
  - An API key
  - A search token (method used by Coveo for Sitecore)
  - SAML
  - Claims
- On each request sent by the browser, the Search UI will add the configured token in the request's header:

The screenshot shows a network request with the following headers:

Request Headers	Value
Accept	*/*
Accept-Encoding	gzip, deflate, br
Accept-Language	en-US, en; q=0.8, fr; q=0.6, zh-CN; q=0.4, zh; q=0.2, nl; q=0.2, pt; q=0.2
Authorization	Bearer [REDACTED]

# API Key

- In Coveo for Sitecore, an API Key is automatically created in the Coveo Cloud organization when it is linked at installation time
- It is encrypted and stored in the `Coveo.CloudPlatformClient.Custom.config` file
- It is used to:
  - Remotely configure the Coveo Cloud organization
  - Request search tokens for individual visitors to the website
- You don't have to manually create any additional API Key

# Search Token

- Allows to perform authenticated and secure search
- Are JWT tokens ([JSON web token](#))
  - Unique for each user
  - Short lived (15 minutes)
  - Cryptographically signed so that they cannot be tampered on the client side
  - Act as credentials for the user
- Are available in the page served by the web server

```
1 <script>
2   fetch('https://my.server.com/gettoken').then(function(token) {
3     Coveo.configureCloudEndpointV2('MyCoveoOrganizationID', token);
4     Coveo.init(document.body);
5   });
6 </script>
7
8 <body>
9 <!-- Content of search page -->
10 </body>
```

# Components Architecture

- 65+ out of the box
- A component == An HTML element with a known CSS class
- None is mandatory
- Can be repeated multiple times in a page
- Loosely coupled
- CSS provides default look for each component, easy to modify
- Configuration
  - HTML attributes and/or
  - JavaScript initialization options
- Can be interacted with using JavaScript code

# Custom Components

- Written in JavaScript, TypeScript, or any language that compiles to JavaScript (e.g.: CoffeeScript)
- Out of the box components use the same general API as custom components
- Possible to submit new components to the [Github repository](#) via pull request
- Custom component starter project
  - <https://github.com/coveo/search-ui-seed>
- Documentation
  - <https://developers.coveo.com/display/public/JsSearchV1/Step+10+-+Advanced+Integration+with+Custom+TypeScript+Component+-+Implementation>

# Usage Analytics

- Logging user actions in Coveo Usage Analytics
  - Plays a central role in every successful Coveo implementation
  - Is a requirement for the Coveo Machine Learning module
- Logging is not done at the Coveo Search API level
- All components are aware of the Coveo Usage Analytics module
- They log all the user actions (searches, refinements, result clicks, etc)
  - A component is also logging page views
  - Custom actions can be logged

# Customization

- Each component exposes multiple options
- Each component exposes multiple methods that you can call
- The outgoing search query can be customized using JavaScript
- Result templates are fully customizable using a templating engine
  - Custom HTML templates for basic needs
  - UnderscoreJS for advanced needs
- Look and design can be customized
  - <https://help.salesforce.com>
  - <https://helpx.adobe.com>
  - <http://support.logitech.com>
  - <http://www.enbridge.com>
  - <https://search.coveo.com>

# Basics

```
1  <!-- Include the library scripts. -->
2  <script src="js/CoveoJsSearch.js"></script>
3  <script src="js/templates/templatesNew.js"></script>
4
5  <!-- Each DOM element with a class starting with "Coveo" (uppercase) will instantiate a component. -->
6  <body id="search" class='CoveoSearchInterface'>
7
8      <!-- Each DOM element with a class starting with "coveo-" (lowercase) is strictly for CSS/alignment purpose. -->
9      <div class='coveo-search-section'>
10
11          <!-- Any Coveo component can be removed (or added); none is actually required for the page to "load". -->
12          <div class="CoveoSearchbox"></div>
13      </div>
14  </body>
```

# Component Options and Initialization

```
1 <body>
2   <!-- The "data-" attributes of each component allow you to pass options to this specific component instance. -->
3   <div class="CoveoFacet" data-title="Author" data-field="@author" data-tab="All"></div>
4   <div class="CoveoFacet" data-title="Year" data-field="@year" data-tab="All"></div>
5   <script>
6     // The following line shows you how you could configure an endpoint against which to perform your search.
7     // Coveo.SearchEndpoint.configureCloudEndpointv2('MyCoveoCloudEndpointName', 'my-authentication-token');
8
9     // We provide a sample endpoint with public sources for demo purposes.
10    Coveo.SearchEndpoint.configureSampleEndpointV2();
11
12    // Initialize the framework by targeting the root in the interface.
13    // It does not have to be the document body.
14    Coveo.init(document.body);
15  </script>
16</body>
```

# State

- Holds data about the state of the search interface (query, facets, etc.)
- Serialized in the browser URL hash (history management)
- Extensible (custom attributes)
- Changes in the state are automatically reflected in the UI, and respectively, using events

# Result Templates

- Use UnderscoreJS template engine
  - Supported delimiters
    - <% %> (Preferred in Coveo for Sitecore Hive)
    - {{ }}
- JavaScript context (this) object is the JSON result plus helper functions
- Components can be used inside results

# Underscore JS

- Rendering

- HTML-escaped <%- %>

```
<%- "text" %>
<%- variableThatMightContainHtml %>
```

- Not HTML-escaped <%= %>

```
<%= variableContainingHtmlToInsertInDom %>
```

- Defining variables

```
<% var aVariable = "aValue"; %>
```

- Condition

```
<% if (condition) { %>
  <div>Content A</div>
<% } else { %>
  <span>Content B</span>
<% } %>
```

# Localization

- 23 language files located in the \js\cultures folder
  - en.js, fr.js...
- Accessed in JavaScript via the Coveo.1 object: Coveo.1("Forward")
- Easy to add/override strings in any language

```
String.toLocaleString({
  "en": {
    "Forward": "Faster",      // overrides an existing string
    "CustomStringId": "Foo"   // defines a new string
  }
});
```



# Query Syntax

# Query Syntax

- Free-text queries
- Boolean operators
- Parenthesis
- Numerical field queries
- Date field queries
- Field queries
- Advanced field operators
- Nested queries
- REST Search API
  - Query extensions
  - No syntax block

# Query Syntax - Free-Text

Returns all items

- Containing a single term
  - Term
- Containing multiple terms (space is an AND boolean operator)
  - Term1 Term2
  - (Term1 Term2)
- Containing an exact phrase or term sequence
  - "Term1 Term2"
- Containing an exact term, not other words sharing the same root
  - +Term

# Query Syntax - Free-Text Wildcard

Disabled by default

Not performant

Must include at least two leading characters before the \*

Returns all items

- Containing a term or any word starting with it and ending with any characters
  - Term\*
- Containing a word with a character replaced by any character
  - Te?m

# Query Syntax - Boolean Operators

Returns all items

- Containing multiple terms
  - Term1 AND Term2
- Containing either terms
  - Term1 OR Term2
- Not containing a term
  - NOT Term
  - -Term
- Containing two terms contiguous within no more than X terms
  - Term1 NEAR:5 Term2

# Query Syntax - Parenthesis

Returns all items

- Matching multiple expressions
  - (...) (...)
  - (...) AND (...)
- Matching either expression
  - (...) OR (...)
- Not containing an expression
  - NOT (...)

# Query Syntax - Field

Returns all items

- With any value for the field
  - @fieldname
- Without a value for the field
  - NOT @fieldname

# Query Syntax - Numerical Field

Returns all items

- With the field greater than a value
  - `@fieldname>5`
- With the field greater or equal than a value
  - `@fieldname>=5`
- With the field smaller than a value
  - `@fieldname<5`
- With the field smaller or equal than a value
  - `@fieldname<=5`

# Query Syntax - Numerical Field

Returns all items

- With the field between two values, inclusive
  - `@fieldname=5..10`
- With the field between one set of two values, inclusive
  - `@fieldname=(5..10,15..20)`

# Query Syntax - Date Field

Same operators as numeric fields

Returns all items

- Related to a specific date
  - @fieldname=2017/01/31
- Related to a specific date and time
  - @fieldname=2017/01/31@22:50:00
- Related to a dynamic date
  - @fieldname=now
  - @fieldname=today
  - @fieldname=yesterday

# Query Syntax - Date Field

Returns all items

- Related to a calculated dynamic date
  - `@fieldname=today-7d`
  - `@fieldname=today+7d`

Valid modifiers are:

- Seconds (s)
- Minutes (m)
- Hours (h)
- Days (d)
- Months (mo)
- Years (y)

# Query Syntax - Field

Returns all items

- With the field containing a term
  - `@fieldname=Term`
  - `fieldname:Term`
- With the field not containing a term
  - `@fieldname<>Term`
  - `NOT @fieldname=Term`

# Query Syntax - Field

Returns all items

- With the field containing multiple terms
  - **@fieldname=(Term1 Term2)**
    - @fieldname=Term1 AND @fieldname=Term2
  - **fieldname:(Term1 Term2)**
- With the field not containing multiple terms
  - **@fieldname<>(Term1 Term2)**
    - @fieldname<>Term1 AND @fieldname<>Term2
  - **NOT @fieldname=("Term1", "Term2")**
    - NOT (@fieldname=Term1 OR @fieldname=Term2)
  - **NOTfieldname:(Term1", "Term2")**

# Query Syntax - Field

Returns all items

- With the field containing either terms
  - **@fieldname=**("Term1", "Term2")
    - @fieldname=Term1 OR @fieldname=Term2
  - **fieldname:**("Term1", "Term2")
- With the field not containing at least one of many terms
  - **@fieldname<>**("Term1", "Term2")
    - @fieldname<>Term1 OR @fieldname<>Term2
  - **NOT @fieldname=**(Term1 Term2)
    - NOT (@fieldname=Term1 AND @fieldname=Term2)
  - **NOT fieldname:**(Term1 Term2)

# Query Syntax - Field

Returns all items

- With the field containing an exact phrase or term sequence
  - @fieldname="Term1 Term2"
  - **fieldname:"Term1 Term2"**
- With the field not containing an exact phrase or term sequence
  - @fieldname<>"Term1 Term2"
  - NOT @fieldname="Term1 Term2"
  - NOT **fieldname:"Term1 Term2"**

# Query Syntax - Field

Returns all items

- With the field value exactly matching a term
  - `@fieldname==Term`
- With the field value exactly matching an exact phrase or term sequence
  - `@fieldname=="Term1 Term2"`
- With the field value exactly different than a term
  - `NOT @fieldname==Term`
- With the field value exactly different than an exact phrase or term sequence
  - `NOT @fieldname=="Term1 Term2"`

# Query Syntax - Advanced Field Operators

Not performant

Field must be configured as facet

- Wildcard match
- Fuzzy match
- Phonetic match
- Regular expression match

# Query Syntax - Field Wildcard

Returns all items

- With the field containing a term or any word starting with it and ending with any characters
  - `@fieldname*=Term*`
- With the field containing a word with a character replaced by any character
  - `@fieldname*=Te?m`

# Query Syntax - Field Fuzzy, Phonetic, Regular Expression

Returns all items

- With the field containing a value approximately the same as the queried value by allowing a reasonable sample (approximately 20%) of the queried characters to be different, missing, or added
  - `@fieldname~="Dipartiment"`
- With the field containing a value that sound similar to the queried values
  - `@fieldname%="Georg"`
- With the field matching the regular expression
  - `@fieldname/=[A-Z0-9._%+-]+@[A-Z0-9.-]+\\.[A-Z]{2,4}"`

# Query Syntax - Nested

Returns all items

- Matching a second query on a subset of items obtained by a first query
  - OutExpression @Outfield=[[@Infield] InExpression]
  - OutExpression [[@Infield] InExpression]
  - OutExpression @Outfield=[[@Infield] InExpression @Outfield2=[[@Infield2] InExpression2]]
  - @filetype=artist [[@artistid] [[@albumid] @songtitle=(love)] @genre=rock]
    - Artists that have a rock album including at least one song that has the word love in its title.

# Query Syntax - Query Extensions

Handled by the REST Search API

- \$qf
  - Calculates a mathematical value from the result's data and injects it in a field
  - `$qf(function: 'dist(@latitude, @longitude, 46.8167, -71.2167)', fieldName: 'distance')`
- \$qre
  - Injects a query ranking expression (QRE) in the query
  - `$qre(expression: '@accountname=="MyCompany"', modifier: '10')`
- \$qrf
  - Injects a query ranking function (QRF) in the query. Calculates a mathematical value from the result's data and adds it to the item score.
  - `$qrf(expression: 'sqrt(@articleviewcount)', normalizeWeight: true)`

# Query Syntax - No Syntax Block

Handled by the REST Search API

Returns all items

- Containing the terms but special characters within the no syntax block (<@- and -@>) are removed
  - **Term1 <@- +Term2\*(Term3) -@> Term4**
    - Term1 Term2 Term3 Term4



## Results Filtering

# Filtering/Boosting Search Results

- **Filtering**
  - Restrict the possible search results at query time by adding query expressions to all the user queries.
  - Example: `@ShowInSearchResults == "1"`
  
- **Boosting**
  - Modify the order of the search results by adding or removing points to their ranking score. Points are added to a subset of the search results matching a query expression.
  - Example: `$qre(expression:'@TemplateName == "Product", modifier: '10')`

# Where to Define the Filtering/Boosting Rules

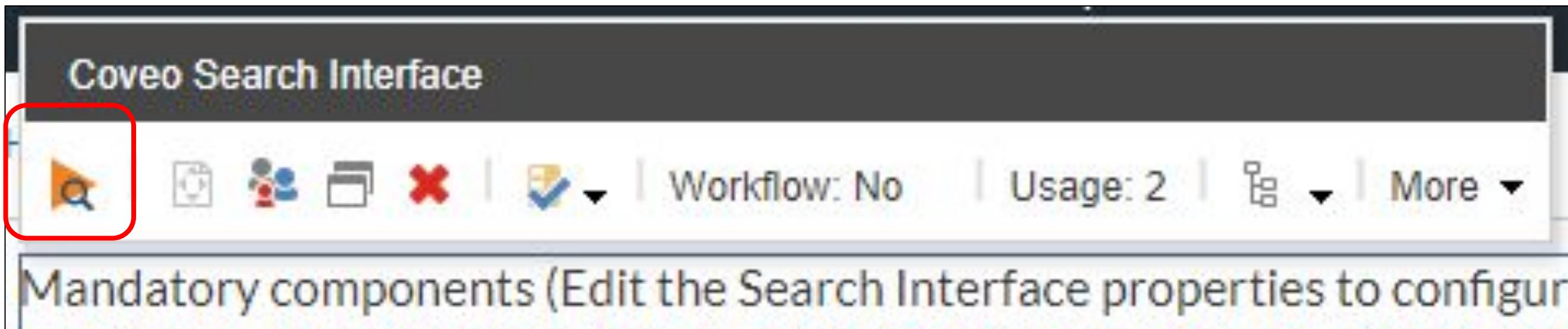
2 alternatives to define the rules:

- Using JavaScript code
  - Only option for the Free/Pro Editions
  - Also used for rules you don't want the content editors to edit
  - Requires a developer who understands the Coveo query syntax
- In the Coveo Search Interface or Coveo Tab component data source item
  - Only available in the Enterprise Edition
  - Uses the Sitecore rule editor control to easily configure rules
  - Understanding of the Coveo query syntax is not needed
  - Used for rules you want the content authors to control

# Filtering Search Results in the Properties Editor (1/4)

## Display Items With "Show In Search Results" Checked

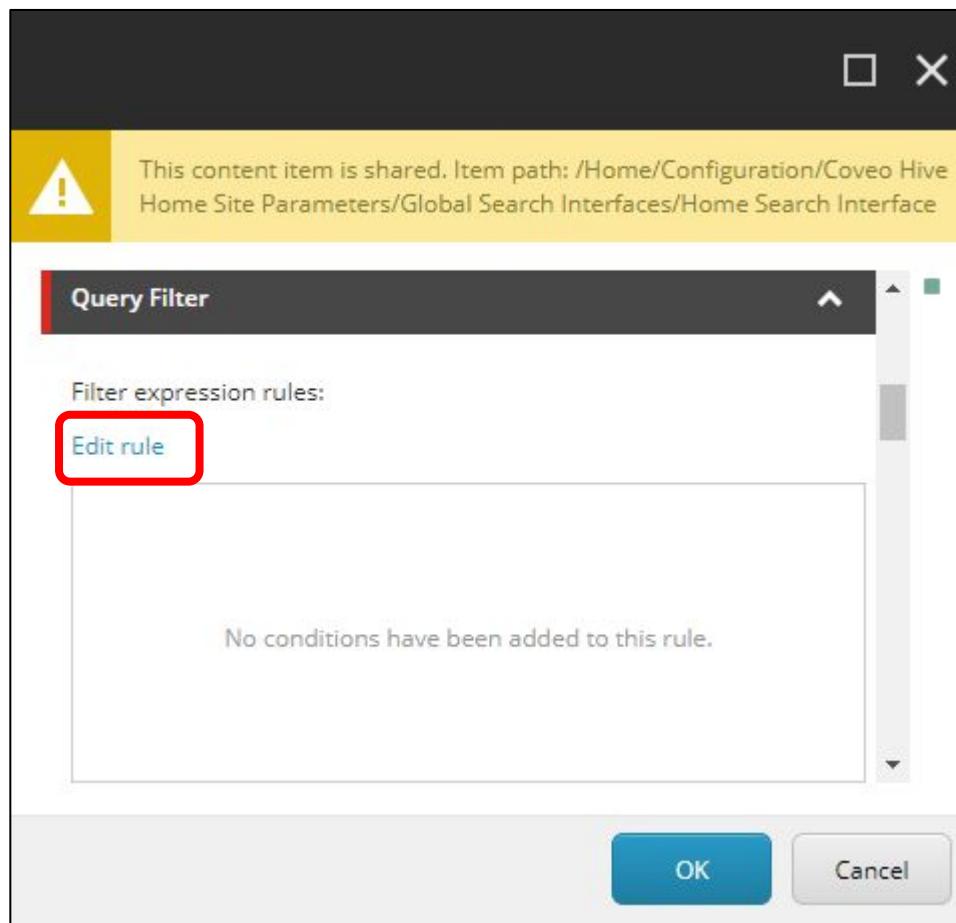
1. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
2. Access the Experience Editor (Publish > Experience Editor)
3. Select the Coveo Search Interface component
4. Click the Coveo experience button in the floating toolbar



# Filtering Search Results in the Properties Editor (2/4)

## Display Items With "Show In Search Results" Checked

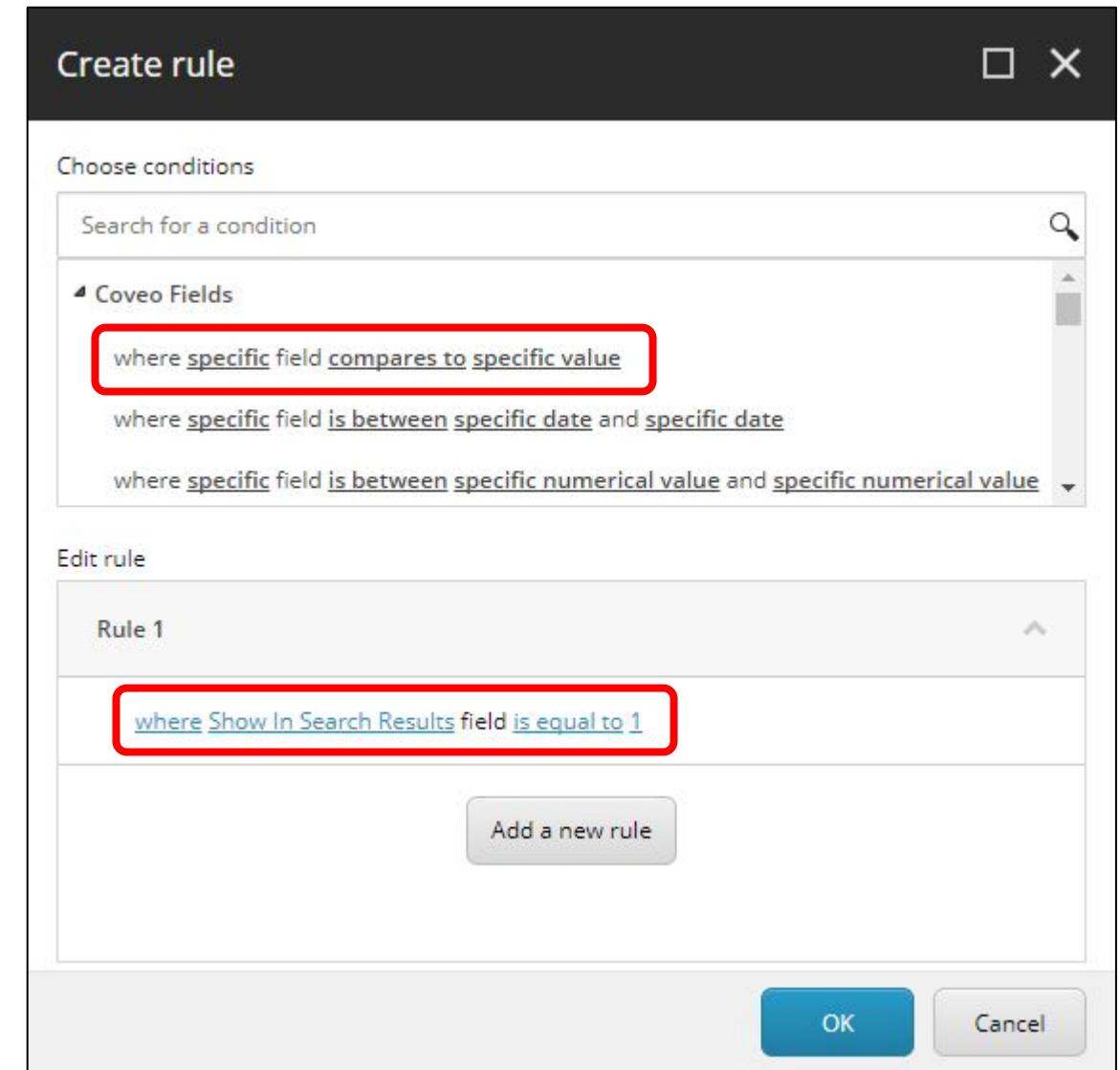
5. In the Control Properties dialog, edit the filter expression rules  
(Query Filter > Filter expression rules > Edit rule link)



# Filtering Search Results in the Properties Editor (3/4)

## Display Items With "Show In Search Results" Checked

6. In the Create rule dialog, click the "where specific field compares to specific value" condition under Coveo Fields
7. In the Edit rule bottom section
  - a. Click the links:
    - i. specific: Type "Show In Search Results"
    - ii. compares to: Select "is equal to"
    - iii. specific value: Type "1" (without the quotes)
  - b. Choose OK
8. In the Control Properties dialog, choose OK



# Filtering Search Results in the Properties Editor (4/4)

## Display Items With "Show In Search Results" Checked

9. Validate that your page is using your filtering rule
  - a. You should see 62 search results

Results 1-10 of 62

[Basic Presentation Explanation - About Launch Sitecore | Launch Sitecore](#)

<sc:Placeholder ID="phHeader" Key="header" runat="server" /> ... This tag identifies the Header Placeholder region in the wireframe above. ... Home ... Audience Landing Pages ... Pattern

[Create a Goal - About Launch Sitecore | Launch Sitecore](#)

1) ... Clicking on the Goals button will bring up a checklist that shows all currently ... A screenshot of one of the Wizard steps from WFFM is below: ... Team ... Pattern ... Detailed

[The Sitecore UIs - About Launch Sitecore | Launch Sitecore](#)

Adding page Components ... Logging off ... Language: View the page content in another language/locale ... Of course, all these views are independent, so you can choose to view your page in a ...

[Create a custom validator - About Launch Sitecore | Launch Sitecore](#)

Field Type Level. ... The screenshot below shows the Required validator and the field that calls out the ... Create an item to represent the validator (as the Required validator does) ... Team

[Home | Launch Sitecore](#)

Learn more » ... Network Architects ... Reliability is key. ... You need a stable easy to manage environment that can scale to meet your needs. ... Home ... Copyright © 1999-2015 Sitecore

[Login | Launch Sitecore](#)

About Launch Sitecore ... Team ... Glossary ... Login ... Copyright © 1999-2015 Sitecore ... Visit Details 1 0 ... Pattern ... No Match ... Unknown ... Visit is not associated with a campaign.

[Our Strategy - About Launch Sitecore | Launch Sitecore](#)

Enable multiple languages ... The separation of authoring and delivery instances allow us to perform these ... Home ... Audience Landing Pages ... About Launch Sitecore ... Team ... Glossary

[Inline Personalization - About Launch Sitecore | Launch Sitecore](#)

Click a rule to add it to the rule description. ... 6. Click ok and add more conditions if necessary. ... Home ... Audience Landing Pages ... About Launch Sitecore ... Team ... Glossary

[Optimizing the Experience - About Launch Sitecore | Launch Sitecore](#)

The values you assign to each page accumulate as a visitor navigates through the site. ... This allows you to build a profile for an individual site visitor, which can then be used ... Pattern

# Anatomy of a Coveo for Sitecore Query With Properties Editor Filtering Rules

Advanced Query:

```
(  
  (  
    Filtering Rules from the Properties Editor  
    Item Templates Exclusion from the Properties Editor Options  
  )  
)  
(  
  Automatic Source Filter  
)
```

Constant Query doesn't change:

Various Filters from the Properties Editor Options

# Lab

## 1. Filter search results using the properties editor

# Fields Translation - Indexing Time

By default, Sitecore field names are translated before being saved in the Coveo index to avoid field collision when:

- Multiple Sitecore instances are indexed in the same Coveo index
- A field type changes in Master and is not yet published in Web
- A field have different configuration in 2 Sitecore indexes

# Fields Translation - Indexing Time

Field names are prefixed by the letter "f" and suffixed by a 5 digits maximum number representing the Coveo source name hash.

All non alphanumeric characters are converted to their character code prefixed by the letter "z" and suffixed by the letter "x".

"z" and "x" characters are also converted to their character code:

- x: "z120x"
- z: "z122x"

Sitecore field: Call To Action Text

Coveo field: **@fcallz32xtoz32xactionz32xtez120xt94572**

# Fields Translation - `toCoveo(string fieldName)`

In the views, translation of field names is done by a JavaScript function

```
CoveoForSitecore.Context.fields.toCoveo(string fieldName)
```

# Fields Translation - Field Name Formats

- **toCoveo('Sitecore Field Name')**
  - Encodes special characters, adds "f" prefix, adds source hash suffix:  
**fsitecorez32xfieldz32xnameXXXXX**
- **toCoveo('Sitecore External Field Name')**
  - Only encodes special characters: **sitecorez32xez120xternalz32xfieldz32xname**
- **toCoveo('NonSitecoreFieldName')**
  - Non Sitecore fields are already in the Coveo format: **NonSitecoreFieldName**

In all cases, the field name can be prefixed by the "@" character:

- **toCoveo('@SitecoreFieldName')**
  - "@" character will be kept as the field name prefix: **@fsitecorefieldnameXXXXX**

# Never Modify Coveo Files and Items

To avoid your changes to be overwritten during an upgrade of Coveo for Sitecore, we recommend to never modify the files and items installed by the package. If you really need a customization, you should instead always duplicate a file and the associated items before attempting a modification.

This applies to:

- All items
  - Layouts, renderings, templates...
- Most of the files
  - Views, JavaScript, CSS, images...
  - Coveo.\*.config (Except Coveo.\*.Custom.config files)

# Filtering Search Results in the JS UI (1/9)

## Display Current Site Items Only

1. From the Sitecore Content Editor, select the Renderings folder

a. /sitecore/Layout/Renderings

2. Insert a new Rendering Folder

(right-click > Insert > Rendering Folder)

a. Name: Training

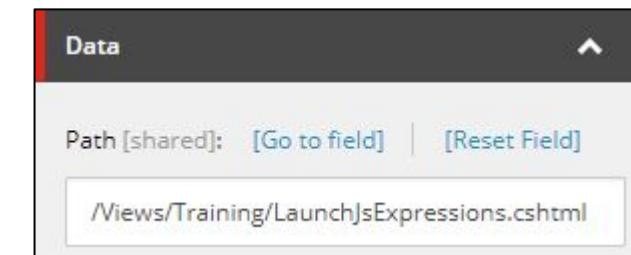
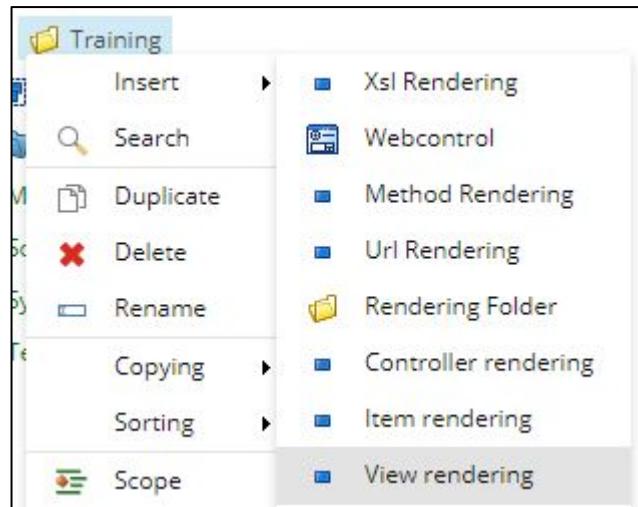
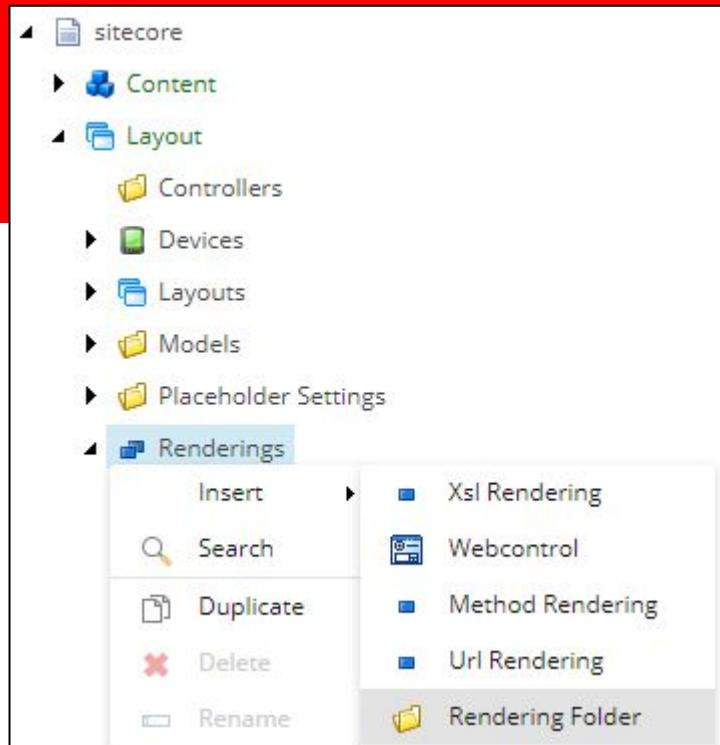
3. In the new folder, insert a new view rendering

(right-click > Insert > View rendering)

a. Name: Launch JavaScript Expressions

4. Change the value of the "Path" field to:

a. /Views/Training/LaunchJsExpressions.cshtml



# Filtering Search Results in the JS UI (2/9)

## Display Current Site Items Only

5. Save the item
6. From the file system, browse to the Views folder
  - a. <your sitecore instance folder>\Website\Views
7. Create and open a "Training" folder
8. Create and open a "LaunchJsExpressions.cshtml" file

# Filtering Search Results in the JS UI (3/9)

## Display Current Site Items Only

### 9. Initialize the file with this code

```
@using Coveo.UI.Components  
@using Coveo.UI.Components.Extensions  
  
@Html.Partial(Partials.EDIT_TITLE, Html.Coveo().GetViewName())  
  
<div id="currentSiteFilterExpression" class="CoveoForSitecoreFilterExpression"></div>  
  
<script>  
    document.addEventListener('DOMContentLoaded', function() {  
        var expressionBuilder = new Coveo.ExpressionBuilder();  
        var fieldName = CoveoForSitecore.Context.fields.toCoveo('@@_path');  
        var siteRootItemShortId = '@LaunchSitecore.Configuration.SiteConfiguration.GetHomeItem().ID.ToShortID()';  
        expressionBuilder.addFieldExpression(fieldName, '==', [siteRootItemShortId]);  
        document.getElementById('currentSiteFilterExpression').dataset.scAdvancedFilter = expressionBuilder.build();  
    });  
</script>
```

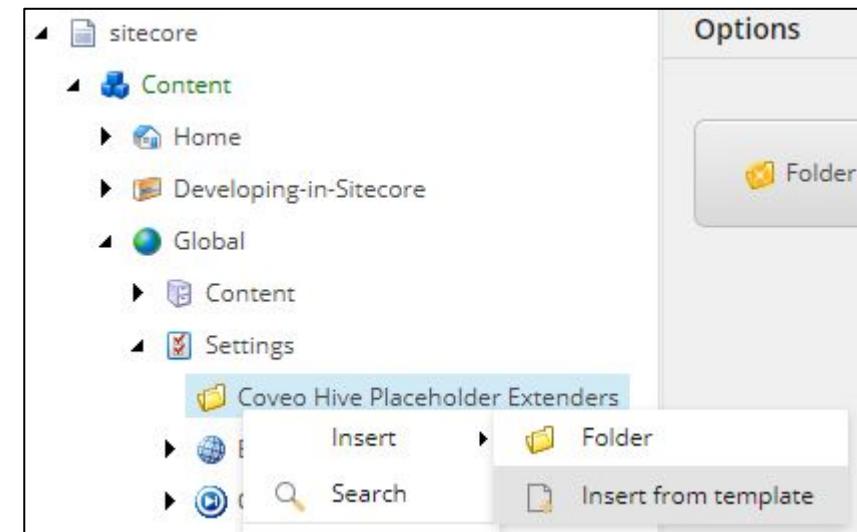
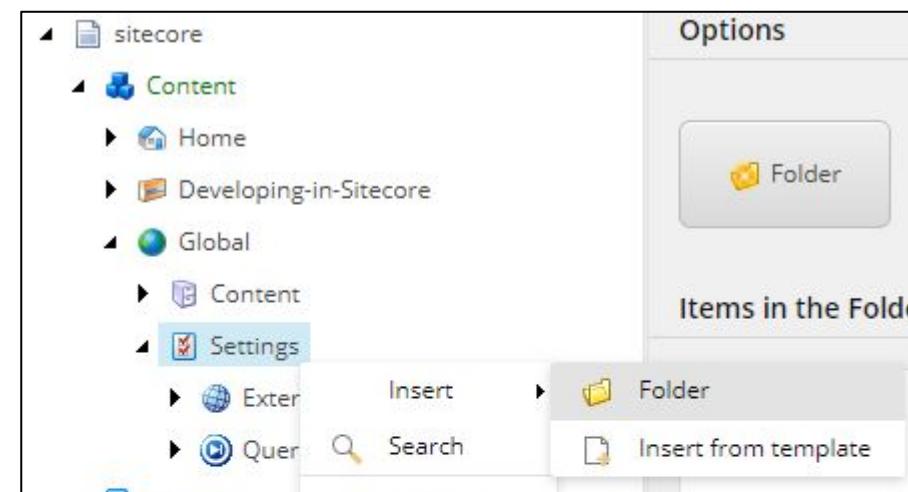
### 10. Save the file

### 11. Keep this file opened in your editor as we will modify it later

# Filtering Search Results in the JS UI (4/9)

## Display Current Site Items Only

12. From the Sitecore Content Editor, select the Settings content item
  - a. /sitecore/Content/Global/Settings
13. Insert a new Folder (right-click > Insert > Folder)
  - a. Name: Coveo Hive Placeholder Extenders
14. In this new folder, insert a new Placeholder Extender item (right-click > Insert > Insert from template)

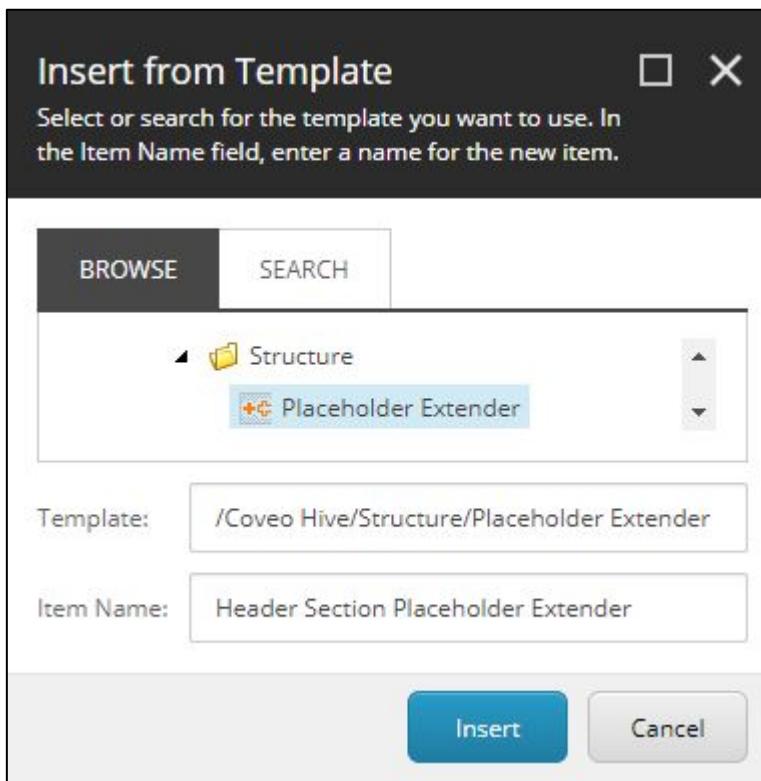


# Filtering Search Results in the JS UI (5/9)

## Display Current Site Items Only

### 15. In the Insert from Template dialog

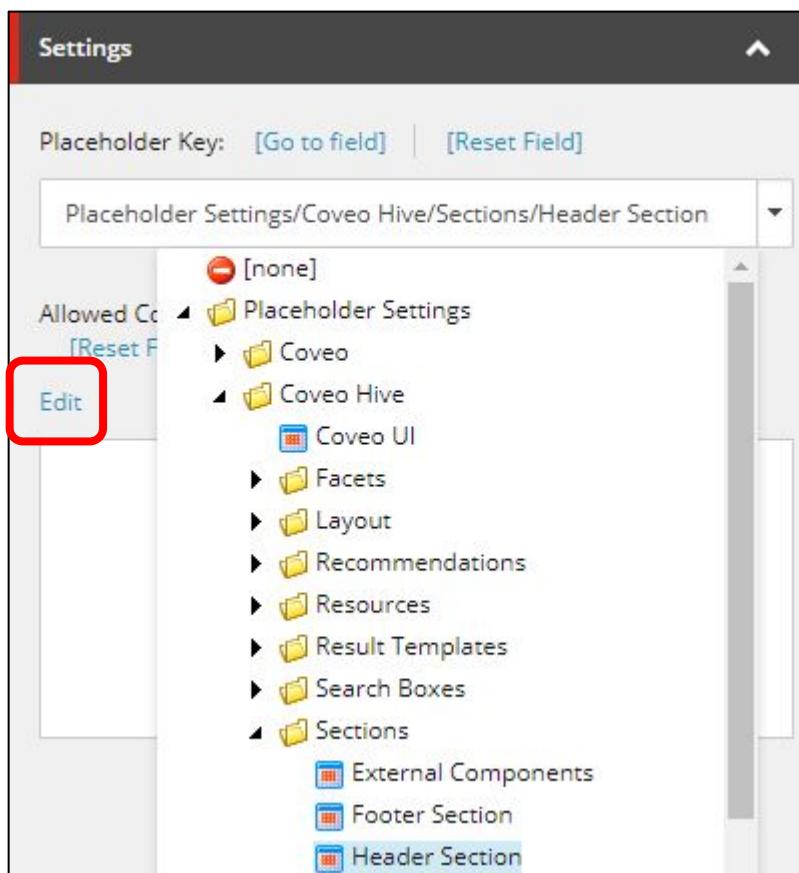
- a. Template: /Templates/Coveo Hive/Structure/Placeholder Extender
- b. Item Name: Header Section Placeholder Extender
- c. Choose Insert



# Filtering Search Results in the JS UI (6/9)

## Display Current Site Items Only

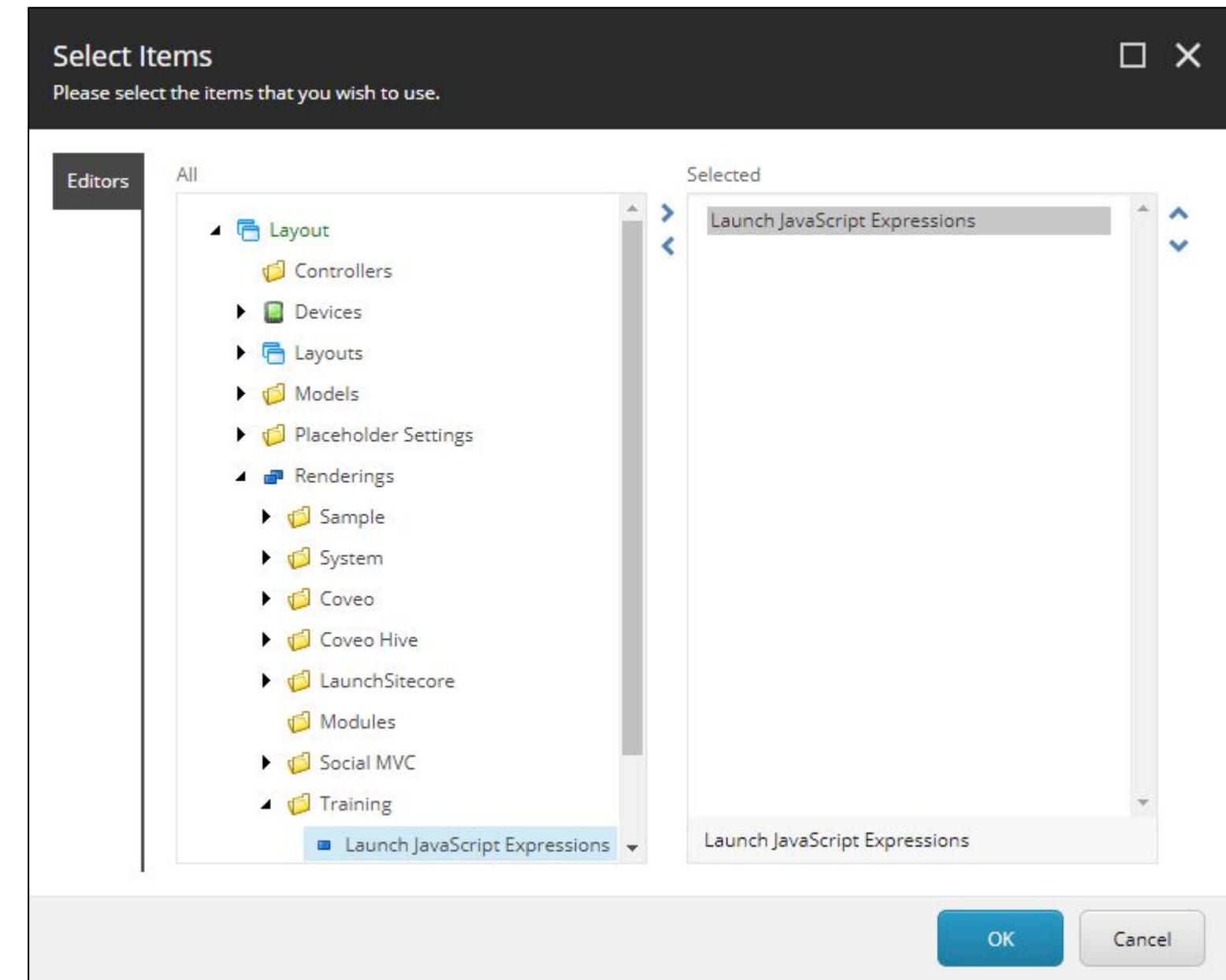
16. In the Placeholder Key field, select the Header Section placeholder
  - a. /Placeholder Settings/Coveo Hive/Sections/Header Section
17. In the Allowed Controls field, click the Edit link



# Filtering Search Results in the JS UI (7/9)

## Display Current Site Items Only

- 18.** In the Select Items dialog
  - a. Select the Launch JavaScript Expressions rendering
    - i. /Layout/Renderings/Training/Launch JavaScript Expressions
  - b. Choose OK
- 19.** Save the item



# Filtering Search Results in the JS UI (8/9)

## Display Current Site Items Only

20. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch

21. Access the Experience Editor (Publish > Experience Editor)

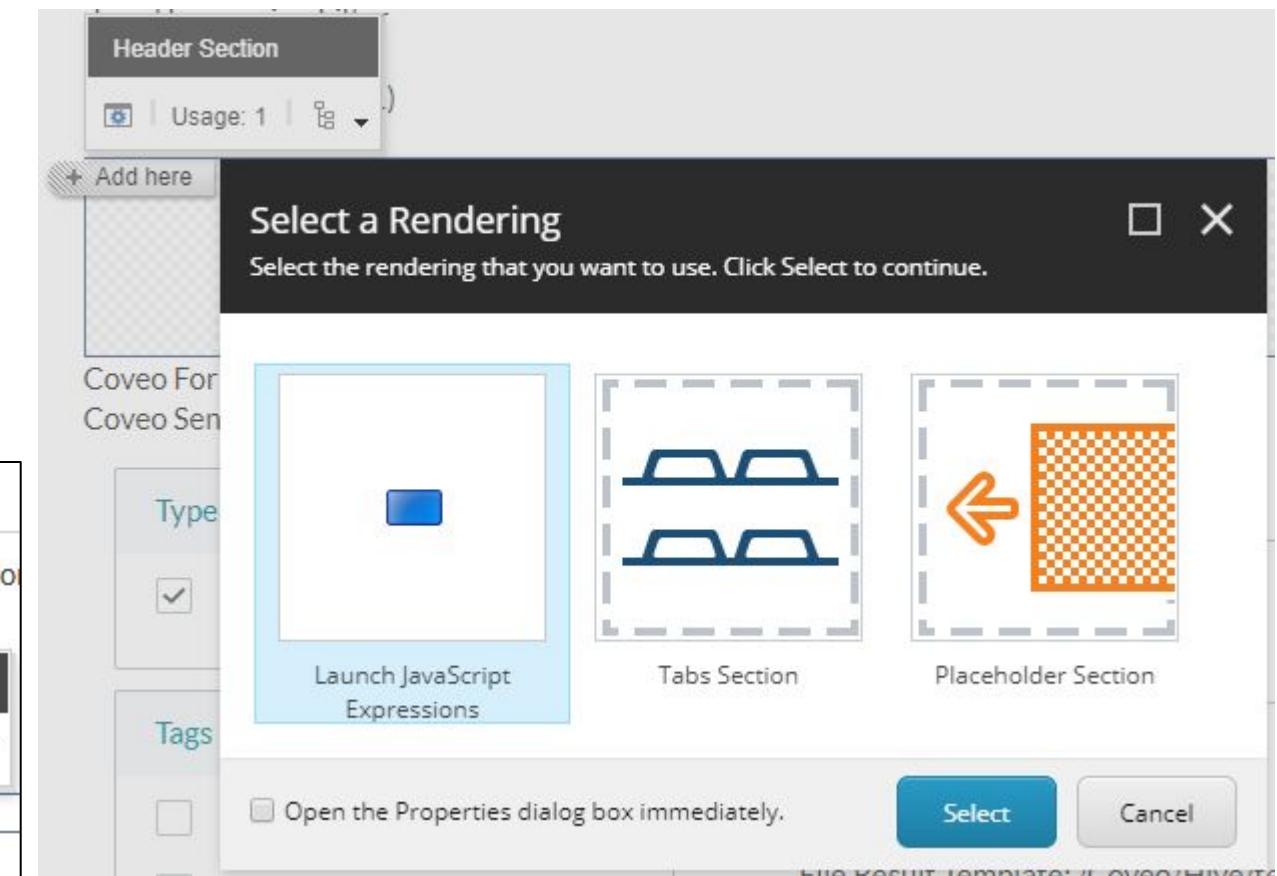
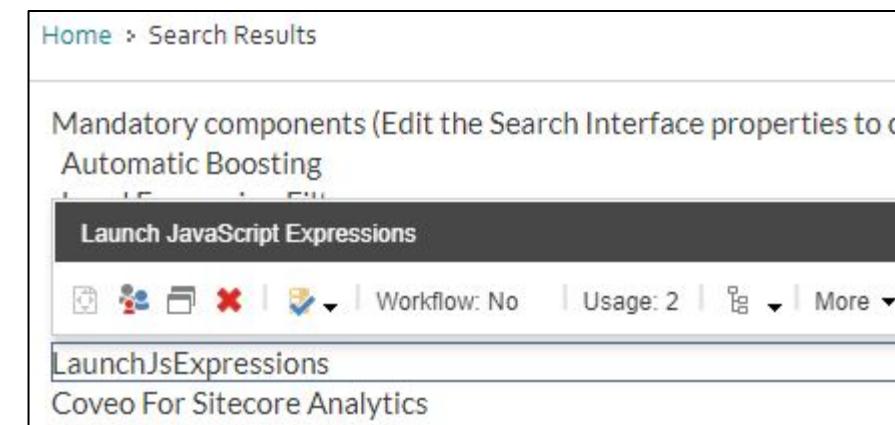
22. Select the Header Section placeholder

23. Click the "Add here" button

24. In the Select a Rendering dialog

- a. Select Launch JavaScript Expressions
- b. Choose Select

25. Save the item



# Filtering Search Results in the JS UI (9/9)

## Display Current Site Items Only

26. Validate that your page is using your filtering rule
  - a. You should see 59 search results

# Anatomy of a Coveo for Sitecore Query With JavaScript Advanced Expression Filters

Advanced Query:

```
(  
  (  
    Filtering Rules from the Properties Editor  
    Item Templates Exclusion from the Properties Editor Options  
  )  
)  
(  
  Advanced Expression Filters Added from the JavaScript Code  
)  
(  
  Automatic Source Filter  
)
```

Constant Query doesn't change:

Various Filters from the Properties Editor Options

# Lab

## 1. Filter search results using JavaScript code

- a. Create a new view rendering
- b. Create the cshtml file
- c. Create a new placeholder extender
- d. Add the new rendering to your search page
- e. Validate the number of results



## Result Ranking

# Result Ranking

The main difference between a search engine and a standard sorted list is the ability to sort documents based on their relevance. This is called result ranking.

There are several tools to affect result ranking, such as the Coveo for Sitecore Ranking Editor. But before we can fully use them, we need to understand how default ranking is calculated.

# Result Ranking Phases - Term Weighting

Term weighting is the first phase. It will add a score to a document based on its proximity with the terms in the user's query. The proximity will be higher if the terms are found:

- In the title
- In the summary
- In the concepts
- In the address (URI)
- With formatting (bold, italic, title...)
- In upper case
- Close to their stemming classes (e.g.: universe/university)

# Result Ranking Phases - Document Weighting

Document weighting will judge the quality of the top 50,000 documents of the previous phase. The most important qualities are:

- Last modified date
- Quality evaluation at indexing time
- Matches the user language
- Query matches the document title

# Result Ranking Phases - Other

Once the first two phases are applied, additional phases are executed on the top 100 documents:

- Collaborative Rating
  - [User-Based Rating](#)
- Term Frequency
  - Frequency of the terms in the free-text searchable body
- Term Proximity
  - Proximity between the searched terms in the documents

# Result Ranking - Summary

The ranking is only done by the index when sorting results by relevance.

Once all the phases are applied, each document will have a score. The search results are sorted by this score.



## Results Boosting

# Filtering/Boosting Search Results

- Filtering
  - Restrict the possible search results at query time by adding query expressions to all the user queries.
  - Example: `@ShowInSearchResults == "1"`
- Boosting
  - Modify the order of the search results by adding or removing points to their ranking score. Points are added to a subset of the search results matching a query expression.
  - Example: `$qre(expression:'@TemplateName == "Product", modifier: '10')`

# Boosting Search Results in the JS UI

## Boosting Article Items

1. From the file system, browse to the Training views folder
  - a. <your sitecore instance folder>\Website\Views\Training
2. Open your "LaunchJsExpressions.cshtml" file
3. Add these lines at the end to add a Query Ranking Expression (QRE)

```
<div id="articlesRankingExpression" class="CoveoForSitecoreRankingExpression"></div>

<script>
    document.addEventListener('DOMContentLoaded', function() {
        var expressionBuilder = new Coveo.ExpressionBuilder();
        var fieldName = CoveoForSitecore.Context.fields.toCoveo('@@_template');
        var articleTemplateShortId = 'd9019e30f95446ccaa703e928c40b5d0';
        expressionBuilder.addFieldExpression(fieldName, '==', [articleTemplateShortId]);
        var qre = "$qre(expression: '" + expressionBuilder.build() + "', modifier: '10')";
        document.getElementById('articlesRankingExpression').dataset.scRankingExpression = qre;
    });
</script>
```

4. Save the file
5. Validate that your page is using your QRE

# Anatomy of a Coveo for Sitecore Query With JavaScript Advanced Expression QREs

Advanced Query:

```
(  
    Advanced Expression QREs Added from the JavaScript Code  
)  
(  
    (  
        Filtering Rules from the Properties Editor  
        Item Templates Exclusion from the Properties Editor Options  
    )  
)  
(  
    Advanced Expression Filters Added from the JavaScript Code  
)  
(  
    Automatic Source Filter  
)
```

Constant Query doesn't change:

Various Filters from the Properties Editor Options

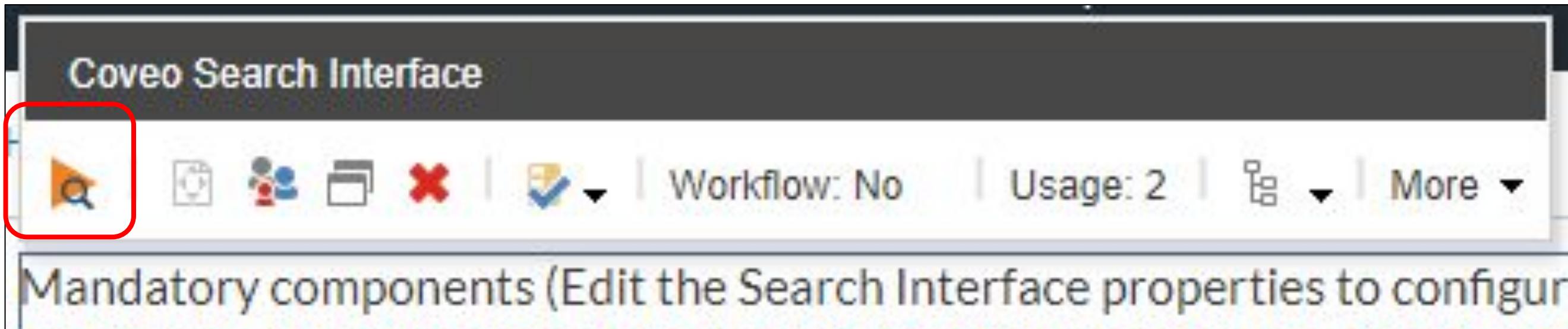
# Lab

## 1. Boost search results using JavaScript code

# Boosting Search Results in the Properties Editor (1/3)

## Negatively Boosting the Home Item

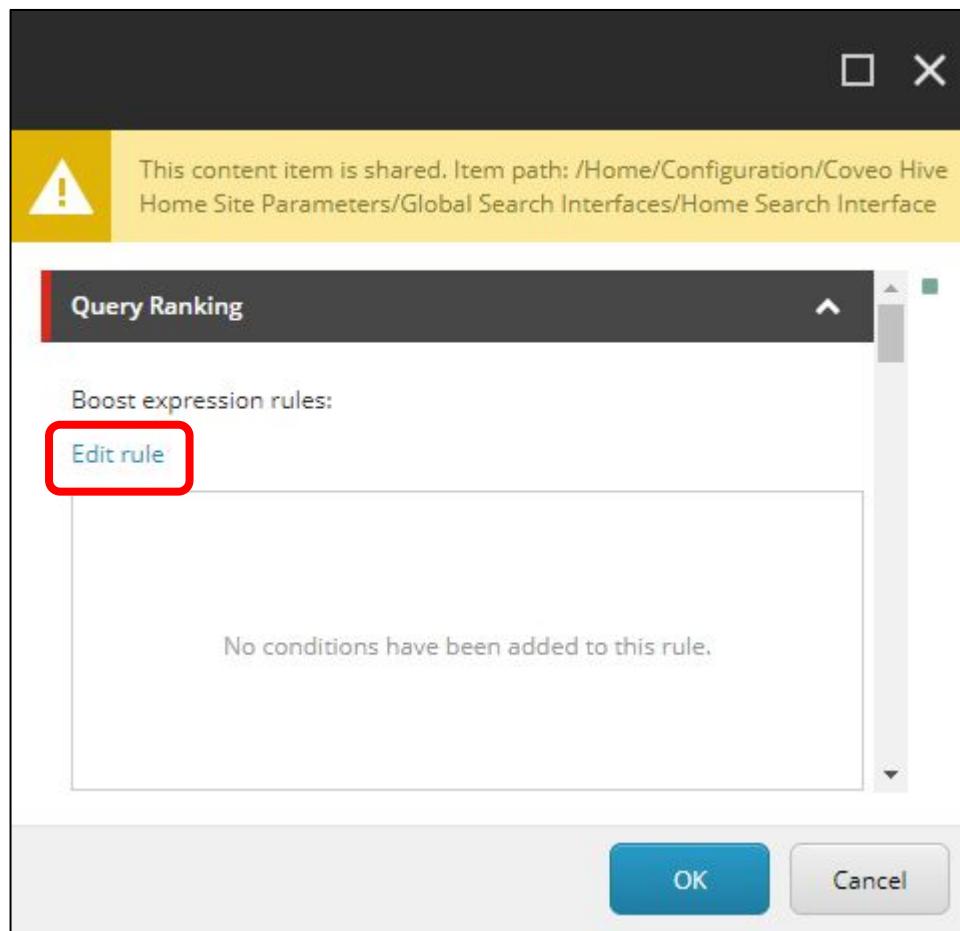
1. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
2. Access the Experience Editor (Publish > Experience Editor)
3. Select the Coveo Search Interface component
4. Click the Coveo experience button in the floating toolbar



# Boosting Search Results in the Properties Editor (2/3)

## Negatively Boosting the Home Item

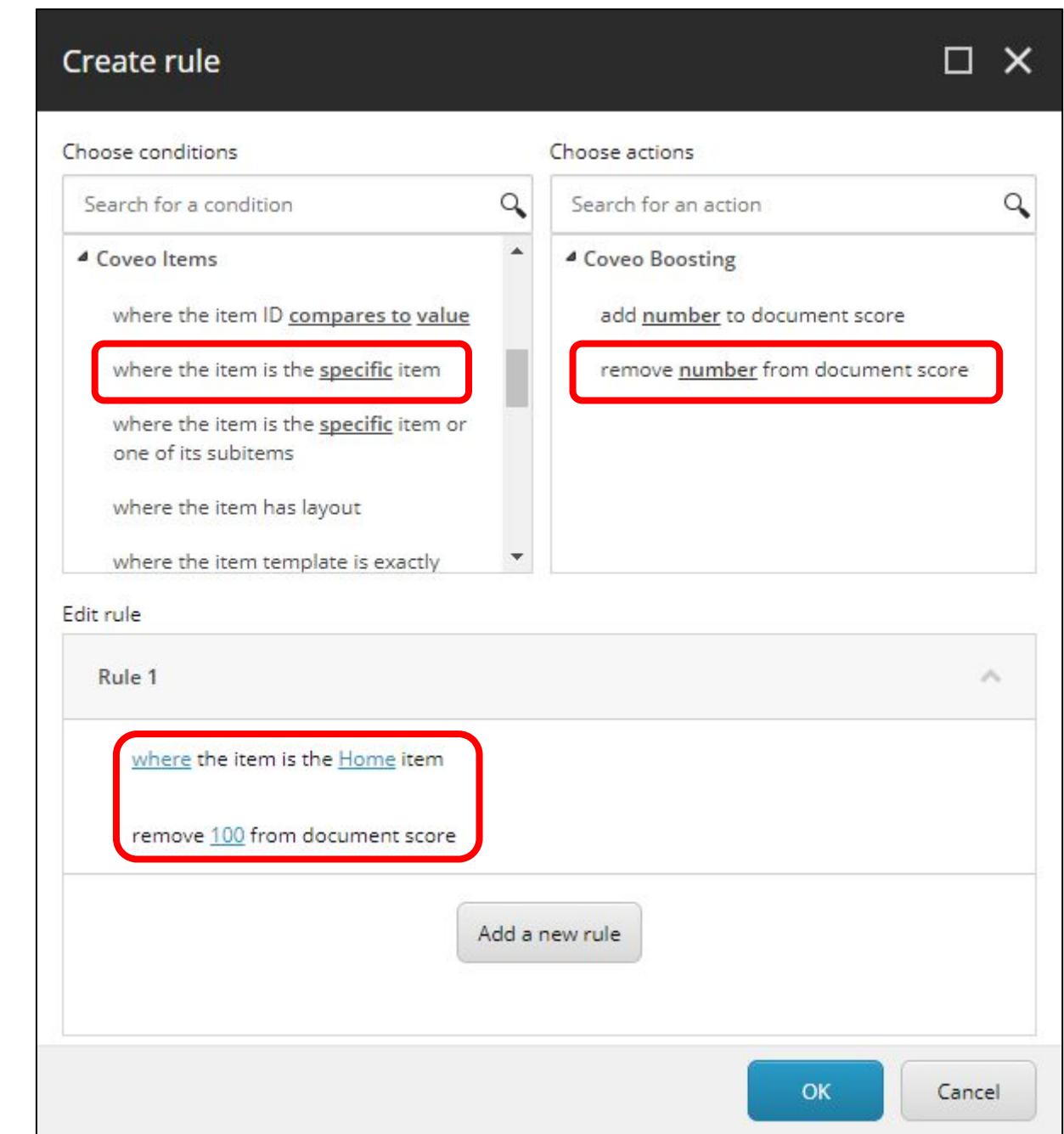
5. In the Control Properties dialog, edit the boost expression rules (Query Ranking > Boost expression rules > Edit rule link)



# Boosting Search Results in the Properties Editor (3/3)

## Negatively Boosting the Home Item

6. In the Create rule dialog, click the
  - a. "where the item is the specific item" condition under Coveo Items
  - b. "remove number from document score" action
7. In the Edit rule bottom section
  - a. Click the links:
    - i. specific: Select "Content/Home" and choose OK
    - ii. number: Enter "100" and choose OK
  - b. Choose OK
8. In the Control Properties dialog, choose OK
9. Validate that your page is using your boosting rule





Results 1-10 of 59

#### [Basic Presentation Explanation - About Launch Sitecore | Launch Sitecore](#)

<sc:Placeholder ID="phHeader" Key="header" runat="server" /> ... This tag identifies the Header Placeholder region in the wireframe above. ... Home ... Audience Landing Pages ... Pattern

#### [Create a Goal - About Launch Sitecore | Launch Sitecore](#)

1) ... Clicking on the Goals button will bring up a checklist that shows all currently ... A screenshot of one of the Wizard steps from WFFM is below: ... Team ... Pattern ... Detailed

#### [The Sitecore UIs - About Launch Sitecore | Launch Sitecore](#)

Adding page Components ... Logging off ... Language: View the page content in another language/locale ... Of course, all these views are independent, so you can choose to view your page in a ...

#### [Create a custom validator - About Launch Sitecore | Launch Sitecore](#)

Field Type Level. ... The screenshot below shows the Required validator and the field that calls out the ... Create an item to represent the validator (as the Required validator does) ... Team

#### [Inline Personalization - About Launch Sitecore | Launch Sitecore](#)

Click a rule to add it to the rule description. ... 6. Click ok and add more conditions if necessary. ... Home ... Audience Landing Pages ... About Launch Sitecore ... Team ... Glossary

#### [The Launch Sitecore Site - About Launch Sitecore | Launch Sitecore](#)

An article section covering the initial strategy considerations for the Launch ... Building the Site ... Optimizing the Experience ... Home ... Audience Landing Pages ... Team ... No Match

#### [Creating a List Control - About Launch Sitecore | Launch Sitecore](#)

The tags for each article shown in the footer also uses this approach. ... <asp:Repeater ID="rptList" runat="server" OnItemDataBound="rptList\_ItemDataBound"> ... <HeaderTemplate> ... Login

#### [Security - Authentication and Authorization - About Launch Sitecore | Launch Sitecore](#)

Contains procedures and best practices and is aimed at introducing new security ... [http://sdn.sitecore.net/upload/sitecore6/security\\_administrators\\_cookbook-usletter.pdf](http://sdn.sitecore.net/upload/sitecore6/security_administrators_cookbook-usletter.pdf) ... Home ... Developer

#### [Minimizing Downtime During Deployment - About Launch Sitecore | Launch Sitecore](#)

Perform your standard backups. ... This allows our two CD servers to continue to serve the existing site. ... Apply all of the file changes to CD2. ... Add CD2 back to the cluster. ... Pattern

# Anatomy of a Coveo for Sitecore Query With Properties Editor Boosting Rules

Advanced Query:

```
(  
  Boosting Rules from the Properties Editor  
)  
(  
  Advanced Expression QREs Added from the JavaScript Code  
)  
(  
  (  
    Filtering Rules from the Properties Editor  
    Item Templates Exclusion from the Properties Editor Options  
)  
)  
(  
  Advanced Expression Filters Added from the JavaScript Code  
)  
(  
  Automatic Source Filter  
)
```

Constant Query doesn't change:

Various Filters from the Properties Editor Options

# Lab

## 1. Boost search results using the properties editor

# Using the Debug Panel (1/5)

1. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
2. Access the Experience Editor (Publish > Experience Editor)
3. Search for "Content Editor"
4. The first result should be the Content Editor glossary page
5. Hold the left ALT key and double click in the white space besides the result title or after the excerpt
  - a. A modal box should appear
6. Hide the Sitecore Experience Editor menu by clicking the top right red button 
7. Check the "Enable Query Debug" check box

# Using the Debug Panel (2/5)

You should now see a rankingInfo section containing the following:

Document weights:

Title: 856; Quality: 180; Date: 82; Adjacency: 1500; Source: 500; Custom: 350; Collaborative rating: 0; QRE: 0; Ranking functions: 0;

Terms weights:

editor: 100, 100;

Title: 739; Concept: 92; Summary: 277; URI: 462; Formatted: 184; Casing: 0; Relation: 184; Frequency: 280;

content: 100, 100;

Title: 60; Concept: 0; Summary: 22; URI: 37; Formatted: 15; Casing: 0; Relation: 15; Frequency: 0;

Total weight: 5835

These are the details on the ranking of your result.

The total weight will define its position on the result list.

# Using the Debug Panel (3/5)

The document weight contains information on the result itself

- Title
  - How many terms it contains
- Quality
  - A subjective rating done by the index which might be removed in the near future
- Date
  - If the item was modified recently
- Adjacency
  - If the terms found on the document are close of each other
- QRE
  - Custom Query Ranking Expressions similar to the one we use for our Home item

# Using the Debug Panel (4/5)

- Custom and Source
  - Constant numbers no longer in use
- Ranking Functions
  - Like QRE but using mathematical functions
- Collaborative ranking
  - User-based ranking

# Using the Debug Panel (5/5)

The term weight contains information on each of the terms (keywords) in the query

- Title
  - If the term is found in the title of the document
- Concept, summary and URI
  - If the term is found in one of these three attributes of the result
- Formatted and Casing
  - If the terms found in the document are accentuated using formatting or casing
- Relation
  - If the word found is exactly the term, or a stemmed equivalent
- Frequency
  - How often the term is found, this will most likely be 0 in small indexes, even if the term is found several time

# Lab

1. Open the debug panel
2. Enable query debug and inspect the ranking info



## Important Search Page Components

# Coveo Facet, Sort, and Tab Components

## Facets

-  Coveo Date Facet Range
-  Coveo Date Facet Slider
-  Coveo Distance Facet Range
-  Coveo Facet
-  Coveo Numeric Facet Range
-  Coveo Numeric Facet Slider

## Sorts

-  Coveo Date Sort
-  Coveo Distance Sort
-  Coveo Field Sort
-  Coveo Relevancy Sort

## Tabs

-  Coveo Tab

# Coveo Section, Scopes, and Various Components

## Sections

-  External Components Section
-  Facets Section
-  Global Search Section
-  Placeholder Section
-  Recommendations Section
-  Results Header Section
-  Results Layout Section
-  Results Section
-  Results Sorts Section
-  Results Summary Section
-  Search Section
-  Tabs Section

## Components

-  Coveo Did You Mean
-  Coveo Pager
-  Coveo Query Summary
-  Coveo Results Per Page
-  Coveo Breadcrumbs
-  Coveo Error Report
-  Coveo Query Duration
-  Coveo Result Layout
-  Coveo Settings Button

## Scopes

-  Automatic Boosting
-  External Content
-  Query Filter
-  Query Ranking
-  Sitecore Instance Filter

# Other Notable Coveo Components

## Analytics

-  Coveo Analytics
-  Coveo For Sitecore Analytics
-  Coveo Page View Analytics
-  Coveo Send Analytics To Sitecore

## Resources

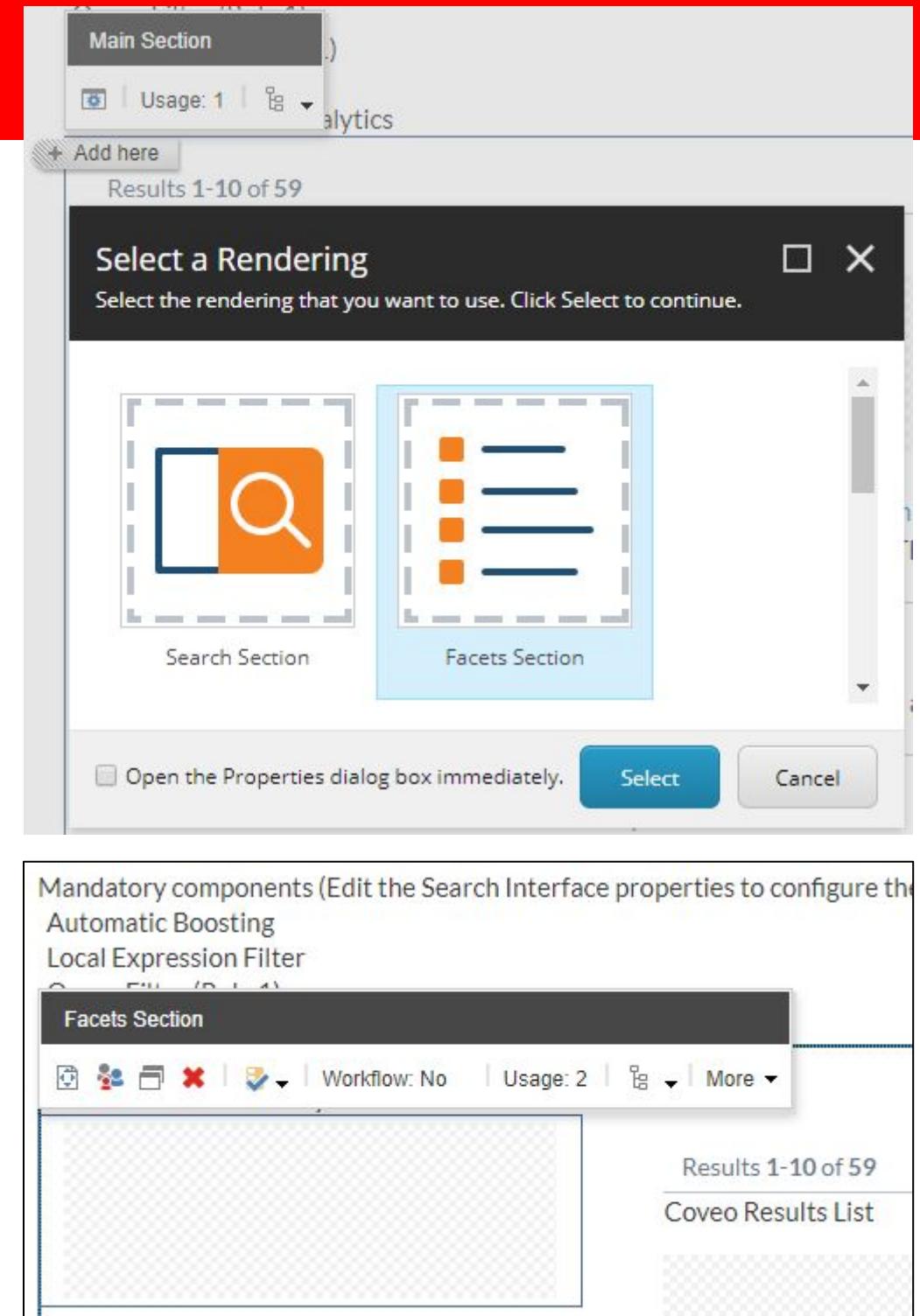
-  Coveo Distance Resources
-  Coveo Search Resources
-  Coveo User Context

## Search Interfaces

-  Coveo Recommendations
-  Coveo Related Query
-  Coveo Search Interface

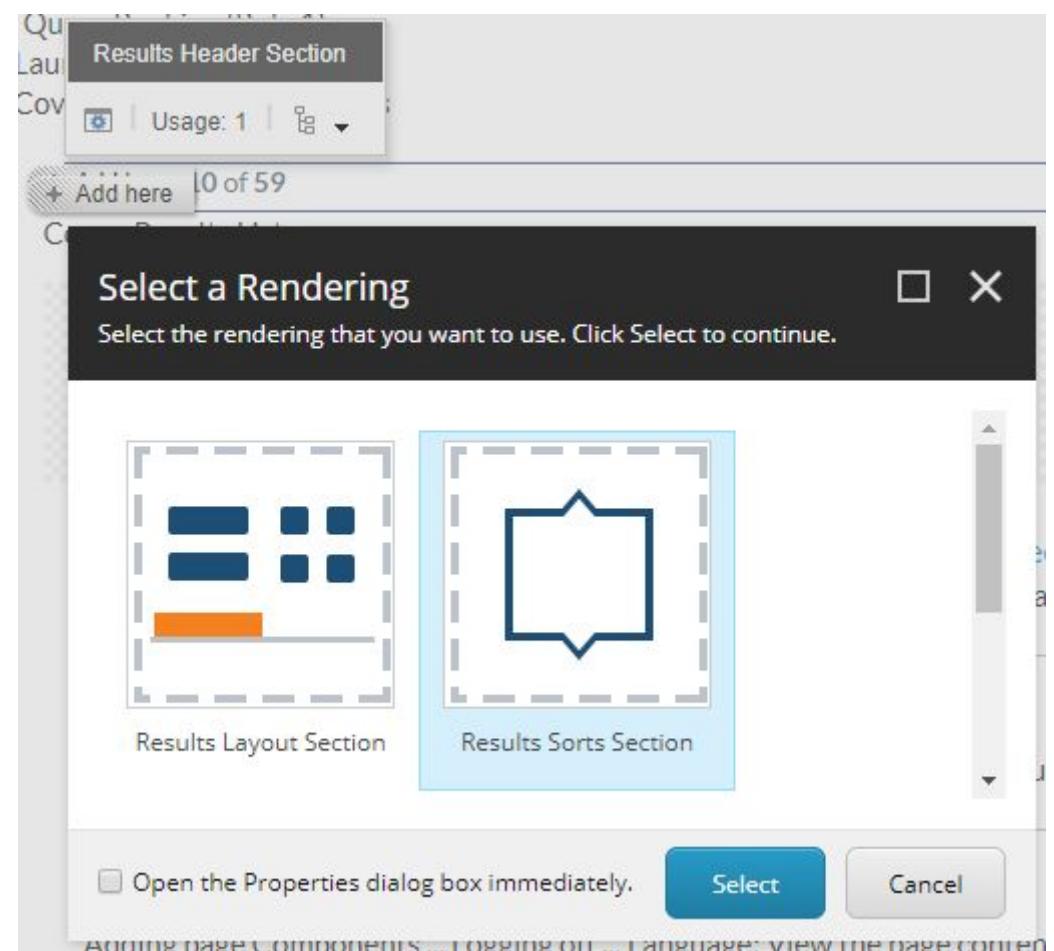
# Add Important Search Interface Components (1/6)

1. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
2. Access the Experience Editor (Publish > Experience Editor)
3. Select the Main Section placeholder
4. Click the first "Add here" button
5. In the Select a Rendering dialog
  - a. Select Facets Section
  - b. Choose Select



# Add Important Search Interface Components (2/6)

6. Select the Results Header Section placeholder
7. Click the last "Add here" button
8. In the Select a Rendering dialog
  - a. Select Results Sorts Section
  - b. Choose Select

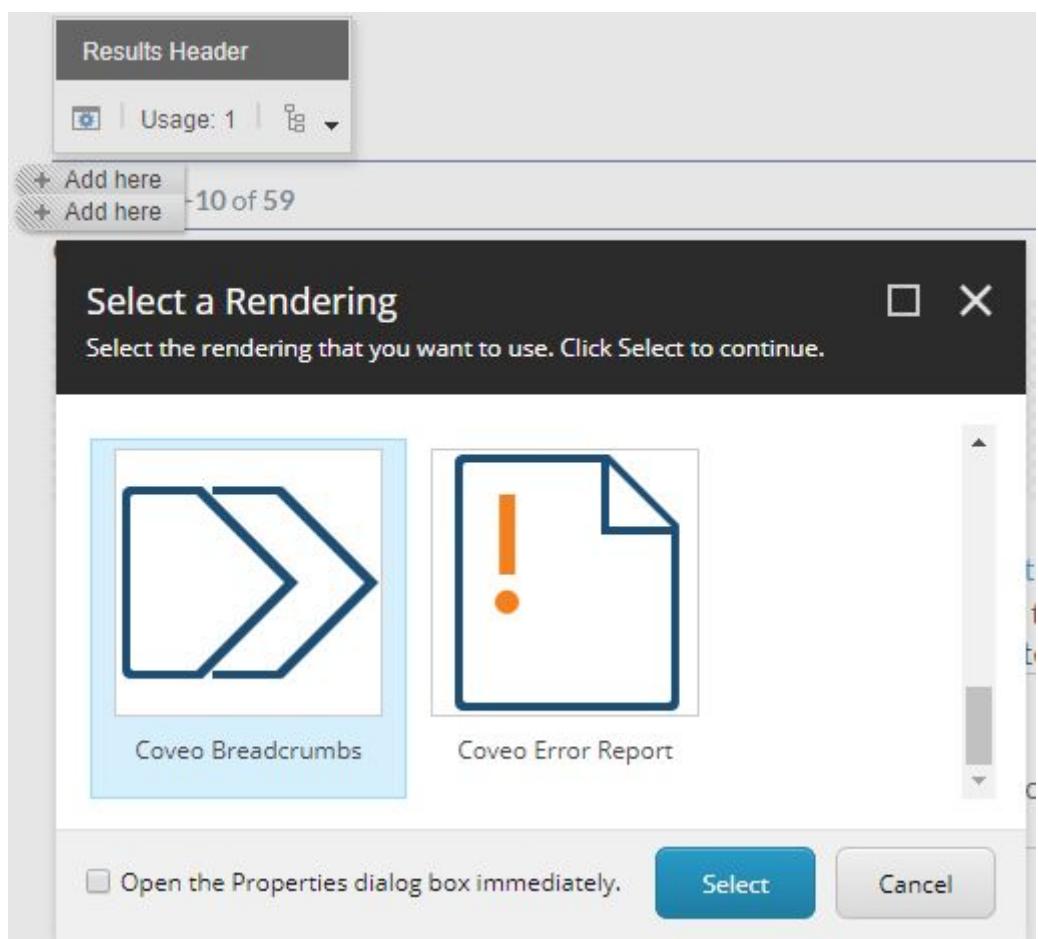


Results 1-10 of 59

Coveo Results List

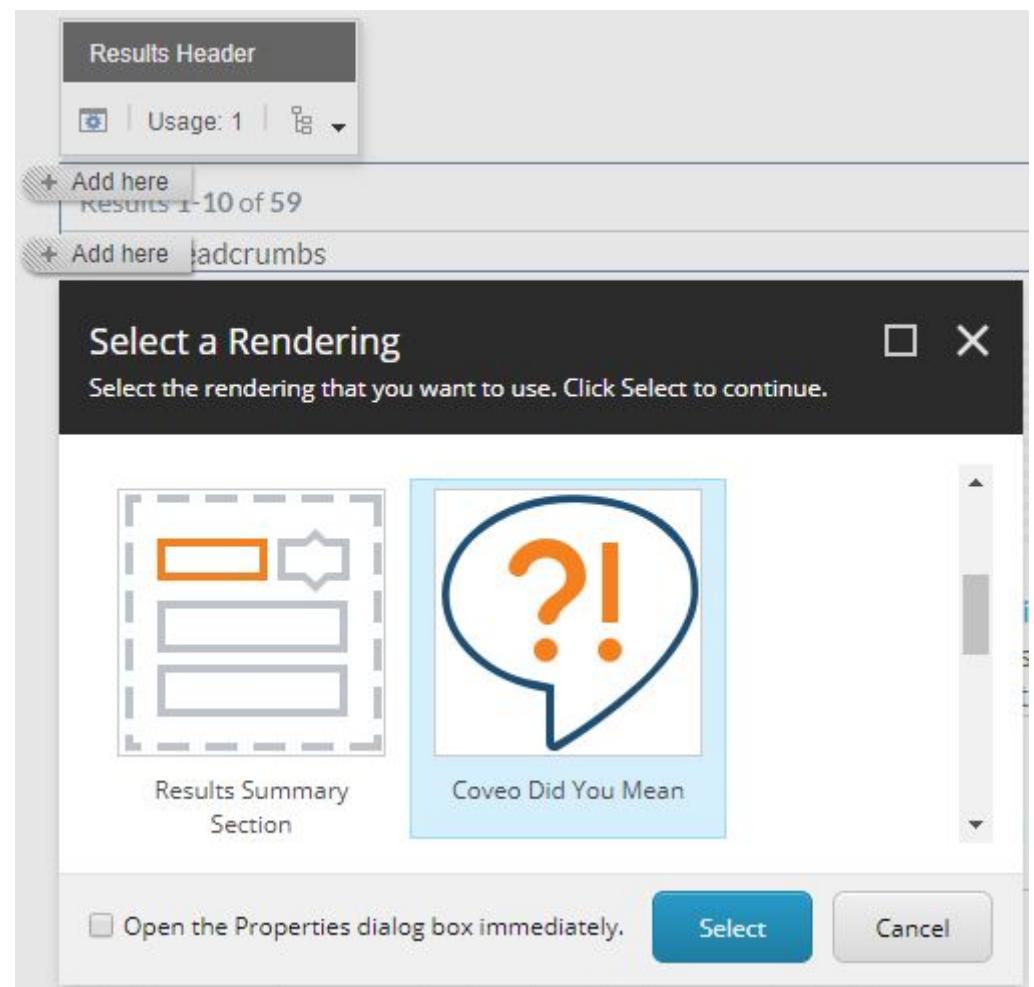
# Add Important Search Interface Components (3/6)

9. Select the Results Header placeholder
10. Click the last "Add here" button
11. In the Select a Rendering dialog
  - a. Select Coveo Breadcrumbs
  - b. Choose Select



# Add Important Search Interface Components (4/6)

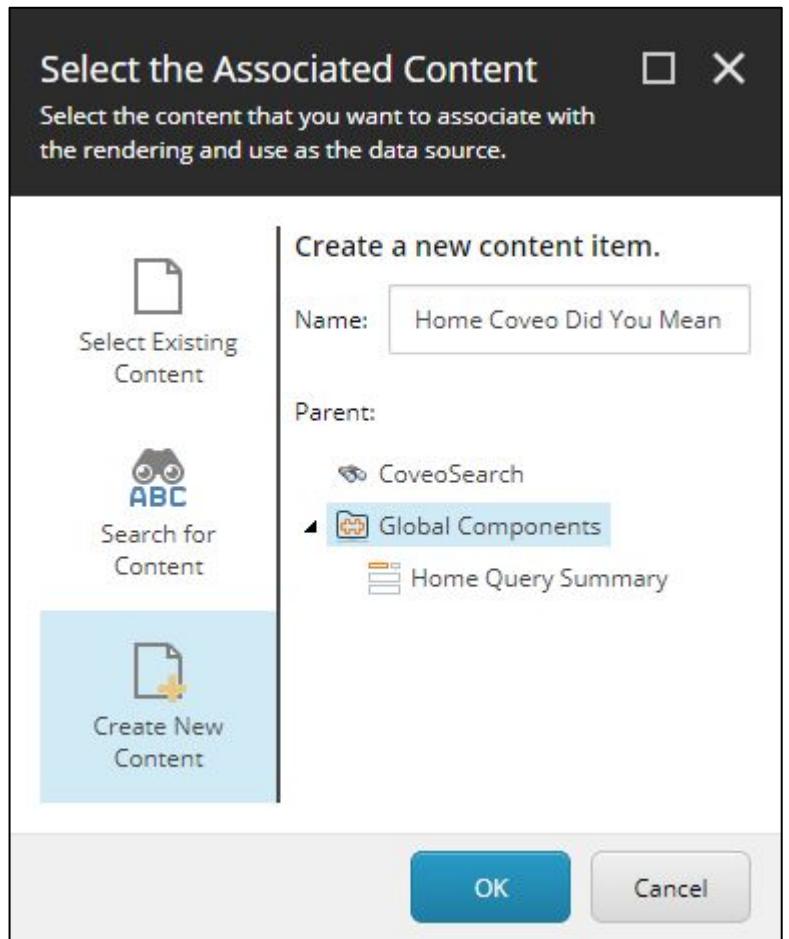
12. Select the Results Header placeholder
13. Click the last "Add here" button
14. In the Select a Rendering dialog
  - a. Select Coveo Did You Mean
  - b. Choose Select



# Add Important Search Interface Components (5/6)

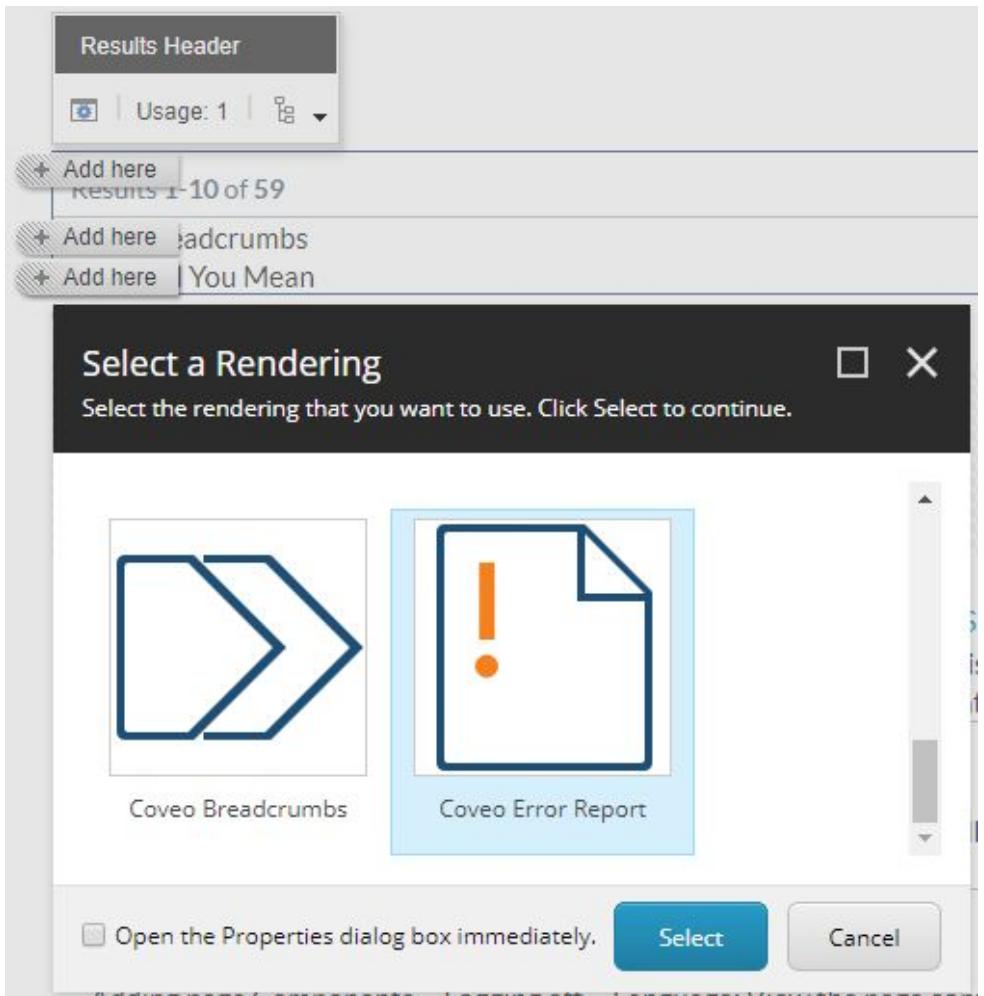
## 15. In the Select the Associated Content dialog

- a. In the left column, select Create New Content
- b. Name: Home Coveo Did You Mean
- c. Parent: Global Components
- d. Choose OK



# Add Important Search Interface Components (6/6)

16. Select the Results Header placeholder
17. Click the last "Add here" button
18. In the Select a Rendering dialog
  - a. Select Coveo Error Report
  - b. Choose Select
19. Save the item



# Lab

## 1. Add the following components

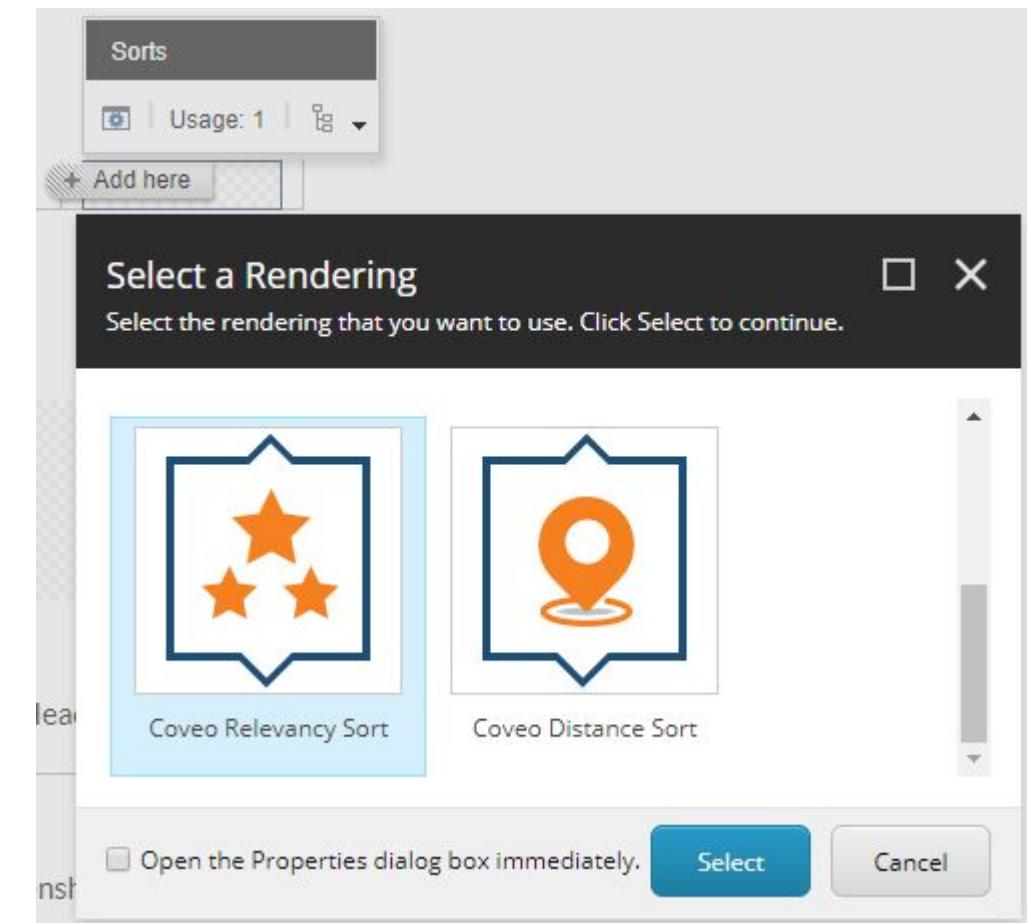
- a. Facets Section
- b. Results Sorts Section
- c. Coveo Breadcrumbs
- d. Coveo Did You Mean
- e. Coveo Error Report



## Sort and Facet Components

# Adding a Coveo Relevancy Sort Component (1/2)

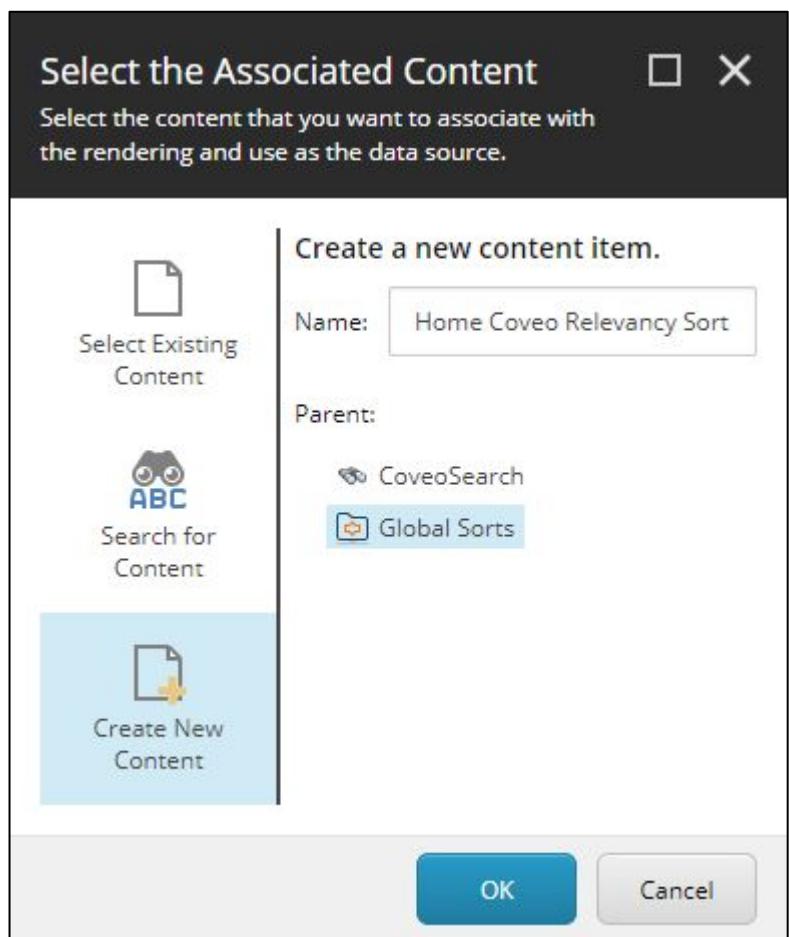
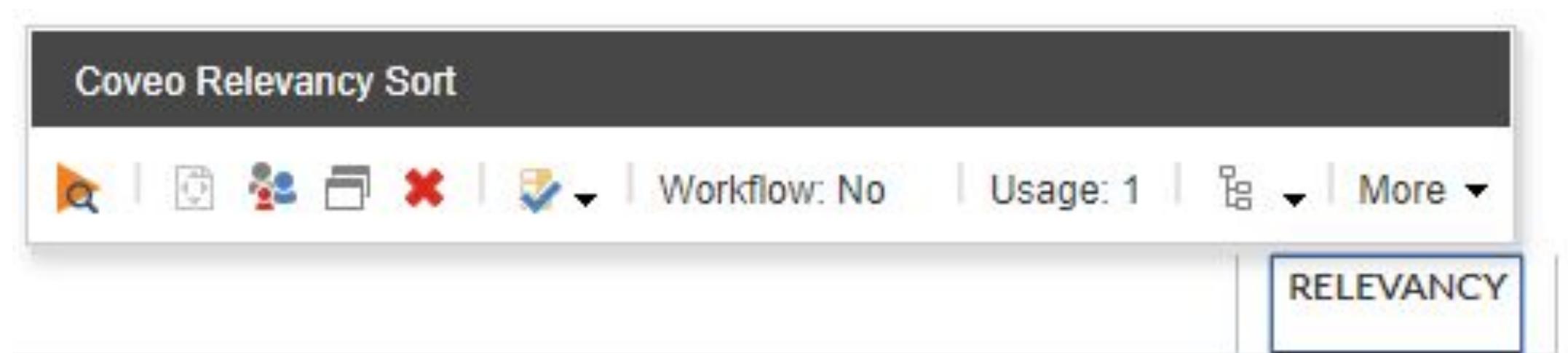
1. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
2. Access the Experience Editor (Publish > Experience Editor)
3. Select the Sorts placeholder
4. Click the "Add here" button
5. In the Select a Rendering dialog
  - a. Select Coveo Relevancy Sort
  - b. Choose Select



# Adding a Coveo Relevancy Sort Component (2/2)

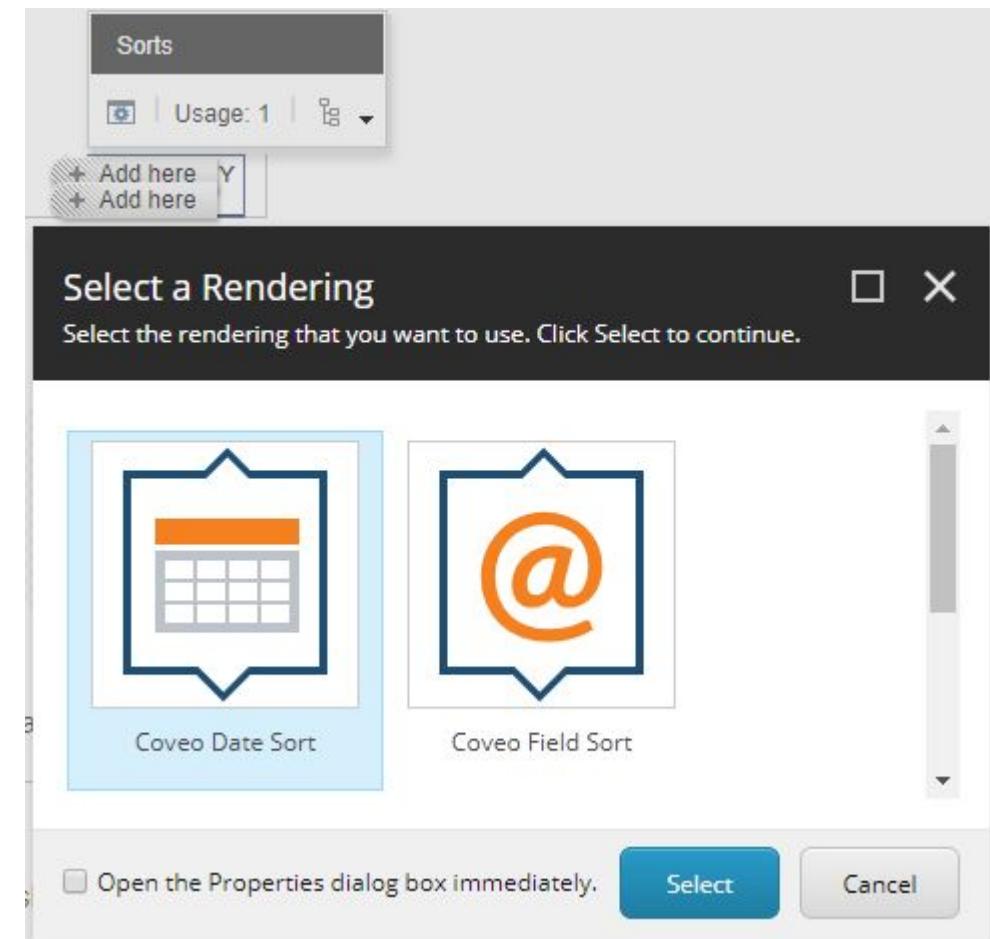
## 6. In the Select the Associated Content dialog

- a. In the left column, select Create New Content
- b. Name: Home Coveo Relevancy Sort
- c. Parent: Global Sorts
- d. Choose OK



# Adding a Coveo Date Sort Component (1/2)

1. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
2. Access the Experience Editor (Publish > Experience Editor)
3. Select the Sorts placeholder
4. Click the last "Add here" button
5. In the Select a Rendering dialog
  - a. Select Coveo Date Sort
  - b. Choose Select



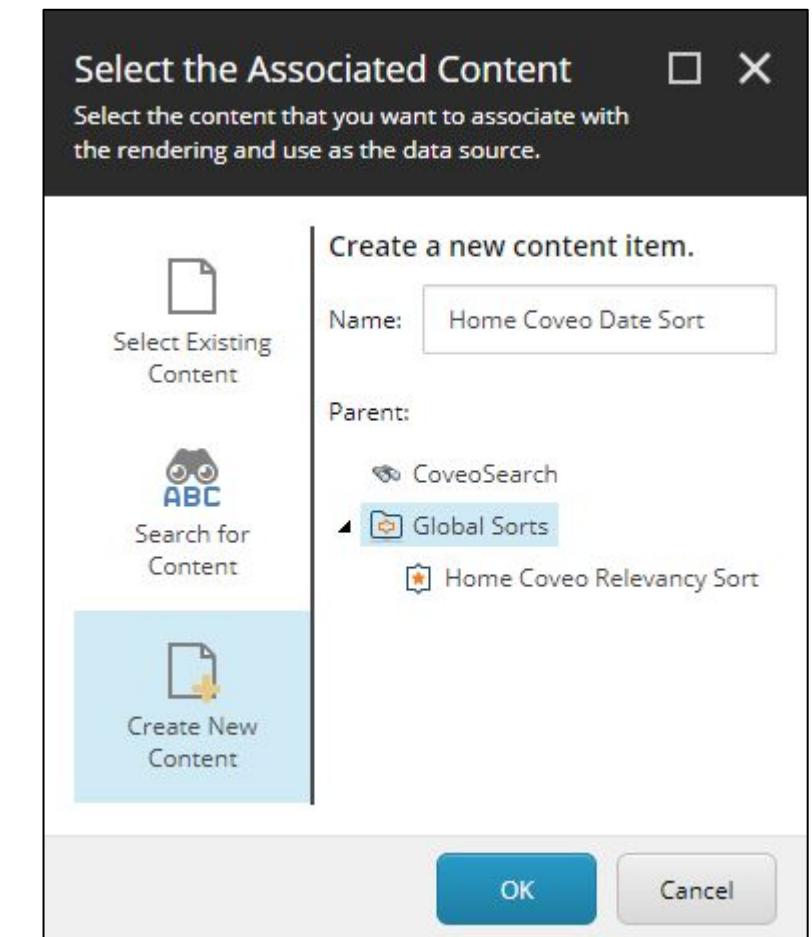
# Adding a Coveo Date Sort Component (2/2)

## 6. In the Select the Associated Content dialog

- a. In the left column, select Create New Content
- b. Name: Home Coveo Date Sort
- c. Parent: Global Sorts
- d. Choose OK

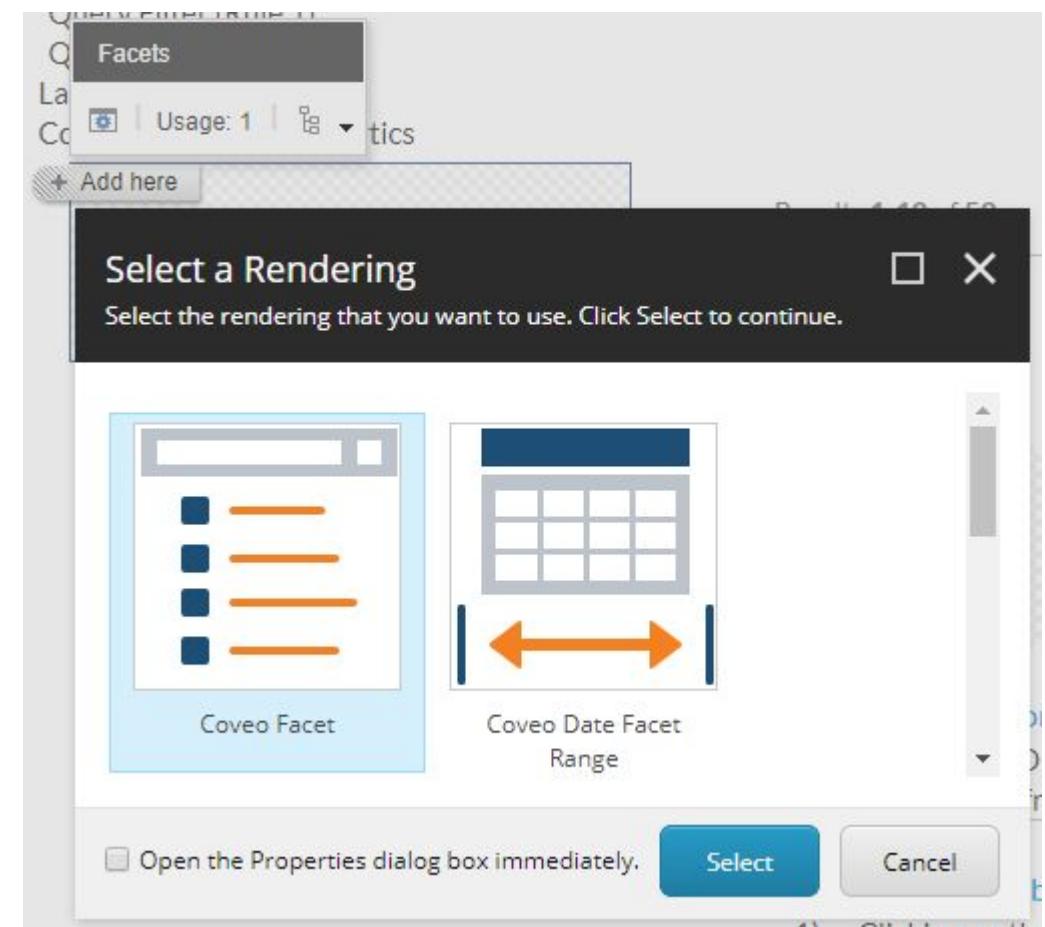
## 7. Save the item

## 8. Validate the sort components behavior by clicking on them



# Adding a Facet for the Items Type (1/4)

1. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
2. Access the Experience Editor (Publish > Experience Editor)
3. Select the Facets placeholder
4. Click the "Add here" button
5. In the Select a Rendering dialog
  - a. Select Coveo Facet
  - b. Choose Select

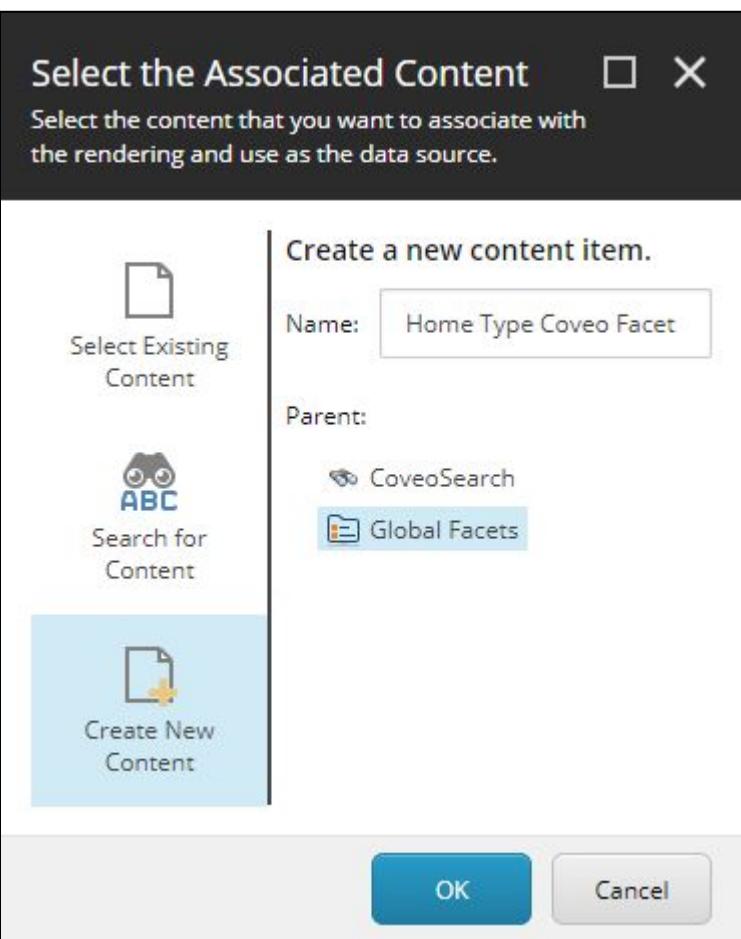
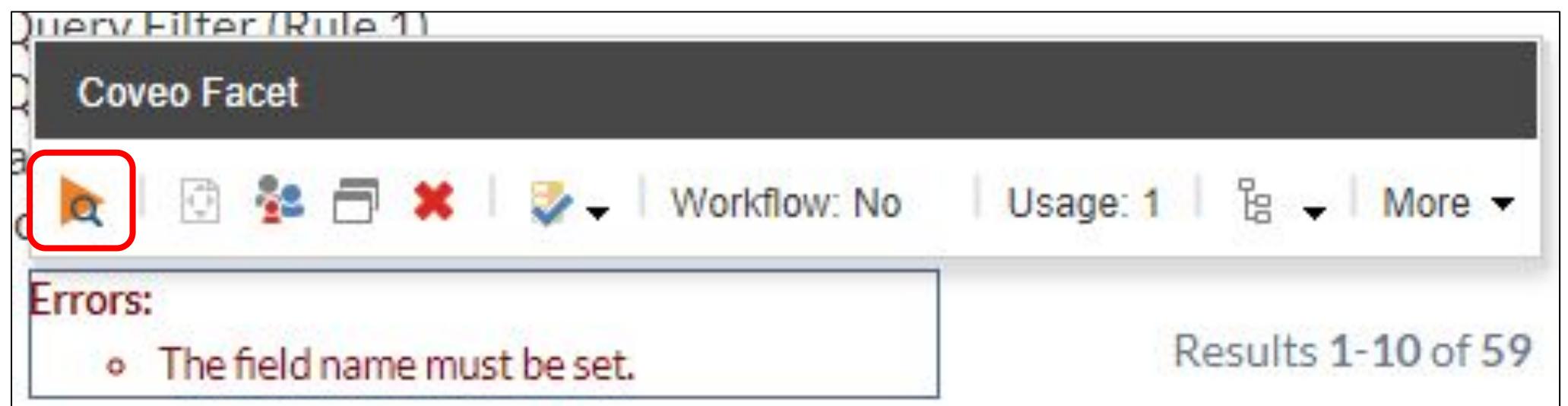


# Adding a Facet for the Items Type (2/4)

## 6. In the Select the Associated Content dialog

- In the left column, select Create New Content
- Name: Home Type Coveo Facet
- Parent: Global Facets
- Choose OK

## 7. Click the Coveo experience button in the floating toolbar



# Adding a Facet for the Items Type (3/4)

## 8. In the Control Properties dialog

### a. Basic Settings

- i. Title: Enter "Type"
- ii. Field: Click the "Select Field" link, then type "template", and select "\_templatename" in the suggestions

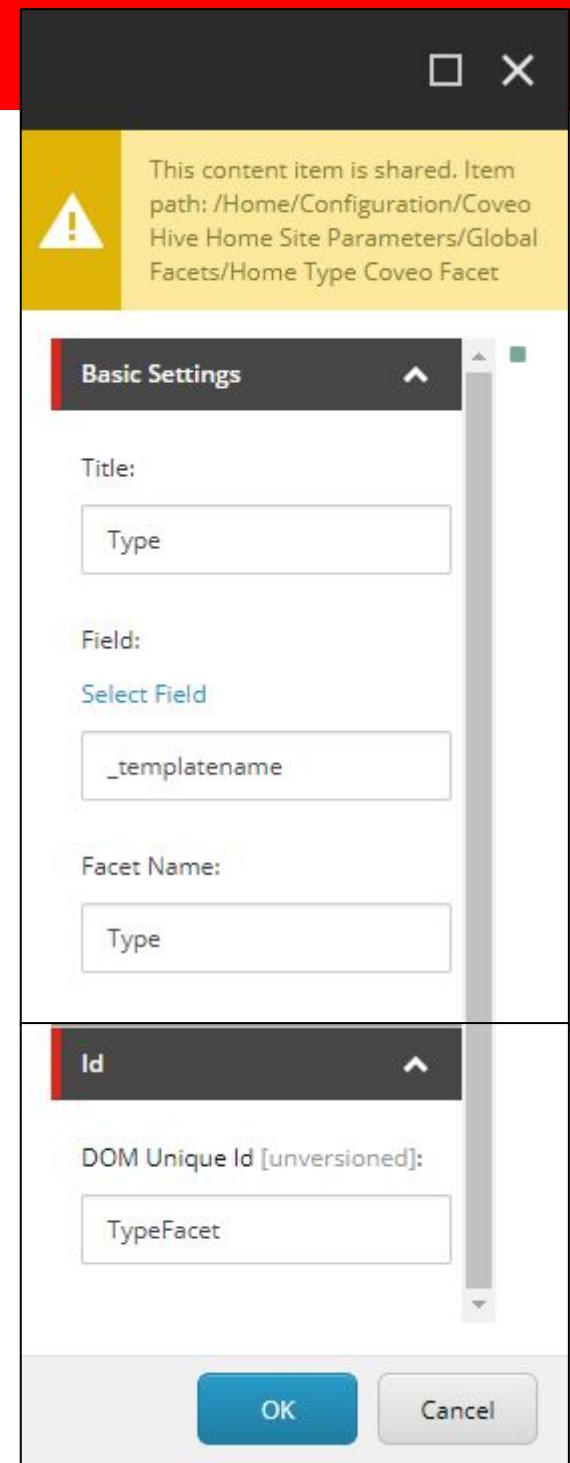
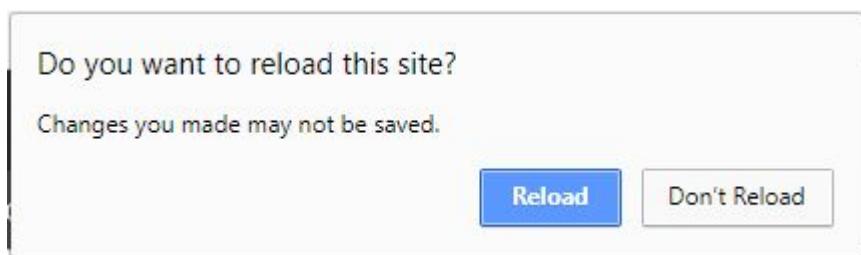
### b. Id

- i. DOM Unique Id: Replace the GUID by "TypeFacet"

### c. Choose OK

## 9. Your browser might ask if you want to reload the page

### a. Choose Don't Reload



# Adding a Facet for the Items Type (4/4)

10. Save the item
11. Validate the facet components behavior

A screenshot of a user interface component titled "Type". It displays a list of items with checkboxes and counts. At the bottom, there is a "Search" button with a magnifying glass icon.

Type	Count
Article	21
Term	16
Team Member	7
Article Group	4
Audience Segment	4

**Search**

# Anatomy of a Coveo for Sitecore Query With Facet Selection

Advanced Query:

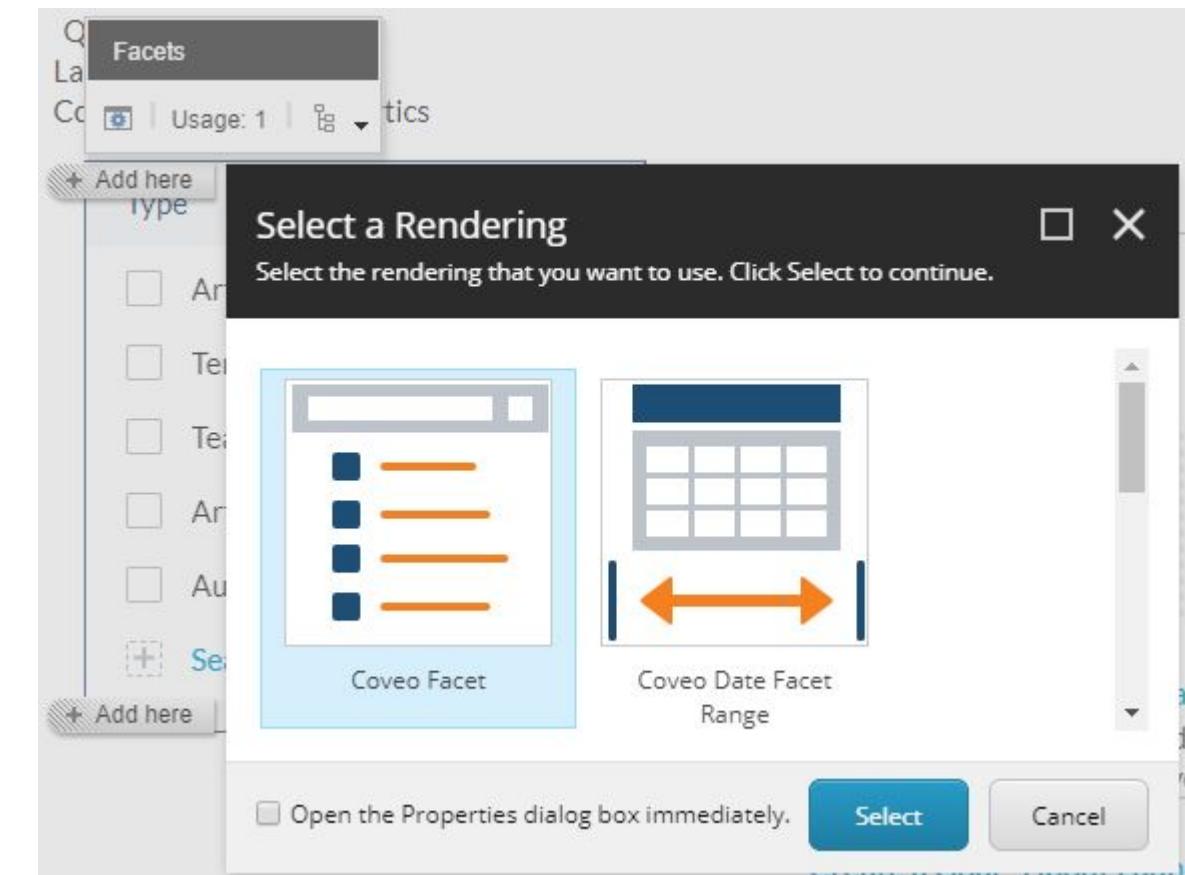
```
(  
    Boosting Rules from the Properties Editor  
)  
(  
    Advanced Expression QREs Added from the JavaScript Code  
)  
(  
    (  
        Filtering Rules from the Properties Editor  
        Item Templates Exclusion from the Properties Editor Options  
)  
)  
(  
    Advanced Expression Filters Added from the JavaScript Code  
)  
(  
    Facet Selection Expressions  
)  
(  
    Automatic Source Filter  
)
```

Constant Query doesn't change:

Various Filters from the Properties Editor Options

# Adding a Facet for the Items Tags (1/4)

1. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
2. Access the Experience Editor (Publish > Experience Editor)
3. Select the Facets placeholder
4. Click the last "Add here" button
5. In the Select a Rendering dialog
  - a. Select Coveo Facet
  - b. Choose Select

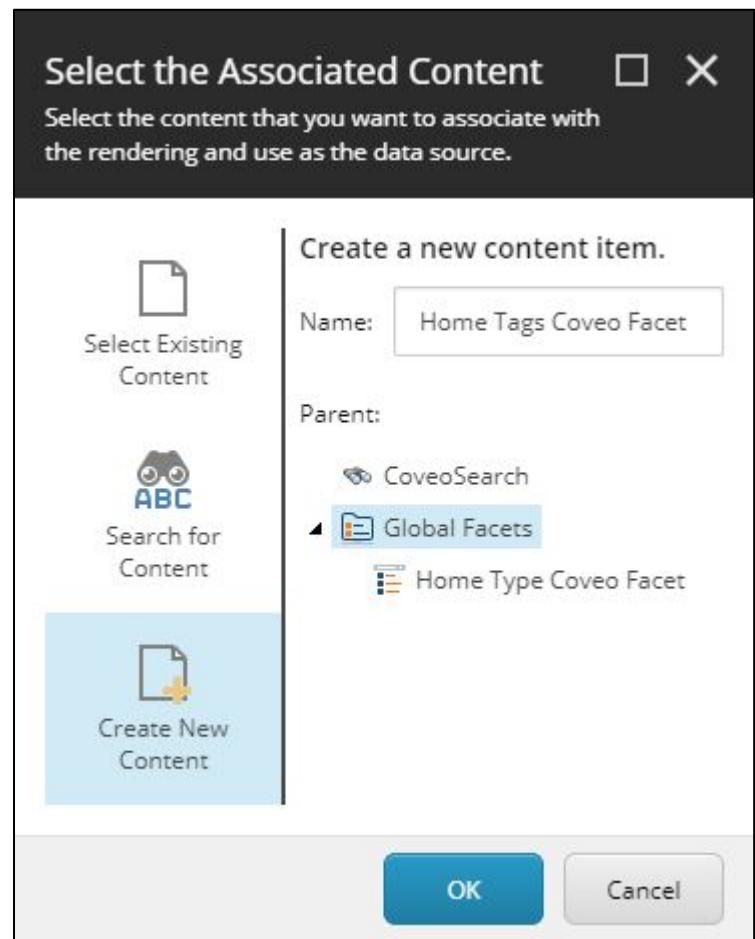
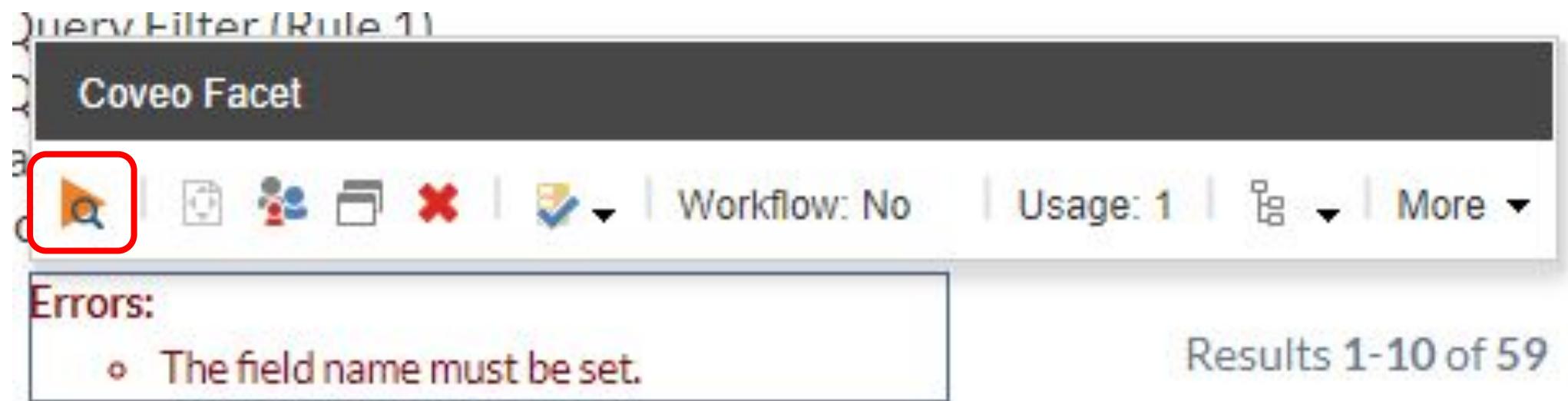


# Adding a Facet for the Items Tags (2/4)

## 6. In the Select the Associated Content dialog

- In the left column, select Create New Content
- Name: Home Tags Coveo Facet
- Parent: Global Facets
- Choose OK

## 7. Click the Coveo experience button in the floating toolbar



# Adding a Facet for the Items Tags (3/4)

## 8. In the Control Properties dialog

### a. Basic Settings

- i. Title: Enter "Tags"
- ii. Field: Click the "Select Field" link, then type "semantics", and select "\_\_semantics" in the suggestions

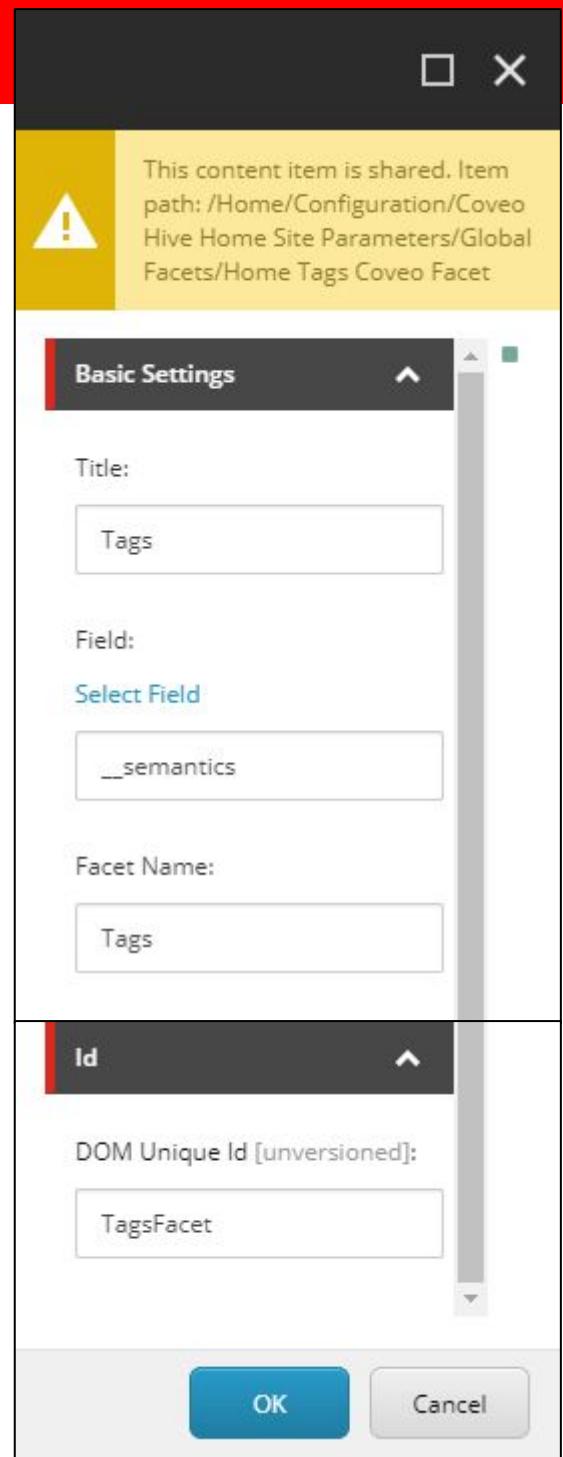
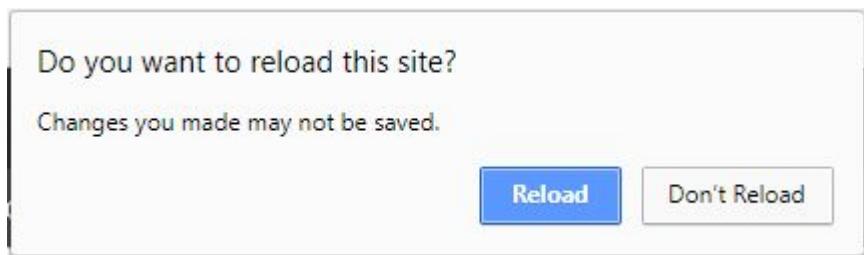
### b. Id

- i. DOM Unique Id: Replace the GUID by "TagsFacet"

### c. Choose OK

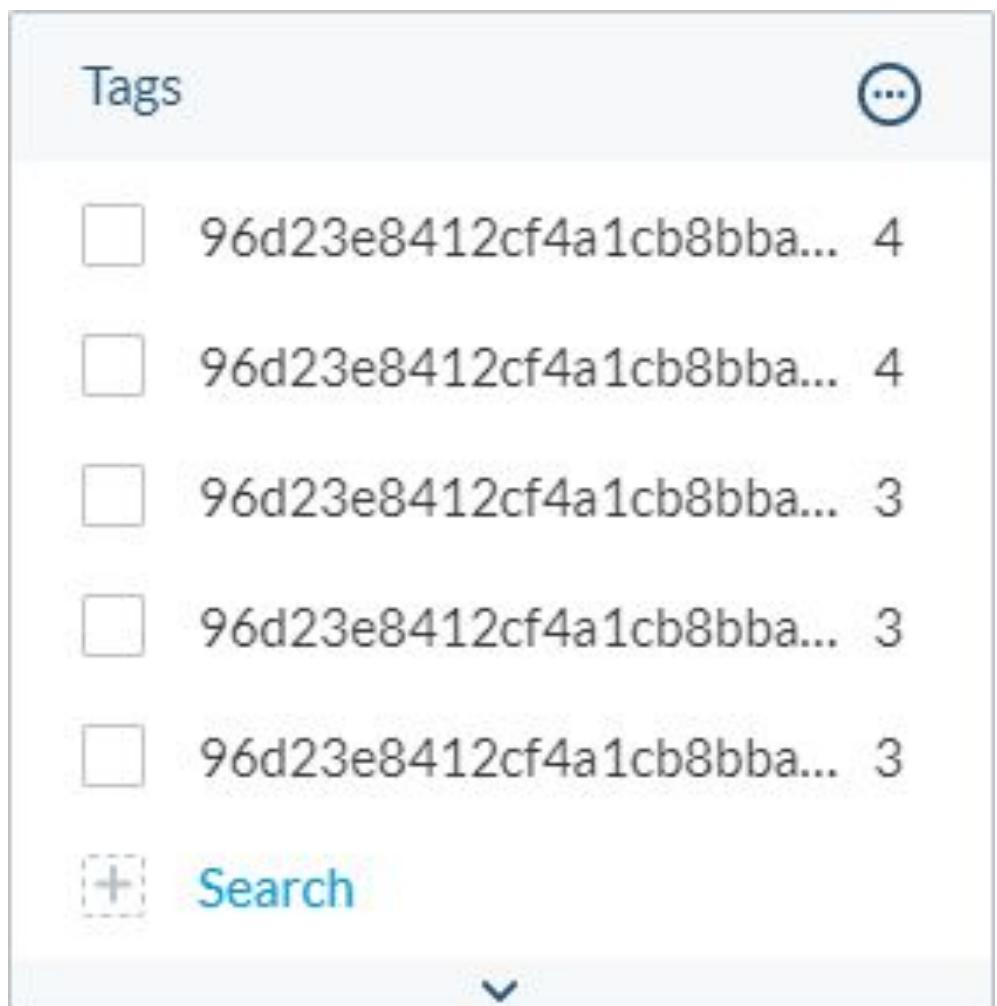
## 9. Your browser might ask if you want to reload the page

### a. Choose Don't Reload



# Adding a Facet for the Items Tags (4/4)

10. Save the item
11. Validate the facet components behavior



# The Tags Facet Displays GUIDs

Why?

# Lab

1. Add the 2 sort and 2 facet components



## Computed Index Fields

# Computed Index Fields

- Sitecore index feature
- Creates new fields that do not exist in templates
- Field values are computed by code

# Out of the Box Coveo Computed Index Fields

Examples are available in the `Coveo.SearchProvider.Custom.config` file

```
<!-- Those are examples of built in Coveo computed fields. Use them to compute an image URL or alt from an image field. -->
<!--
  <field fieldName="imageUrl" sourceField="Image">Coveo.SearchProvider.ComputedFields.ImageUrlComputedField, Coveo.SearchProviderBase</field>
  <field fieldName="imageAlt" sourceField="Image">Coveo.SearchProvider.ComputedFields.ImageAltComputedField, Coveo.SearchProviderBase</field>
-->
<!-- This is another example of built in Coveo computed fields. Use it to get any value from a referenced field.
    Supported attributes, fieldName: The new field to create.
    sourceField: The field containing the referenced item id.
    referencedFieldName: Name of the field on the referenced item.
    returnType: The type of the field. Default is string. -->
<!--
  <field fieldName="theNewFieldToCreate" sourceField="theReferencedFieldId"
        referencedFieldName="referencedField">Coveo.SearchProvider.ComputedFields.ReferencedFieldComputedField, Coveo.SearchProviderBase</field>
-->
```

Simply copy or uncomment them and configure them as you wish. You can as well add them multiple times with different `fieldName` values.

They all inherits from `Coveo.SearchProvider.ComputedFields.ConfigurableComputedField` that implements the `Sitecore.ContentSearch.ComputedFields.IComputedIndexField` interface.

# ConfigurableComputedField.GetAttributeValue(string)

ConfigurableComputedField based computed index fields can use the attributes of the <field> element in the Coveo.SearchProvider.config file.

```
<field fieldName="imageUrl"
      sourceField="Image">Coveo.SearchProvider.ComputedFields.ImageUrlComputedField, Coveo.SearchProviderBase</field>
```

For example, the ImageUrlComputedField reads its sourceField attribute value by calling

```
string sourceField = GetAttributeValue("sourceField");
```

# ReturnType Property

If you create a computed index field, you must implement the getter of the "ReturnType" property with one of the valid values:

- Integer
- Number
- date
- datetime
- string

```
public string ReturnType
{
    get {
        return "datetime";
    }
    set {
    }
}
```

# Date Computed Fields

If you create a computed index field to store a date, you need to keep in mind:

- The "ReturnType" property must be "datetime"

```
public string ReturnType
{
    get {
        return "datetime";
    }
    set {
    }
}
```

- The ComputeFieldValue method must return the date in the CES date format yyyy/MM/dd@HH:mm:ssZ

# Warning About Computed Index Fields



Computed index fields code is executed at indexing time. If they do too much work or their code is not optimized, they can really slow down the whole indexing process.

Be really cautious with your code!

- Avoid running HTTP requests
- Avoid querying a search index
  - A warning is written in the Sitecore logs when computed index fields execute search queries

# Deploying the Coveo.Training.LaunchSitecore Assembly

1. Open the bin folder of the Coveo.Training.LaunchSitecore Visual Studio solution
  - a. \Coveo.Training.LaunchSitecore MONTH YEAR\Coveo.Training.LaunchSitecore\Coveo.Training.LaunchSitecore\bin
2. Copy the Coveo.Training.LaunchSitecore.dll file to your sitecore instance bin folder
  - a. <your sitecore instance folder>\Website\bin

# Lab

1. Deploy the Coveo.Training.LaunchSitecore.dll file in your Sitecore instance

# Configuring Computed Index Fields (1/2)

1. From the file system, browse to your Training configuration folder
  - a. <your sitecore instance folder>\Website\App\_Config\Include\Training
2. Create and open a "ComputedIndexFields.config" file
3. Initialize the file with the configuration on the next page to index in new fields
  - a. The URL of icon and image fields
  - b. The alternative text of icon and image fields
  - c. The tags names

# Configuring Computed Index Fields (2/2)

```
<configuration>
  <sitecore>
    <coveo>
      <defaultIndexConfiguration>
        <documentOptions>
          <fields hint="raw:AddComputedIndexField">
            <field fieldName="iconUrl" sourceField="Icon">Coveo.SearchProvider.ComputedFields.ImageUrlComputedField, Coveo.SearchProviderBase</field>
            <field fieldName="iconAlt" sourceField="Icon">Coveo.SearchProvider.ComputedFields.ImageAltComputedField, Coveo.SearchProviderBase</field>
            <field fieldName="imageUrl" sourceField="Image">Coveo.SearchProvider.ComputedFields.ImageUrlComputedField, Coveo.SearchProviderBase</field>
            <field fieldName="imageAlt" sourceField="Image">Coveo.SearchProvider.ComputedFields.ImageAltComputedField, Coveo.SearchProviderBase</field>
            <field fieldName="tagsnames">Coveo.Training.LaunchSitecore.ComputedFields.TagsNamesComputedField, Coveo.Training.LaunchSitecore</field>
          </fields>
        </documentOptions>
      </defaultIndexConfiguration>
    </coveo>
  </sitecore>
</configuration>
```

4. Save the file
5. Keep this file opened in your editor

# Configuring a Multi-Value String Field

1. In your "ComputedIndexFields.config" file, look for the <documentOptions> node
2. Insert these nodes before its opening tag to configure the new "tagsnames" field as a multi-value field

```
<defaultIndexConfiguration>
  <fieldMap>
    <fieldNames hint="raw:AddFieldByFieldName">
      <fieldType fieldName="tagsnames"
        isMultiValue="true"
        settingType="Coveo.Framework.Configuration.FieldConfiguration, Coveo.Framework" />
    </fieldNames>
  </fieldMap>
  <documentOptions>
```

3. Save the file

# Re-indexing Items with the New Configuration

1. Re-index the /sitecore/Content/Home tree (Developer > Re-Index Tree)
2. Log into the Coveo Cloud Administration Console
3. Navigate to the Content Browser section (Content > Content Browser)
4. In the Source facet on the left, select the "Coveo\_master..." item
5. Sort by indexed date descending
6. Wait for the transaction to be applied by clicking the search button every now and then until the indexed date of the first result matches the time when the re-index was performed

# Validating Field Values (1/3)

1. In the Content Browser
  - a. Sort by relevance
  - b. Search for a Sitecore item that have an icon and an image such "Casey"
  - c. Double click on the search result
2. Enter "ficon" in the Filter text box in the top-right corner of the Fields section
3. Confirm that the "iconalt" and "iconurl" fields are indexed and have the expected values

Mike Casey - Team   Launch Sitecore		
FIELDS	PERMISSIONS	PERMISSION DETAILS
<input type="text" value="ficon"/> <span style="float: right;">x</span>		
Field	Value	
ficonalt95973	user-4	
ficonurl95973	/~/media/Launch Sitecore/Site Images/Icons/Monochrome/t-z/user-4.ashx	

# Validating Field Values (2/3)

4. Enter "fimage" in the Filter text box in the top-right corner of the Fields section
5. Confirm that the "iconalt" and "iconurl" fields are indexed and have the expected values
6. Close the modal box

Field	Value
fimagealt95973	mike casey
fimageurl95973	/~/media/Launch Sitecore/Site Images/Contributors/Mike Casey.ashx

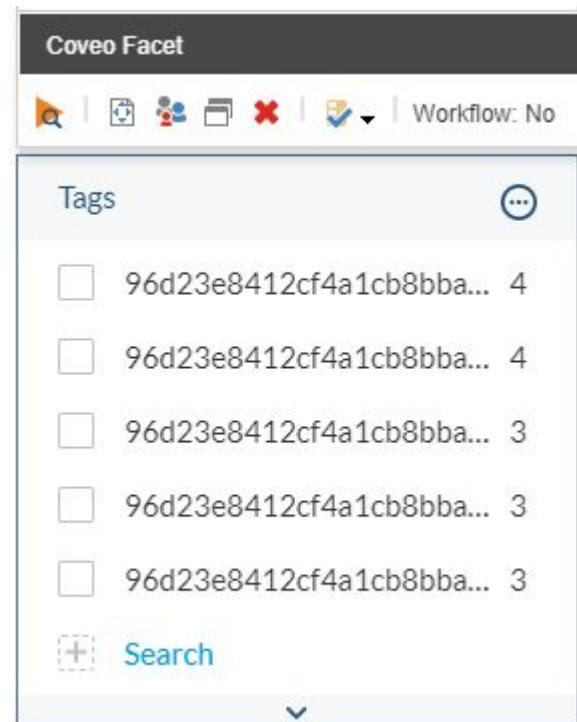
# Validating Field Values (3/3)

7. In the Content Browser
  - a. Search for a Sitecore item that have tags such as "Create a Goal"
  - b. Double click on the search result
8. Enter "tagsnames" in the Filter text box in the top-right corner of the Fields section
9. Confirm that the "tagsnames" field is indexed and have the expected values

Field	Value
ftagsnames95973	DMS,How-to,Beginner

# Using the TagsNames Field in the Tags Facet (1/2)

1. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
2. Access the Experience Editor (Publish > Experience Editor)
3. Select the Tags facet
4. Click the Coveo experience button in the floating toolbar



# Using the TagsNames Field in the Tags Facet (2/2)

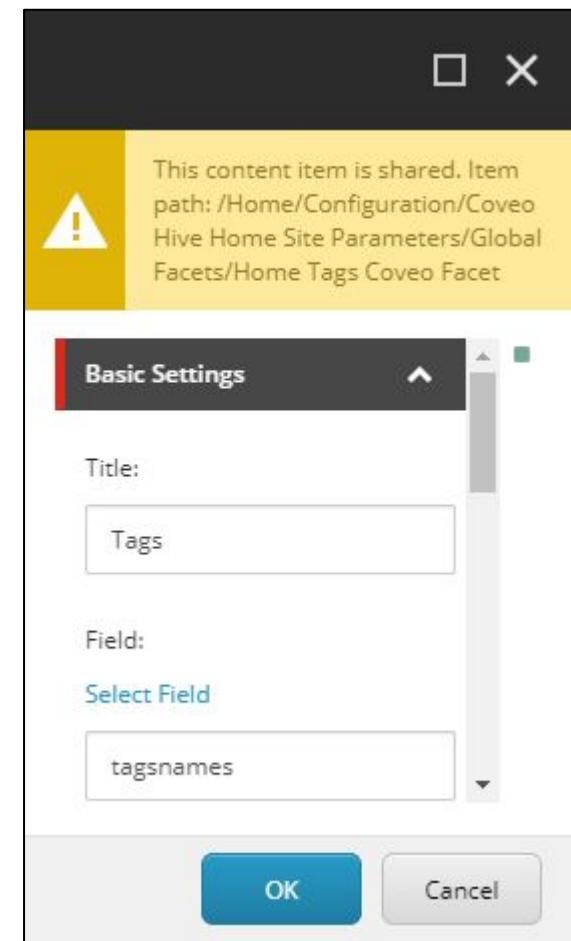
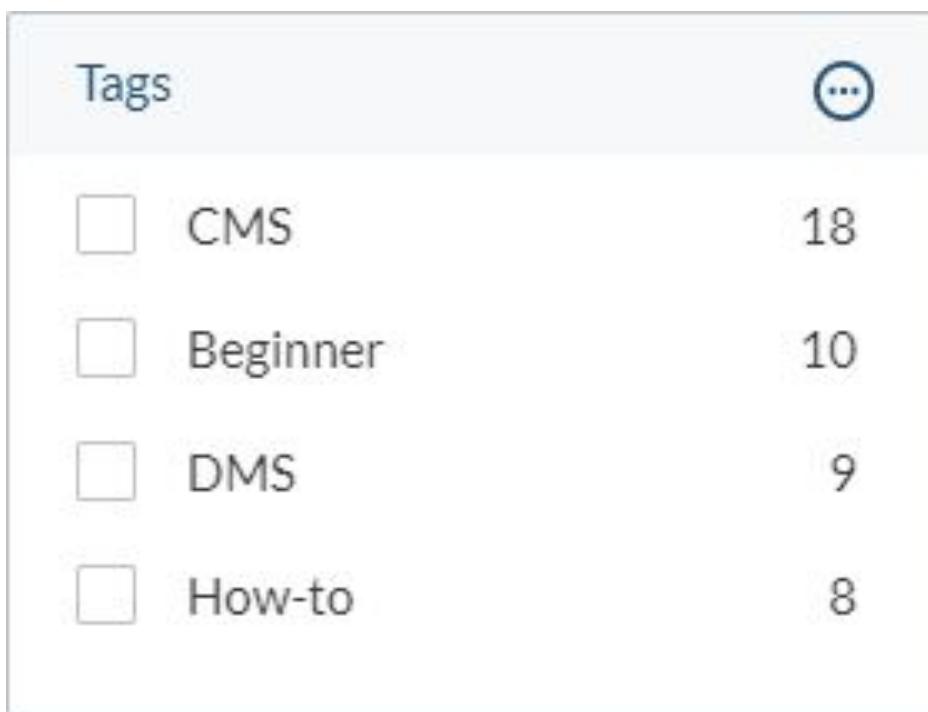
## 5. In the Control Properties dialog

### a. Basic Settings

- i. Field: Enter "tagsnames"

### b. Choose OK

## 6. Validate the Tags facet component behavior



# Adjusting a Facet for a Multi-Value Field

A facet will compute the values count dynamically. If a facet with the OR behavior has a selected value, the count of the other facet values will show the additional results that would be added if a user selects the value.

For a multi-value field, the count will be wrong, since a result can have multiple values for that field.

Tags	X	...
<input checked="" type="checkbox"/> CMS	18	
<input type="checkbox"/> Beginner	+10	
<input type="checkbox"/> DMS	+9	
<input type="checkbox"/> How-to	+8	

# Adjusting a Facet for a Multi-Value Field (1/2)

1. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
2. Access the Experience Editor (Publish > Experience Editor)
3. Select the Tags facet
4. Click the Coveo experience button in the floating toolbar

The screenshot shows the Sitecore Experience Editor interface. A floating toolbar titled 'Coveo Facet' is visible at the top, featuring icons for search, refresh, user, and workflow, along with a status message 'Workflow: No'. Below the toolbar is a list titled 'Tags' with the following items:

Tags	...
<input type="checkbox"/> CMS	18
<input type="checkbox"/> Beginner	10
<input type="checkbox"/> DMS	9
<input type="checkbox"/> How-to	8

# Adjusting a Facet for a Multi-Value Field (2/2)

## 5. In the Control Properties dialog

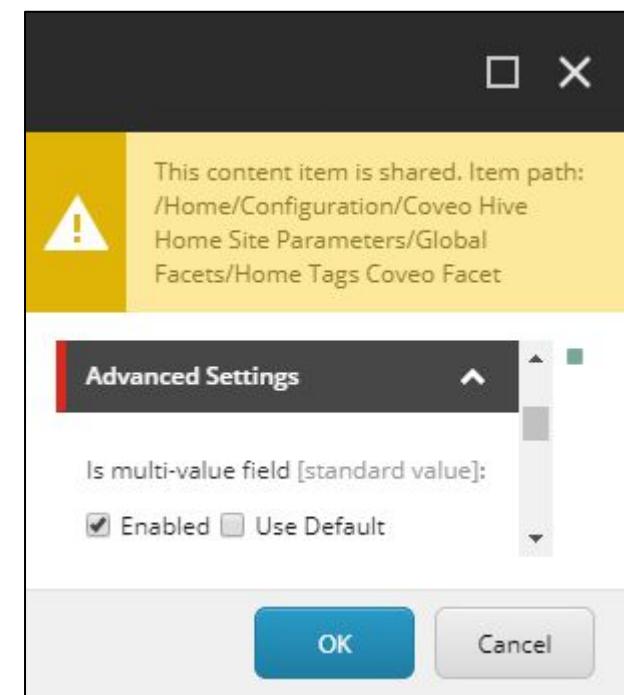
### a. Advanced Settings

- i. Is multi-value field: Uncheck "Use Default", then check "Enabled"

### b. Choose OK

## 6. Validate the Tags facet component behavior

Tag	Count
CMS	18
Beginner	+3
DMS	+3
How-to	+3



# Lab

1. Configure computed index fields
2. Configure multi-value field
3. Re-index the /sitecore/Content/Home subtree
4. Validate the indexed fields values
5. Change the Tags facet field
6. Change the Tags facet item count behavior



# Result Templates

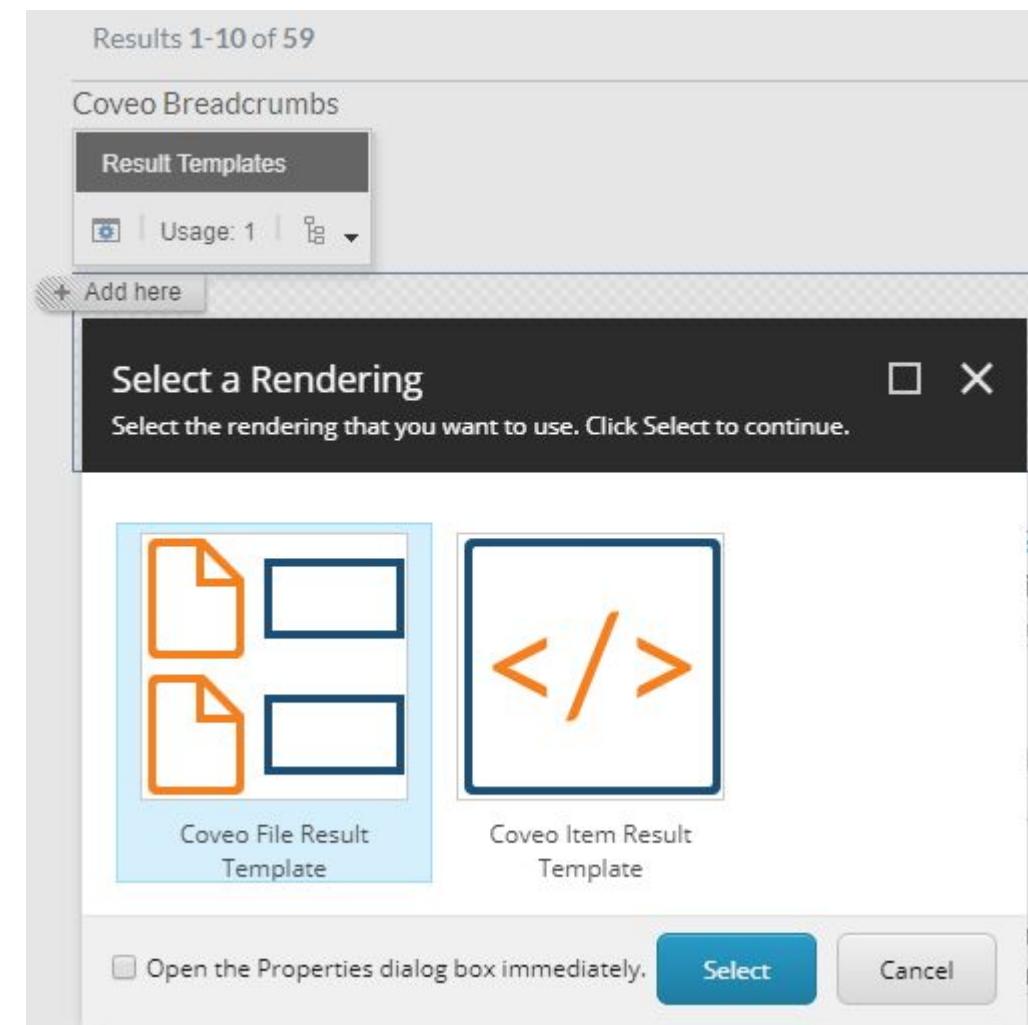
# Result Templates

Result templates can be stored in 2 locations:

- File system
  - <your sitecore instance folder>\Website\Coveo\Hive\templates
  - One cshtml file per template
- Sitecore content tree
  - In a Result Templates folder
  - Code stored in a multi-line field
  - Can be versioned, have languages
  - Can be published with content without code deploy

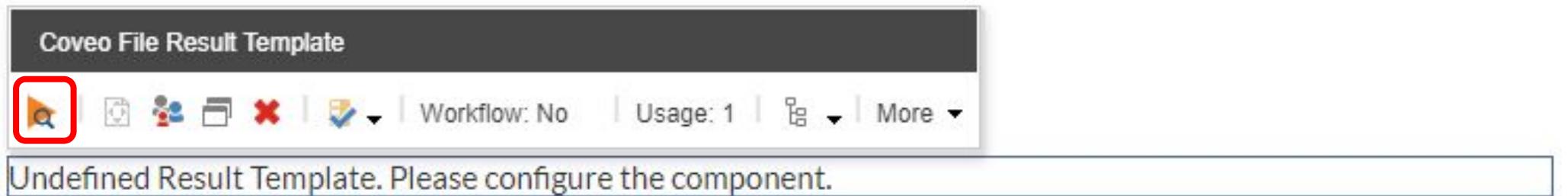
# Using the Default Result Template (1/3)

1. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
2. Access the Experience Editor (Publish > Experience Editor)
3. Select the Result Templates placeholder
4. Click the "Add here" button
5. In the Select a Rendering dialog
  - a. Select Coveo File Result Template
  - b. Choose Select

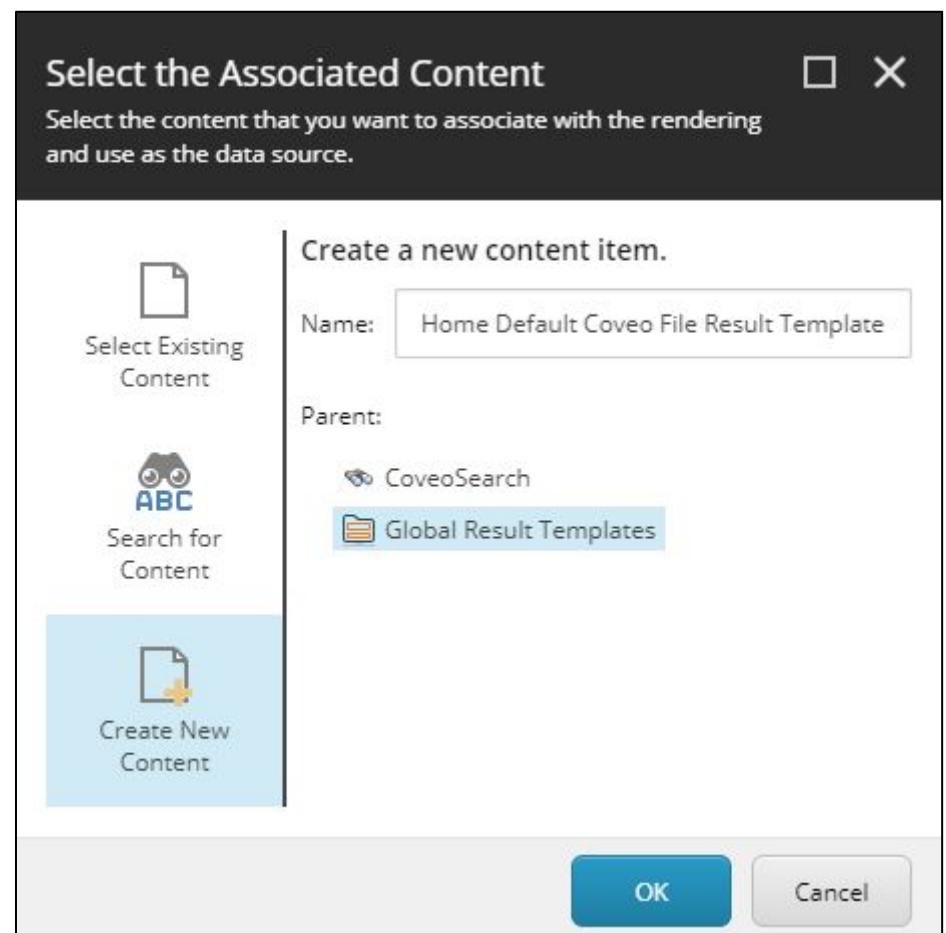


# Using the Default Result Template (2/3)

6. In the Select the Associated Content dialog
  - a. In the left column, select Create New Content
  - b. Name: Home Default Coveo File Result Template
  - c. Parent: Global Result Templates
  - d. Choose OK



7. Click the Coveo experience button in the floating toolbar



# Using the Default Result Template (3/3)

## 8. In the Control Properties dialog

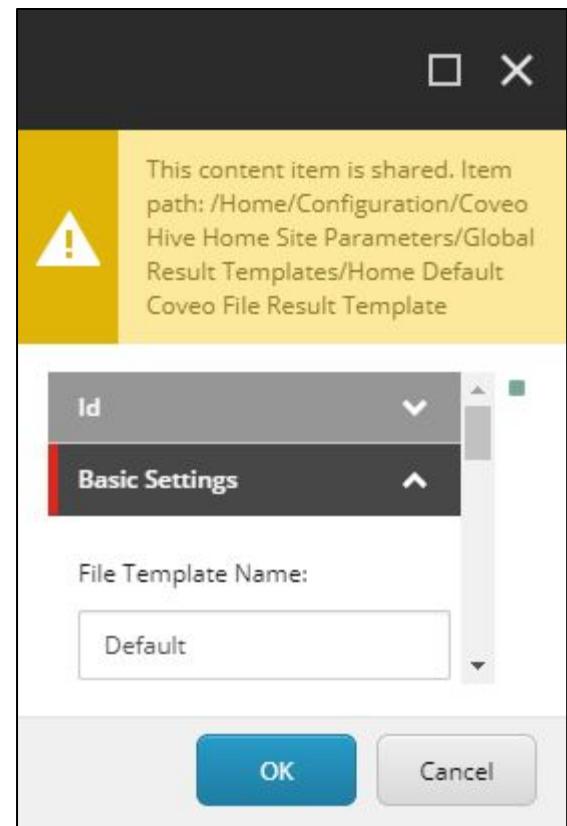
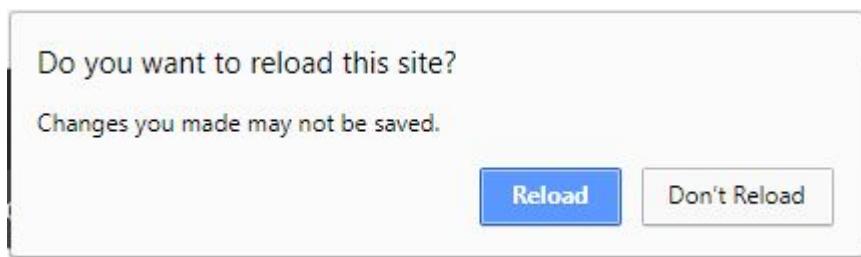
### a. Basic Settings

i. File Template Name: Enter "Default"

### b. Choose OK

## 9. Your browser might ask if you want to reload the page

### a. Choose Don't Reload



## 10. Save the item

## 11. Validate the result template is used



Type	...
<input type="checkbox"/> Article	21
<input type="checkbox"/> Term	16
<input type="checkbox"/> Team Member	7
<input type="checkbox"/> Article Group	4
<input type="checkbox"/> Audience Segment	4
<a href="#">Search</a>	▼
Tags	...
<input type="checkbox"/> CMS	18
<input type="checkbox"/> Beginner	10
<input type="checkbox"/> DMS	9
<input type="checkbox"/> How-to	8

Results 1-10 of 59

RELEVANCY DATE ▾

- [Basic Presentation Explanation - About Launch Sitecore | Launch Sitecore](#) 4/16/2016  
Sitecore  
You will see a dialog similar to the screenshot below: ... Home ... Audience Landing Pages ... About Launch Sitecore ... Team ... Glossary ... Login ... Copyright © 1999-2015 Sitecore
- [The Sitecore UIs - About Launch Sitecore | Launch Sitecore](#) 4/16/2016  
Team ... Edit enables all in-line editing, adding/modifying of components, layout, etc. ... Adding page Components ... Inserting new content ... Publishing ... Logging off ... Experience Tab
- [Create a Goal - About Launch Sitecore | Launch Sitecore](#) 4/16/2016  
1) ... In a real-world scenario, this could be a "Thank You" page where a visitor is sent to ... 2)  
Sitecore (and from our Shared Source offering, developers from the community) ... Login
- [Create a custom validator - About Launch Sitecore | Launch Sitecore](#) 4/16/2016  
Look for these in the content tree at: /sitecore/system/Settings/Validation Rules/ ... protected  
override ValidatorResult Evaluate() ... if (!string.IsNullOrEmpty(base.ControlValidationValue))

# Lab

1. Add a result template
2. Configure the result template to use the file named default

# Displaying Field Values in Result Templates

Field values can be displayed using multiple techniques:

- In a CoveoFieldTable row

```
<tr data-field='<%- coveoFieldName("@@SitecoreFieldNameToDisplay") %>'  
    data-caption='<%- translateLabel("TableRowCaption") %>' />
```

- With the CoveoFieldValue component

```
<span class="CoveoFieldValue" data-field='<%- coveoFieldName("@@SitecoreFieldNameToDisplay") %>'></span>
```

- With Underscore JS templating

```
<%= raw[coveoFieldName("SitecoreFieldNameToDisplayHtmlUnescaped")] %>  
<%- raw[coveoFieldName("SitecoreFieldNameToDisplayHtmlEscaped")] %>
```

```
<%= coveoFieldValue("SitecoreFieldNameToDisplayHtmlUnescaped") %>  
<%- coveoFieldValue("SitecoreFieldNameToDisplayHtmlEscaped") %>
```

# Displaying the Tags Names in the Result Template (1/3)

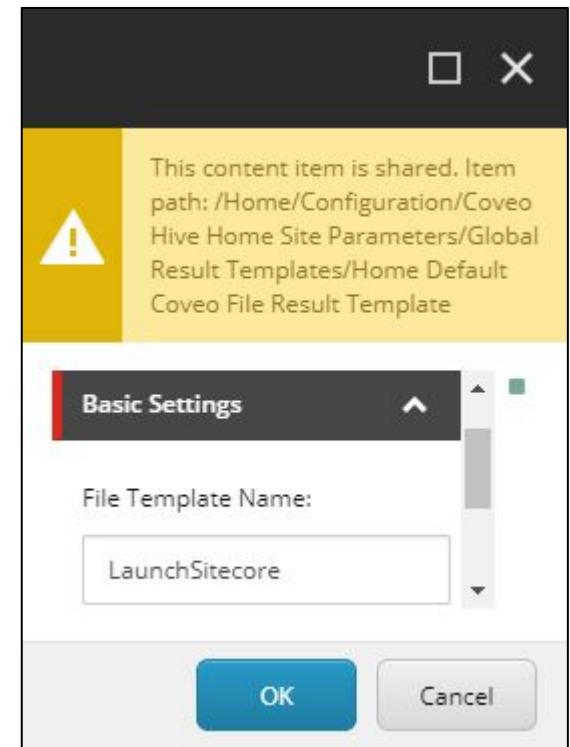
1. From the file system, browse to the result templates folder
  - a. <your sitecore instance folder>\Website\Coveo\Hive\templates
2. Duplicate the "default.cshtml" file to "LaunchSitecore.cshtml"
3. Open your "LaunchSitecore.cshtml" file and look for the  
`<table class='CoveoFieldTable'>` element
4. Insert a table row at its end to display the item tags

```
<table class="CoveoFieldTable">
  <tbody>
    ...
    <tr data-field='<%= coveoFieldName("@@tagsnames") %>' data-caption='<%= translateLabel("Tags") %>'
      data-split-values='true' />
  </tbody>
</table>
```

5. Save the file and keep it opened

# Displaying the Tags Names in the Result Template (2/3)

6. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
7. Access the Experience Editor (Publish > Experience Editor)
8. Select the Default Coveo File Result Template component
9. Click the Coveo experience button in the floating toolbar



10. In the Control Properties dialog
  - a. Basic Settings
    - i. File Template Name: Enter "LaunchSitecore"
  - b. Choose OK

# Displaying the Tags Names in the Result Template (3/3)

## 11. Validate the tags names are displayed in the result template

File Result Template: /Coveo/Hive/templates/LaunchSitecore.cshtml

HTML File

The Sitecore UIs - About Launch Sitecore | Launch Sitecore 4/16/2016

</>

Team ... Edit enables all in-line editing, adding/modifying of components, layout, etc. ... Adding page Components ... Inserting new content ... Publishing ... Logging off ... Experience Tab

Details

Template	Article
Created By	sitecorechc
Created	12/20/2011
Updated By	sitecoreadmin
Language	english
URI	<a href="http://devtraining41/about-launch-sitecore/getting-started/the-sitecore-uis">http://devtraining41/about-launch-sitecore/getting-started/the-sitecore-uis</a>
Tags	CMS, DMS, Beginner

# Lab

1. Display the tags in each result fields table

# Displaying the Item Icon in the JS UI Result Template

Launch Sitecore items all have a unique icon.  
We will use these icons in the search result template.

## Launch Sitecore

 [What is Content?](#)  
A Content Management System by its name manages content, but it also must handle the presentation of that content. While these are related, it is important to look at them individually.

 [Basic Presentation Explanation](#)  
This article explains the basic presentation layer used in this site. Core concepts of Layouts, Sublayouts and Placeholders will be described in the context of the Launch Sitecore site's "Basic" layout theme.

 [Workflow](#)  
An introduction to the Sitecore workflow engine. This article walks through the Sample Workflow that ships with Sitecore.

## Search Page

 [Basic Presentation Explanation - About Launch Sitecore | Launch Sitecore](#) 4/16/2016  
You will see a dialog similar to the screenshot below: ... Home ... Audience Landing Pages ... About Launch Sitecore ... Team ... Glossary ... Login ... Copyright © 1999-2015 Sitecore

 [The Sitecore UIs - About Launch Sitecore | Launch Sitecore](#) 4/16/2016  
Team ... Edit enables all in-line editing, adding/modifying of components, layout, etc. ... Adding page Components ... Inserting new content ... Publishing ... Logging off ... Experience Tab

 [Create a Goal - About Launch Sitecore | Launch Sitecore](#) 4/16/2016  
1) ... In a real-world scenario, this could be a "Thank You" page where a visitor is sent to ... 2) Sitecore (and from our Shared Source offering, developers from the community) ... Login

# Displaying the Item Icon in the JS UI Result Template

1. From the file system, browse to the result templates folder
  - a. <your sitecore instance folder>\Website\Coveo\Hive\templates
2. Open your "LaunchSitecore.cshtml" file and look for the CoveoIcon  
`<span class="CoveoIcon"></span>`

## 3. Replace it by this Underscore JS code

```
<% var iconUrl = coveoFieldValue("iconurl") %>
<% if (iconUrl) { %>
    <% var iconAlt = coveoFieldValue("iconalt") || "" %>
    <span class="coveo-icon"><%= image(iconUrl, { alt: iconAlt, width: '60px', height: '60px' }) %></span>
<% } else { %>
    <span class="CoveoIcon"></span>
<% } %>
```



The Sitecore UIs - About Launch Sitecore | Launch Sitecore

Team ... Edit enables all in-line editing, adding/modifying of components, layout, etc. ... Adding page Components ... Inserting new content ... Publishing ... Logging off ... Experience Tab

4. Save the file
5. Validate that your results are displaying the icon

4/16/2016

# Lab

1. Change the results icons for Launch Sitecore ones

# Templating Team Members Results (1/8)

1. From the file system, browse to the result templates folder
  - a. <your sitecore instance folder>\Website\Coveo\Hive\templates
2. Create and open a "TeamMember.cshtml" file
3. Initialize the file with this code and save the file

```
<div class="coveo-result-frame coveoforsitecore-template">
  <div class="coveo-result-row">
    <div class="coveo-result-cell coveo-quick-view-container">
      <% var imageUrl = coveoFieldValue("imageurl") || coveoFieldValue("iconurl") %>
      <% if (imageUrl) { %>
        <% var imageAlt = coveoFieldValue("imagealt") || coveoFieldValue("iconalt") || "" %>
        <span class="coveo-icon"><%= image(imageUrl, { alt: imageAlt, width: '60px', height: '60px' }) %></span>
      <% } else { %>
        <span class="CoveoIcon"></span>
      <% } %>
    </div>
```

# Templating Team Members Results (2/8)

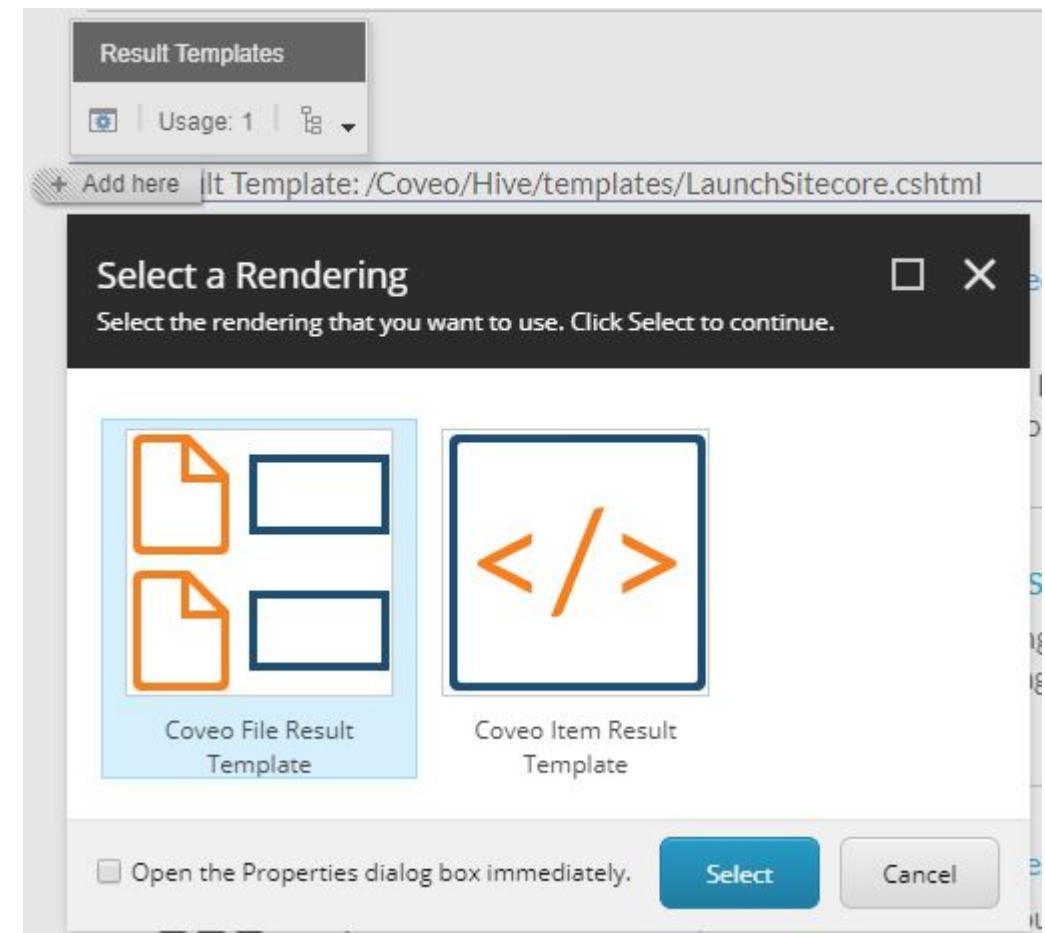
```
<div class="coveo-result-cell coveoforsitecore-information-section">
    <div class="coveo-result-row">
        <div class="coveo-result-cell">
            <div class='coveo-title'>
                <% if (coveoFieldValue("haslayout") === "1" || raw.syssource !== currentSourceName()) { %>
                    <a class='CoveoResultLink'><%= title ? highlight(title, titleHighlights) : clickUri%></a>
                <% } else { %>
                    <span class='CoveoResultTitle'><%= title ? highlight(title, titleHighlights) : ''%></span>
                <% } %>
            </div>
        </div>
    </div>
</div>

<div class="coveo-result-row">
    <div class="coveo-result-cell" style="font-size:14px;">
        <span class="CoveoFieldValue" data-field="<%= coveoFieldName("@@job title") %>"></span>
    </div>
</div>

<div class="coveo-result-row" style="margin-top:0.5em;">
    <div class="coveo-result-cell">
        <span><%= highlightStreamText(coveoFieldValue("abstract").replace("<p>","").replace("</p>","","").replace("<P>","","").replace("</P>","","")) %></span>
    </div>
</div>
</div>
</div>
```

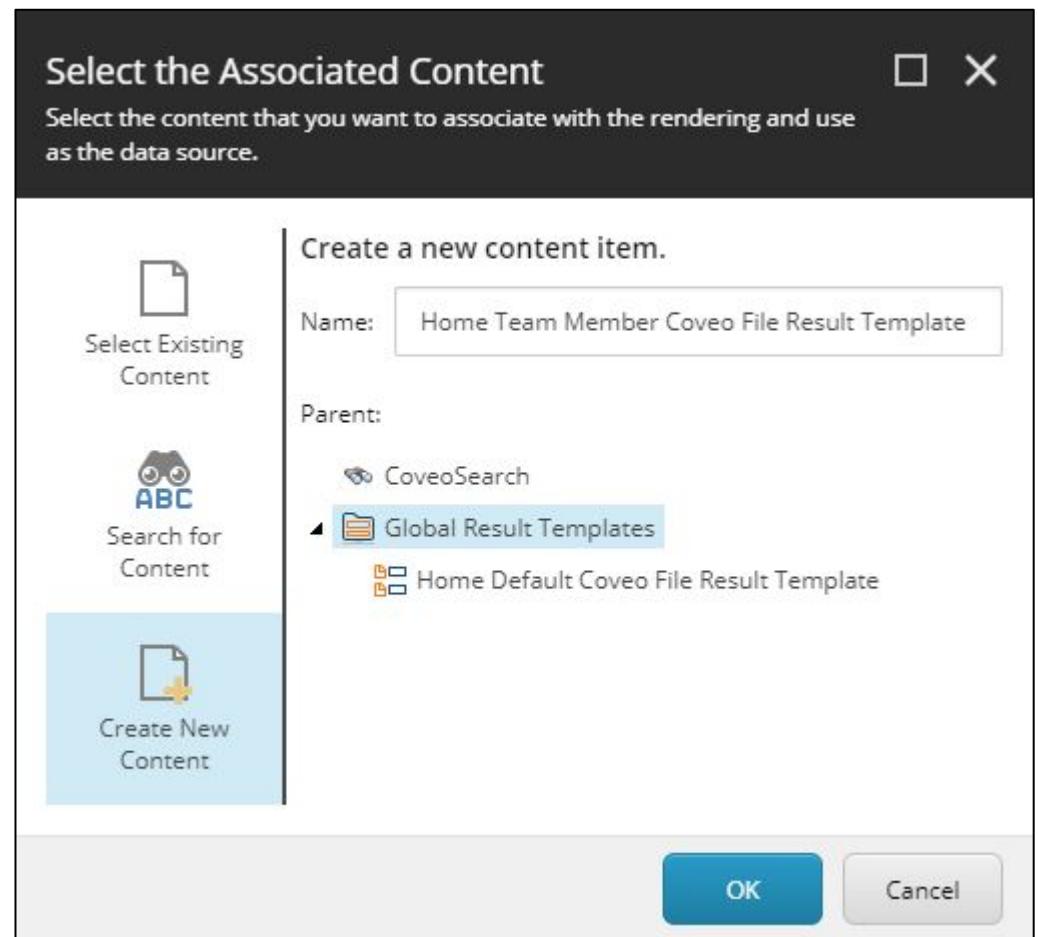
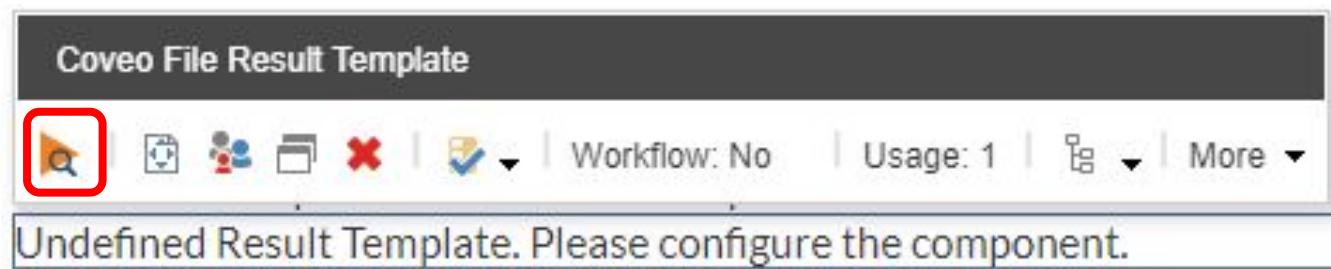
# Templating Team Members Results (3/8)

4. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
5. Access the Experience Editor (Publish > Experience Editor)
6. Select the Result Templates placeholder
7. Click the visible "Add here" button
8. In the Select a Rendering dialog
  - a. Select Coveo File Result Template
  - b. Choose Select



# Templating Team Members Results (4/8)

9. In the Select the Associated Content dialog
- In the left column, select Create New Content
  - Name: Home Team Member Coveo File Result Template
  - Parent: Global Result Templates
  - Choose OK



10. Click the Coveo experience button in the floating toolbar

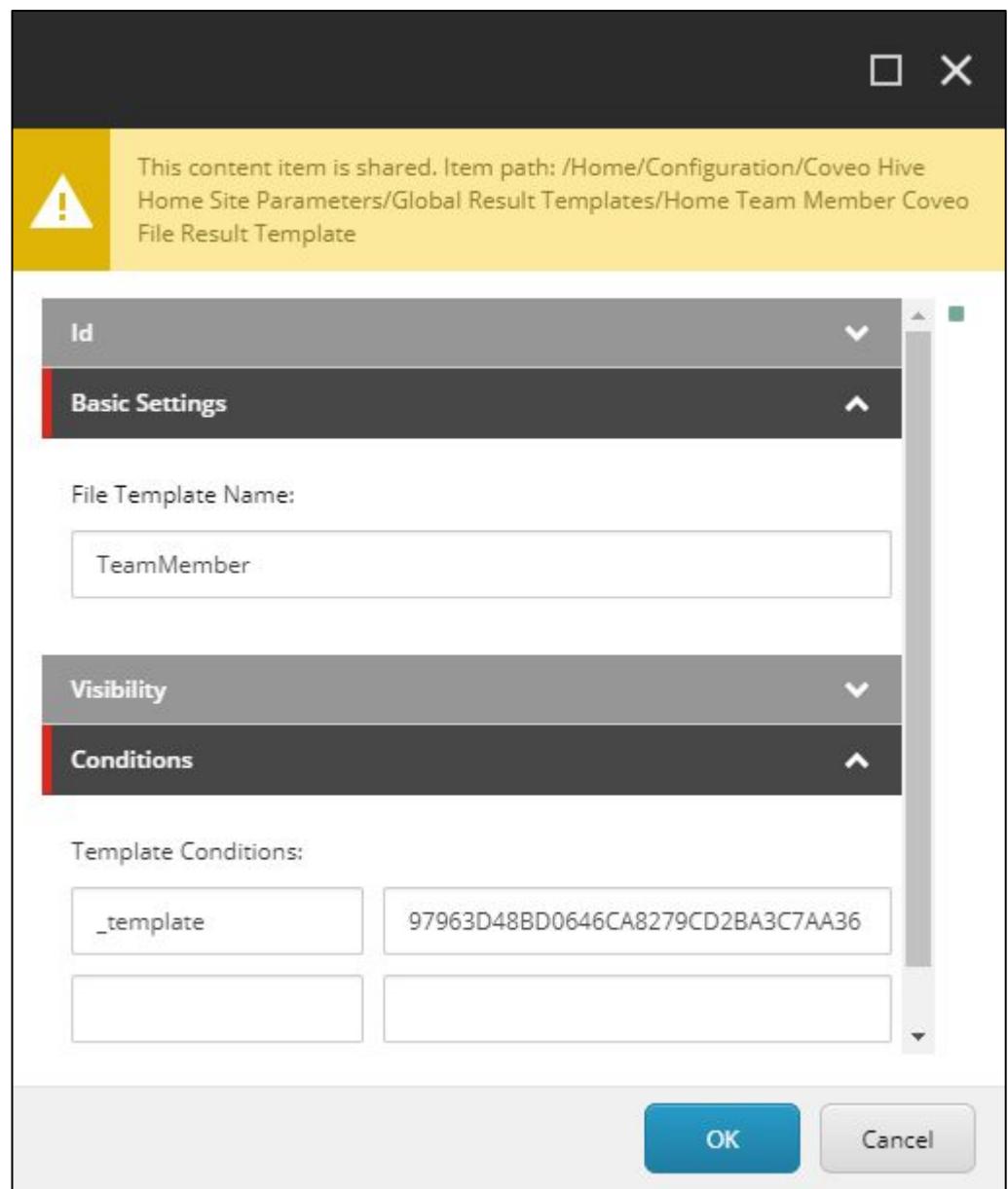
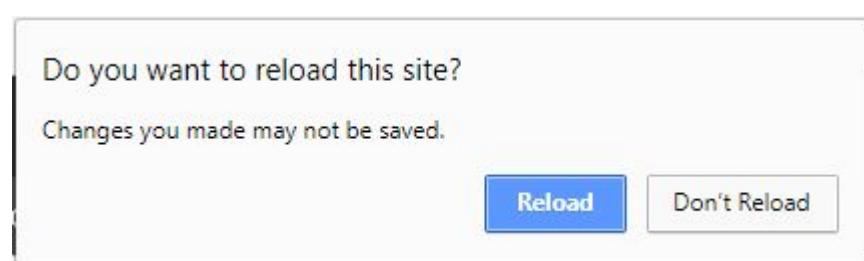
# Templating Team Members Results (5/8)

## 11. In the Control Properties dialog

- a. Basic Settings
  - i. File Template Name: Enter "TeamMember"
- b. Conditions
  - i. \_template: 97963D48BD0646CA8279CD2BA3C7AA36
- c. Choose OK

## 12. Your browser might ask if you want to reload the page

- a. Choose Don't Reload



# Templating Team Members Results (6/8)

13. Save the item
14. In the Type facet, select Team Member
15. Validate that your results are displaying the team members differently

Team Member 7  
 Article Group +4  
 Audience Segment +4  
+ Search

Coveo Did You Mean  
Coveo Error Report  
Coveo Results List

File Result Template: /Coveo/Hive/templates/LaunchSitecore.cshtml  
File Result Template: /Coveo/Hive/templates/TeamMember.cshtml

Tags

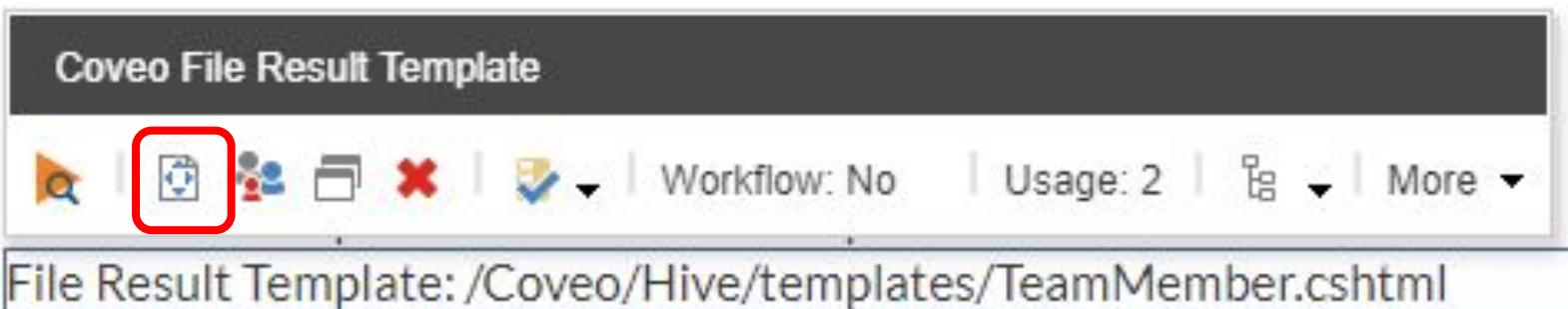
 Kevin DiTroia - Team | Launch Sitecore 7/15/2015  
Kevin began his career at Sitecore on the technical services team, providing ... As a Solution Architect, he now focuses on implementation strategy and development ... Articles ... Pattern  
[Details](#)

# The Team Members Results Are not Using their Template

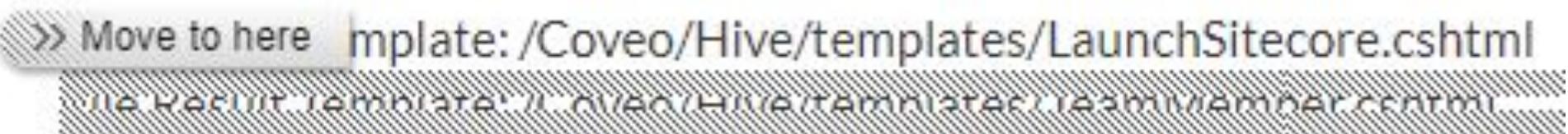
Why?

# Templating Team Members Results (7/8)

16. Select the Team Member Coveo File Result Template component



17. Click the Coveo experience button in the floating toolbar
18. Click the "Move to here" button



19. Save the item

# Templating Team Members Results (8/8)

## 20. Validate that your results are displaying the team members differently

<input checked="" type="checkbox"/> Team Member	7
<input type="checkbox"/> Article Group	+4
<input type="checkbox"/> Audience Segment	+4
<b>Search</b>	

▼

Tags 

Coveo Did You Mean  
Coveo Error Report  
Coveo Results List  
  
File Result Template: /Coveo/Hive/templates/TeamMember.cshtml  
File Result Template: /Coveo/Hive/templates/LaunchSitecore.cshtml



### Kevin DiTroia - Team | Launch Sitecore

Senior Regional Sales Engineer, West

Kevin is a Sitecore architect and developer who is constantly looking for exciting ways to utilize Sitecore to meet unique business needs.

# Lab

1. Add a different result template for team members

# Publishing Changes

1. Smart publish your site
2. Log into the Coveo Cloud Administration Console
3. Navigate to the Content Browser section (Content > Content Browser)
4. In the Source facet on the left, select the "Coveo\_web\_in..." item
5. Sort by indexed date descending
6. Wait for the transaction to be applied by clicking the search button every now and then until the indexed date of the first result matches the time when the publish was performed

# Validating the Published Search Page

1. In another browser or an incognito window, open your published search page
  - a. <http://YourInstanceName/CoveoSearch>
2. Validate that:
  - a. Facets are displaying and working correctly
  - b. Search results are displaying the custom icons
  - c. Search results are displaying the tags names
  - d. Team member results are displaying with the right template

# The Search Page is Not Displaying as Expected

Why?

# The Search Page is Not Displaying as Expected

The publish site operation only updated and re-indexed Sitecore items that have changed in the master database.

It didn't re-index the articles, team members and other items to add the new computed fields values.

The web index needs to be rebuilt.

1. Rebuild your Coveo\_web\_index from the Indexing Manager or Developer toolbar (Developer > Rebuild index)

# Validating the Published Search Page (2<sup>nd</sup> time)

1. Refresh the published search page in your other browser or incognito window
2. Validate that:
  - a. Facets are displaying and working correctly
  - b. Search results are displaying the custom icons
  - c. Search results are displaying the tags names
  - d. Team member results are displaying with the right template

# Lab

1. Publish the site
2. Rebuild your Coveo\_web\_index
3. Validate the published website is working correctly



## Free-Text Search

# Free-Text Search

In a Coveo index, documents can be free-text searched only by their:

- Title
- URI
- Body (converted and set as the QuickView content)
- Free-text searchable fields

To obtain a good relevancy, it's very important to ensure the body of the searchable documents contains all the text the visitor can see on the page except the common text visible on all the pages such as the header, footer, navigation, sidebars, ads, etc.

# Body Indexing Options

By default, Coveo for Sitecore sets the body of indexed media items to the file they reference.

For other Sitecore items, it is configurable at installation time:

- Do not set the body
- [Set the body as the Sitecore rendered HTML for the item](#)

This installation step configures a processor in the  
`<coveoPostItemProcessingPipeline>`

# Alternate Body Indexing Options

Instead of the Sitecore rendered HTML, you can choose to set the body to something else by either:

- Coding your own processor to set a custom body
- Indexing items basic HTML content

# BasicHtmlContentInBodyProcessor

- All the fields and their names are rendered in the body of the indexed document by default
- The processor have options
  - IncludeFieldNames: Whether to render the field names
  - IncludeTextFieldsOnly: Whether to only include text (single and multi line) fields
  - FieldsToInclude: Comma separated list of field names to include
  - TemplatesToInclude: Comma separated list of template names for which to run the processor

# HtmlContentInBodyWithRequestsProcessor

- The HTTP request:
  - Is issued only for items that have a layout
  - Is always targeting the published website (web database)
  - Uses the "Coveo Sitecore Search Provider" user agent
- Site header, footer, navigation and sidebars are by default included in body
  - Because they are in the web database Default device presentation details
- To index the item's presentation without those elements, you must either:
  - Create a Quick View device and layout. Then, define this device layout on each of your templates with only the minimal components
  - Modify the header, footer, navigation and sidebars code to render nothing when the request comes from the "Coveo Sitecore Search Provider" user agent
  - Configure the processor to remove undesired parts of the HTML before indexing

# Removing Undesired Text From Body - Processor

1. From the file system, browse to your Training configuration folder
  - a. <your sitecore instance folder>\Website\App\_Config\Include\Training
2. Create and open a "CleanIndexedHtml.config" file
3. Initialize the file with this configuration to clean the HTML

```
<configuration>
  <sitecore>
    <pipelines>
      <coveoPostItemProcessingPipeline>
        <processor type="Coveo.SearchProvider.Processors.HtmlContentInBodyWithRequestsProcessor, Coveo.SearchProviderBase">
          <StartCommentText>BEGIN NOINDEX</StartCommentText>
          <EndCommentText>END NOINDEX</EndCommentText>
        </processor>
      </coveoPostItemProcessingPipeline>
    </pipelines>
  </sitecore>
</configuration>
```

4. Save the file

# Removing Undesired Text From Body - Main Layout (1/2)

1. From the file system, open the Launch Sitecore Main layout file
  - a. <your sitecore instance folder>\Website\Views\Shared\Main.cshtml
2. Add NOINDEX comments around the header and footer markup

```
<!-- BEGIN NOINDEX -->
<!-- HEADER AREA -->
@Html.Sitecore().Controller("Navigation", "Header")
<!-- END NOINDEX -->

<!-- BEGIN NOINDEX -->
<!-- FOOTER STARTS HERE -->
<footer id="footer">
    ...
</footer>
<!-- END FOOTER -->
<!-- END NOINDEX -->
```

# Removing Undesired Text From Body - Main Layout (2/2)

## 3. Add NOINDEX comments around the personalization markup

```
<!-- BEGIN NOINDEX -->
<div id="mtp-toggle">
  ...
</div>
<div id="mtp-wrapper">
  ...
</div>
<!-- END NOINDEX -->
```

## 4. Add NOINDEX comments around the title element

```
<!-- BEGIN NOINDEX -->
<title>@Html.Sitecore().Controller("Meta", "PageTitle")</title>
<!-- END NOINDEX -->
```

## 5. Save the file

# Removing Undesired Text From Body - Rebuild

1. Rebuild your Coveo\_web\_index from the Indexing Manager

# Removing Undesired Text From Body - Validation

1. Open your published website
  - a. <http://YourInstanceName/CoveoSearch>
2. Validate that searching for terms like "copyright", "home", or "team" doesn't return all the indexed documents
3. Validate that the Quick View of search results only contains the desired text
4. Validate that the title of the search results does not contain "- About Launch Sitecore | Launch Sitecore" anymore

# Lab

1. Configure the `HtmlContentInBodyWithRequestsProcessor` to clean the HTML of the pages before indexing them
2. Add `NOINDEX` comments in the Launch Sitecore main layout
3. Rebuild the web index
4. Validate the quick view is clean and generic queries return relevant results

# Free-Text Searchable Fields

Sometimes, the Sitecore rendered HTML of an item doesn't contain everything you want to be searchable.

You could have alternate product numbers, sub-part numbers or synonyms defined but not displayed for an item.

Additionally to indexing the body, you can also configure fields to be included in free-text search.

# Configuring a Free-Text Searchable Field (1/2)

1. From the file system, browse to your Training configuration folder
  - a. <your sitecore instance folder>\Website\App\_Config\Include\Training
2. Create and open a "FreeTextSearchbleField.config" file
3. Initialize the file with this configuration to configure the "abstract" field as a free-text field

```
<configuration>
  <sitecore>
    <coveo>
      <defaultIndexConfiguration>
        <fieldMap>
          <fieldNames hint="raw:AddFieldByFieldName">
            <fieldType fieldName="abstract" includeForFreeTextSearch="true" settingType="Coveo.Framework.Configuration.FieldConfiguration,
Coveo.Framework" />
          </fieldNames>
        </fieldMap>
      </defaultIndexConfiguration>
    </coveo>
  </sitecore>
</configuration>
```

# Configuring a Free-Text Searchable Field (2/2)

4. Save the file
5. Re-index the /sitecore/Content/Home tree (Developer > Re-Index Tree)
6. Log into the Coveo Cloud Administration Console
7. Navigate to the Content Browser section (Content > Content Browser)
8. In the Source facet on the left, select the "Coveo\_master..." item
9. Sort by indexed date descending
10. Wait for the transaction to be applied by clicking the search button every now and then until the indexed date of the first result matches the time when the re-index was performed
11. Validate that searching for an abstract text returns the desired result

# Lab

1. Configure a free-text searchable field
2. Validate its text is searchable



## External Content

# External Content

- External content must be indexed in separate sources
- All the filtering/boosting rules and the facet fields of your search pages also apply to external documents
- External documents must be available either to:
  - Sitecore users and roles
  - Anonymous email users
  - Public (Anonymous access)

# Using an External Source/Collection in Sitecore

3 alternatives to define the external sources (or CES collections) to use:

- **In the HTML markup**

- Used when you don't want the content editors to modify

```
<div class="CoveoForSitecoreExternalContent"  
     data-sc-sources="Habitat,AnotherExtenalSource"></div>
```

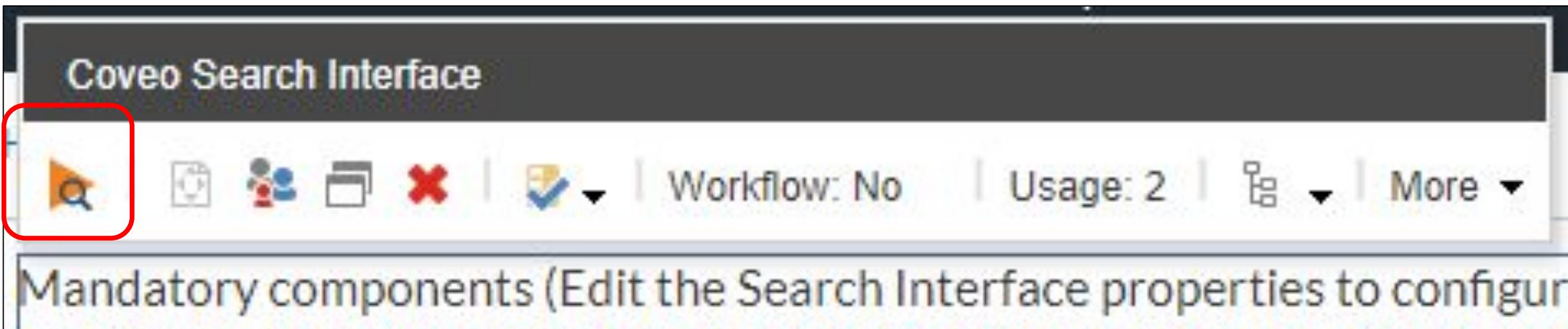
- In the Properties Editor for the Coveo Search Interface component

- By inserting an External Content component in the search page

- Both are used when you want the content editors to control
  - Allows to easily select collections to use from a list when indexing on-premises

# Using an External Source from the Properties Editor (1/2)

1. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
2. Access the Experience Editor (Publish > Experience Editor)
3. Select the Coveo Search Interface component
4. Click the Coveo experience button in the floating toolbar





# Using an External Source from the Properties Editor (2/2)

## 5. In the Control Properties dialog

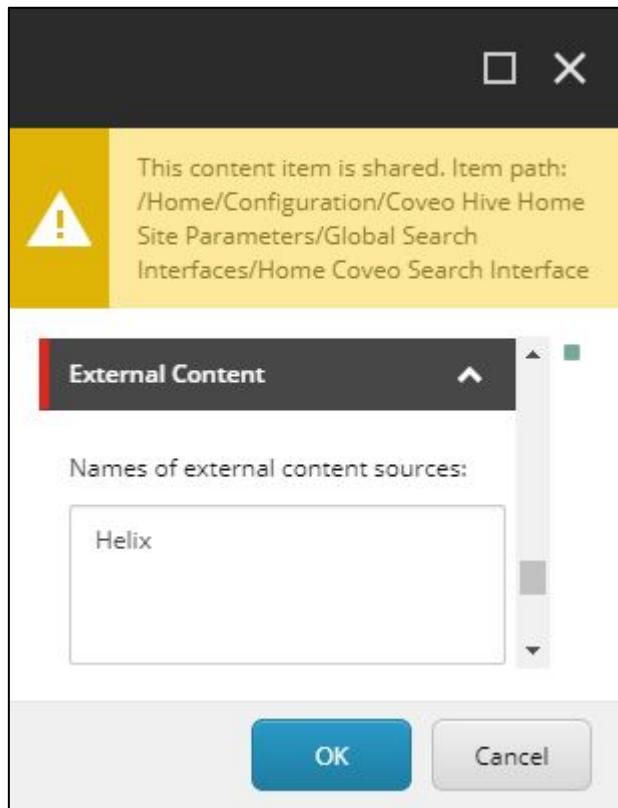
### a. External Content

i. Names of External Content Sources: Enter "Helix"

### b. Choose OK

## 6. Search for "helix"

a. No results should be displayed because of filters



# Anatomy of a Coveo for Sitecore Query With External Content

## Advanced Query:

```
(  
  Boosting Rules from the Properties Editor  
)  
(  
  Advanced Expression QREs Added from the JavaScript Code  
)  
(  
  (  
    Filtering Rules from the Properties Editor  
    Item Templates Exclusion from the Properties Editor Options  
)  
)  
(  
  Advanced Expression Filters Added from the JavaScript Code  
)  
(  
  Facet Selection Expressions  
)  
(  
  Automatic and External Source Filter  
)
```

## Constant Query:

```
(  
  Various Filters from the Properties Editor Options  
)  
OR  
(  
  External Sources Filter  
)  
OR  
(  
  External Collections Filter  
)
```

# Filter Expressions VS External Content

- Filtering rules from the properties editor apply also to all external content
- Filtering rules from the JavaScript code apply also to all external content

To make a Sitecore field filtering rule to match external content, there are 2 alternatives:

- Arrange for the Sitecore field to be filled with a matching value for external documents
  - a. Fields that have a unique value for the entire source can be set in the source mapping
  - b. Document-dependant fields can be set by a web scraping configuration
- Add an (OR @syssource=="The external source name") expression to a filtering rule to cancel its application to the external source

# External Content Fields

When a Sitecore field should also exist on external documents, either:

- Define all the translated variations of the field in the external source mapping
  - a. fshowz32xinz32xsearchz32xresultsXXXXX (Master source)
  - b. fshowz32xinz32xsearchz32xresultsYYYYY (Web source)
- Configure the Sitecore field to be treated as non-translated external field (`isExternal`)
  - a. showz32xinz32xsearchz32xresults

# External Fields Definition

There are two kind of external fields in Coveo for Sitecore. Both are not translated to the Coveo format by the `coveoFieldName/toCoveo` function.

- Sitecore fields

- Useful when you want to use the field on both Sitecore content and external content
- Non alpha-numeric characters are still translated to numbers
- Configurable with the "isExternal" attribute in the field map section

```
<fieldType fieldName="show in search results" isExternal="true" settingType="Coveo.Framework.Configuration.FieldConfiguration, Coveo.Framework" />
```

- Non-Sitecore fields that could be used in a Sitecore search page

- Useful when you want to use a Coveo system field or a field unique to an external source
- Configurable in a specific section in the field map

```
<!-- Fields in this section are fields that doesn't exist on Sitecore items, but exist in CES or in external sources. -->
<externalFields hint="raw:AddExternalField">
    <field fieldName="sysauthor" />
    <field fieldName="syscollection" />
    <field fieldName="syssource" />
    ...
</externalFields>
```

# Configuring an External Sitecore Field (1/3)

1. From the file system, browse to your Training configuration folder
  - a. <your sitecore instance folder>\Website\App\_Config\Include\Training
2. Create and open a "ExternalField.config" file
3. Initialize the file with this configuration to configure the "show in search results" field as an external field

```
<configuration>
  <sitecore>
    <coveo>
      <defaultIndexConfiguration>
        <fieldMap>
          <fieldNames hint="raw:AddFieldByFieldName">
            <fieldType fieldName="Show In Search Results" isExternal="true" settingType="Coveo.Framework.Configuration.FieldConfiguration,
Coveo.Framework" />
          </fieldNames>
        </fieldMap>
      </defaultIndexConfiguration>
    </coveo>
  </sitecore>
</configuration>
```

# Configuring an External Sitecore Field (2/3)

4. Save the file
5. Re-index the /sitecore/Content/Home tree (Developer > Re-Index Tree)
6. Log into the Coveo Cloud Administration Console
7. Navigate to the Content Browser section (Content > Content Browser)

# Configuring an External Sitecore Field (3/3)

8. In the Source facet on the left, select the "Coveo\_master..." item
9. Sort by indexed date descending
10. Wait for the transaction to be applied by clicking the search button every now and then until the indexed date of the first result matches the time when the re-index was performed
11. Expand one document properties
12. Validate that the "show in search results" field is now named "showz32xinz32xsearchz32xresults"

Template Manager - Glossary   ...		
FIELDS	PERMISSIONS	PERMISSION DETAILS
QUICK VIEW		
	show	X
Field	Value	
fshowz32xchildrenz32xinz32xmenu12897	1	
fshowz32xitemz32xinz32xmenu12897	1	
fshowz32xitemz32xinz32xsecondaryz32xmenu12897	1	
showz32xinz32xsearchz32xresults	1	

# Mapping a Field to an External Source (1/4)

1. Log into the Coveo Cloud Administration Console
2. Navigate to the Sources section (Content > Sources)
3. Select the "Helix" source
4. In the top toolbar, click the More button
5. Choose Manage mappings...
6. In the Edit the Mappings of a Source dialog
  - a. Click top-right ADD button
  - b. Choose Mapping

Field	Type	Rule	Facet	Multi-Value Facet	Sortable	Free Text
lastname	String	%[lastname]	X	X		X

SAVE AND REBUILD SOURCE    SAVE    CANCEL

# Mapping a Field to an External Source (2/4)

## 7. In the Apply a Mapping on All Item Types of a Source dialog

- a. Field: Type "show", then choose "showz32xinz32xsearchz32xresults" in the suggestions
- b. Apply to: Choose "All Item (common)"
- c. Rules: Type "1" (without quotes)
- d. Click Apply button

Apply a Mapping on All Item Types of a Source...

Field

showz32xinz32xsearchz32xresults

Apply To

All items (common)  Specific item types

Rules (?)

1

Enter a field mapping rule (e.g., %[metadataName])

# Mapping a Field to an External Source (3/4)

## 8. Click Save And Rebuild Source button

Edit the Mappings of a Source: Habitat ?

ADD ▾

COMMON SPECIFIC JSON

Field	Type	Rule	Facet	Multi-Value Facet	Sortable	Free Text
showz32xinz32xsearchz32xresults	String	1	X	X	X	X

SAVE AND REBUILD SOURCE    SAVE    CANCEL

9. Navigate to the Content Browser section (Content > Content Browser)
10. In the Source facet on the left, select the "Helix" item

# Mapping a Field to an External Source (4/4)

11. Sort by indexed date descending
12. Wait for the transaction to be applied by clicking the search button every now and then until the indexed date of the first result matches the time when the re-index was performed
13. Expand one document properties
14. Validate that the "showz32xinz32xsearchz32xresults" field is present and the value set to "1"

- Habitat Sitecore Framework Rammer		X	
FIELDS	PERMISSIONS	PERMISSION DETAILS	QUICK VIEW
	show		X
Field	Value		
showz32xinz32xsearchz32xresults	1		

# Excluding an External Source from a Filtering Rule (1/2)

1. From the file system, browse to the Training views folder
  - a. <your sitecore instance folder>\Website\Views\Training
2. Open your "LaunchJsExpressions.cshtml" file
3. Replace the first <script> by the one on the next page to also include external source content in the JavaScript filter
4. Save the file
5. Validate that you see the external content

# Excluding an External Source from a Filtering Rule (2/2)

```
<script>
  document.addEventListener('DOMContentLoaded', function() {
    var currentSiteExpressionBuilder = new Coveo.ExpressionBuilder();
    var fieldName = CoveoForSitecore.Context.fields.toCoveo('@@_path');
    var siteRootItemShortId = '@LaunchSitecore.Configuration.SiteConfiguration.GetHomeItem().ID.ToShortID()';
    currentSiteExpressionBuilder.addFieldExpression(fieldName, '==', [siteRootItemShortId]);

    var habitatSourceExpressionBuilder = new Coveo.ExpressionBuilder();
    fieldName = CoveoForSitecore.Context.fields.toCoveo('@@syssource');
    habitatSourceExpressionBuilder.addFieldExpression(fieldName, '==', ['Helix']);

    var mergedExpressionBuilder = Coveo.ExpressionBuilder.mergeUsingOr(
      currentSiteExpressionBuilder,
      habitatSourceExpressionBuilder
    );

    document.getElementById('currentSiteFilterExpression').dataset.scAdvancedFilter = mergedExpressionBuilder.build();
  });
</script>
```

# Publishing Changes

1. Smart Publish your site
2. Rebuild your Coveo\_web\_index using the Indexing Manager

# Validating Changes

1. In another browser or an incognito window, open your published site
  - a. <http://YourInstanceName>
2. Validate that the external content is returned

# Lab

1. Configure your Coveo Search Interface component external sources
2. Configure the "Show In Search Results" field as an external field
3. Map the "Show In Search Results" field to the Habitat source
4. Rebuild the Habitat source
5. Edit your JavaScript filter expression to include the Habitat documents
6. Smart publish your site
7. Rebuild the Coveo\_web\_index



# Styling

# Styling the Coveo Searchbox (1/4)

The Coveo JavaScript Search Framework (JS UI) uses CSS classes extensively and is very easy to customize to your needs.

1. From the file system, browse to the Launch Sitecore CSS assets folder
  - a. <your sitecore instance folder>\Website\assets\css
2. Create and open a new "training.css" file

# Styling the Coveo Searchbox (2/4)

## 4. Initialize the file with this code and the one on the next page

```
header .magic-box .magic-box-input .magic-box-underlay,  
header .magic-box .magic-box-input > input {  
    padding: 8px 0;  
    text-indent: 8px;  
    height: 35px;  
}  
header .magic-box .magic-box-input {  
    height: 35px;  
}
```

# Styling the Coveo Searchbox (3/4)

```
header .magic-box .magic-box-clear {  
    height: 35px;  
    line-height: 35px;  
}  
  
header .magic-box.magic-box-notEmpty .magic-box-clear {  
    width: 35px;  
}  
  
header .CoveoSearchButton {  
    height: 37px;  
    width: 50px;  
}  
  
header .CoveoSearchbox {  
    margin-top: 6px;  
    background-color: #ffffff;  
}
```

# Styling the Coveo Searchbox (4/4)

5. Save the file
6. From the file system, open the Launch Sitecore Main layout file
  - a. <your sitecore instance folder>\Website\Views\Shared\Main.cshtml
7. Look for the "responsive.css" stylesheet

```
<link rel="stylesheet" href="/assets/css/responsive.css" />
```
8. Insert the "training.css" stylesheet just after it

```
<link rel="stylesheet" href="/assets/css/responsive.css" />
<link rel="stylesheet" href="/assets/css/training.css" />
```
9. Save the file
10. Validate that the home page search box is now smaller



# Lab

## 1. Style the Coveo searchbox

# Applying Launch Sitecore Color Scheme to the JS UI (1/3)

Launch Sitecore comes with 8 site colors. Here's how to adapt some of the JS UI components to this color scheme.

1. From the file system, browse to the Launch Sitecore CSS assets folder
  - a. <your sitecore instance folder>\Website\assets\css
2. Edit the "color.css" file and append the CSS of next page at the end

# Applying Launch Sitecore Color Scheme to the JS UI (2/3)

```
/* Coveo Styles */
body .CoveoResultLink,
body a.CoveoResultLink,
body .CoveoResult a.CoveoResultLink,
body .CoveoResult a:hover,
body .CoveoResult a:visited,
body .CoveoResultLink:hover,
body .CoveoResultLink:visited,
body a.CoveoResultLink:hover,
body a.CoveoResultLink:visited,
body .CoveoResult a.CoveoResultLink:hover,
body .CoveoResult a.CoveoResultLink:visited,
body .coveo-icon-for-quickview-svg,
body .coveo-field-table-toggle,
body .coveo-field-table-toggle:hover,
body .coveo-field-table-toggle:visited,
body .coveo-facet-breadcrumb-value.coveo-selected *,
body .coveo-facet-slider-breadcrumb-value.coveo-selected *,
body .coveo-facet-breadcrumb-value.coveo-selected *,
body .coveo-facet-slider-breadcrumb-value.coveo-selected *,
body .coveo-breadcrumb-clear-all,
body .coveo-facet-header-eraser,
body .coveo-facet-settings-more-svg,
body .coveo-facet-more-icon-svg,
body .coveo-facet-less-icon-svg,
body .coveo-field-table-toggle-down-svg,
body .CoveoSearchInterface.coveo-small-tabs .coveo-dropdown-header,
body .CoveoSearchInterface.coveo-small-facets .coveo-dropdown-header,
body .CoveoSearchInterface.coveo-small-recommendation .coveo-dropdown-header,
body .coveo-facet-value.coveo-facet-search-button,
body .coveo-facet-search-select-all,
body .coveo-facet-header-title {
    color: #008998;
}
```

```
body .coveo-facet-search-input {
    margin-bottom: 0px;
    height: 16px;
}
```

# Applying Launch Sitecore Color Scheme to the JS UI (3/3)

3. Save the file
4. Validate that the search page displays its elements based on the configured site color

The screenshot shows the Sitecore search results page. The header features the Sitecore logo and navigation links for 'About Launch Sitecore', 'Team', and 'Glossary'. A search bar is on the right. The main content area shows a search result for 'Article' with 21 results. The results list includes a card for 'Basic Presentation Explanation' with details: Created By: sitecoremic, Created: 11/8/2011, Updated By: sitecorereadmin, Language: english, URI: <http://devtraining41.local/about-launch-sitecore/our-strategy/basic-presentation-explanation>. On the left, there are filters for 'Type' (Article checked, Term, Team Member, Article Group, Audience Segment) and 'Tags' (CMS, How-to).

# Lab

1. Style the Coveo for Sitecore components in the Launch Sitecore CSS color file

## 2. Customization

Q&A



### 3. Personalization & Usage Analytics

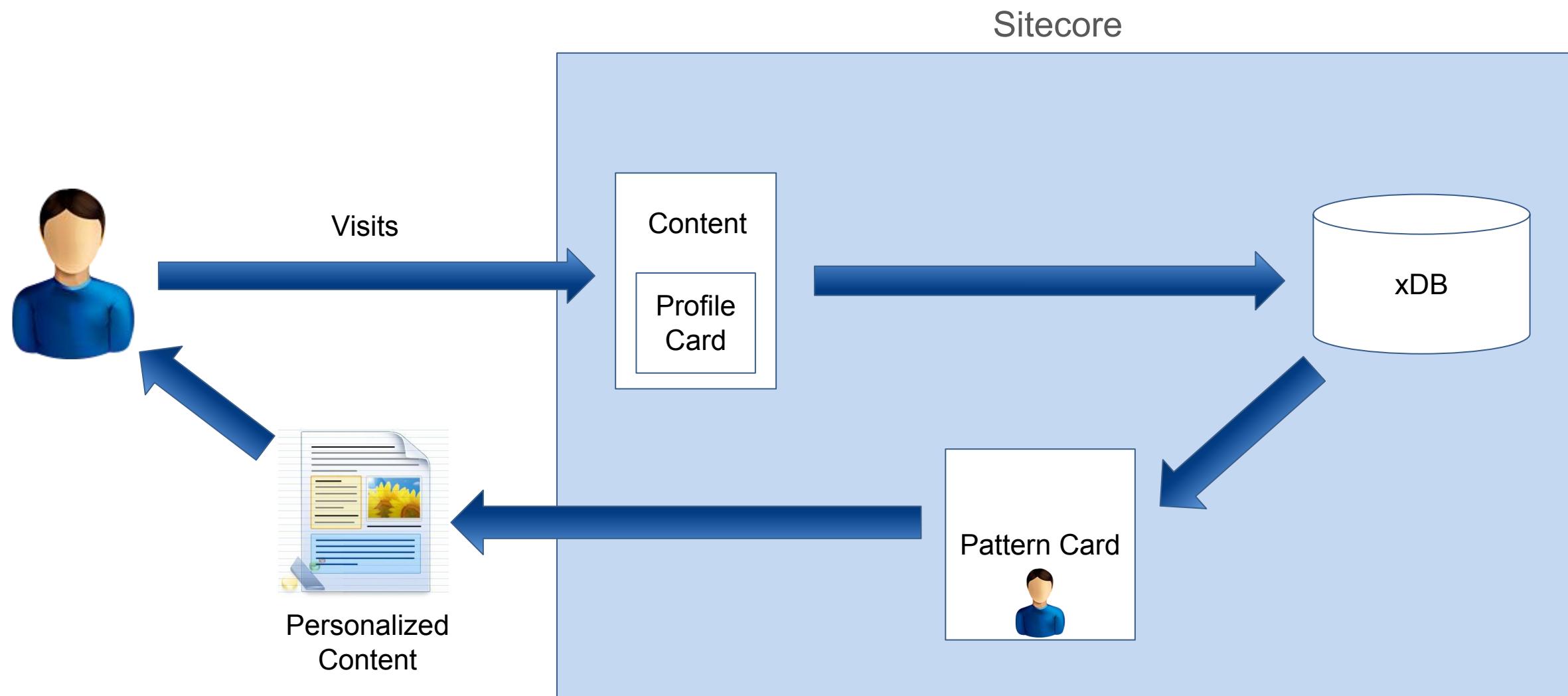
# 3. Personalization & Usage Analytics

1. Understanding the Sitecore xDB
2. Creating a Personalized Search Experience
  - a. Personalizing the Search Layout
  - b. Personalizing the Search Results
3. Understanding How Coveo for Sitecore Leverages the xDB Data
4. Usage Analytics
  - a. Coveo Usage Analytics
  - b. Sitecore Usage Analytics
5. Query Pipelines
6. Machine Learning

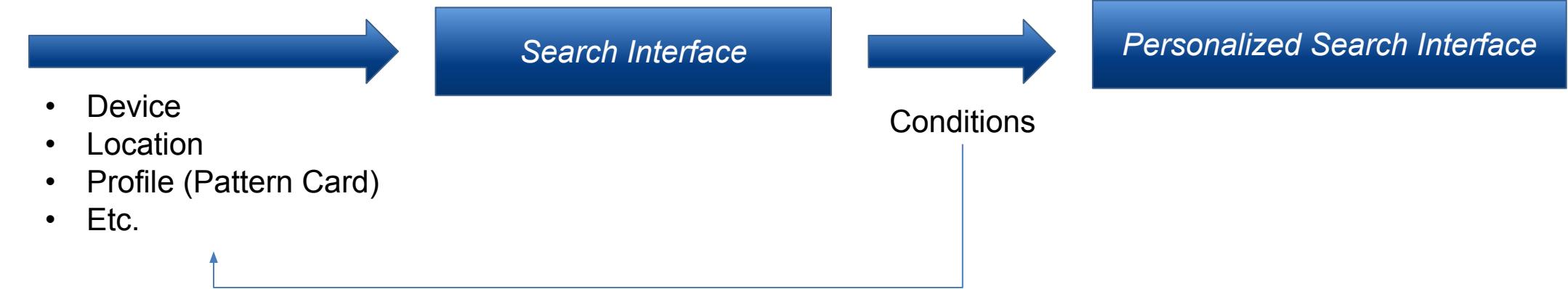


## Personalization

# Understanding the Sitecore xDB



# Creating a Personalized Search Experience



# Personalizing the Search Layout (UX)

Coveo for Sitecore components can be personalized based on conditions.

Conditions includes:

- Visitor information (Location, security, social...)
- Item information
- Profile card and pattern
- Visit
- ...

Coveo also created a rule to match visitors that previously performed a query  
"when the current visitor performed a query that compares to specific value"

# Hiding a Component for a Target Audience (1/3)

Let's say we want to hide the Tags facet to the digital marketers.

1. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
2. Access the Experience Editor (Publish > Experience Editor)
3. Select the Tags facet component
4. Click the personalization experience button in the floating toolbar
5. In the Personalize the Component dialog
  - a. Click the "New Condition" button 
  - b. Enter "Visitor is a digital marketer" in place of the condition name
  - c. Click the "Edit rule" button

# Hiding a Component for a Target Audience (2/3)

## 6. In the Create rule dialog

- a. Search for "pattern"
- b. Click the "where the current visit matches the specific pattern card in the specific profile" condition under the Visit category
- c. In the Edit rule part at the bottom, click these links:
  - i. Last "specific" link: Select the "Audience Segment"
  - ii. First "specific" link: Select the "Digital Marketer"
- d. Choose OK

## 7. Back in the Personalize the Component dialog

- a. Select the "Hide" radiobutton besides the new condition
- b. Choose OK

# Hiding a Component for a Target Audience (3/3)

8. Save the item
9. Switch to the Experience Explorer mode (Home > Other > Explore)
10. Switch for the Marketer preset (2nd to the left of Default)
11. Choose Apply
12. Validate that your page is hiding the Tags facet for digital marketers

# Lab

1. Hide the Tags facet to digital marketers using Sitecore personalization

# Changing a Component Parameters for a Target Audience (1/5)

Let's say we want developers to have the Tags facet selected items ANDed in addition to the default ORed mode.

1. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
2. Access the Experience Editor (Publish > Experience Editor)
3. Select the Tags facet component
4. Click the personalization experience button in the floating toolbar
5. In the Personalize the Component dialog
  - a. Click the "New Condition" button 
  - b. Enter "Visitor is a developer" in place of the condition name
  - c. Click the "Edit" button

# Changing a Component Parameters for a Target Audience (2/5)

## 6. In the Create rule dialog

- a. Search for "pattern"
- b. Click the "where the current visit matches the specific pattern card in the specific profile" condition under the Visit category
- c. In the Edit rule part at the bottom, click these links:
  - i. Last "specific" link: Select the "Audience Segment" profile
  - ii. First "specific" link: Select the "Developer" pattern
- d. Choose OK

## 7. Back in the Personalize the Component dialog

- a. Click the "..." button next to the "Content" text box

# Changing a Component Parameters for a Target Audience (3/5)

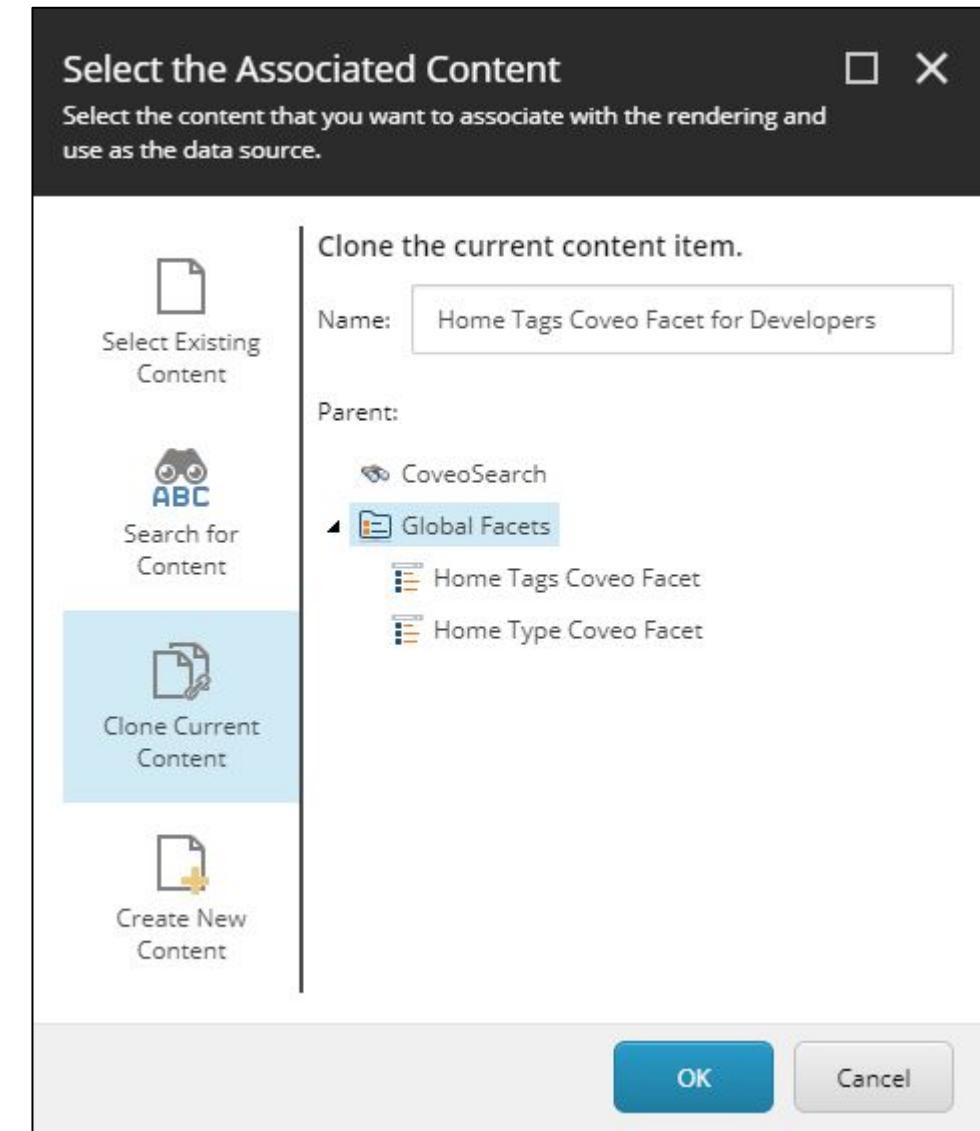
## 8. In the Select the Associated Content dialog

- a. In the left column, select Clone Current Content
- b. Name: Home Tags Coveo Facet for Developers
- c. Parent: Global Facets
- d. Choose OK

## 9. Back in the Personalize the Component dialog

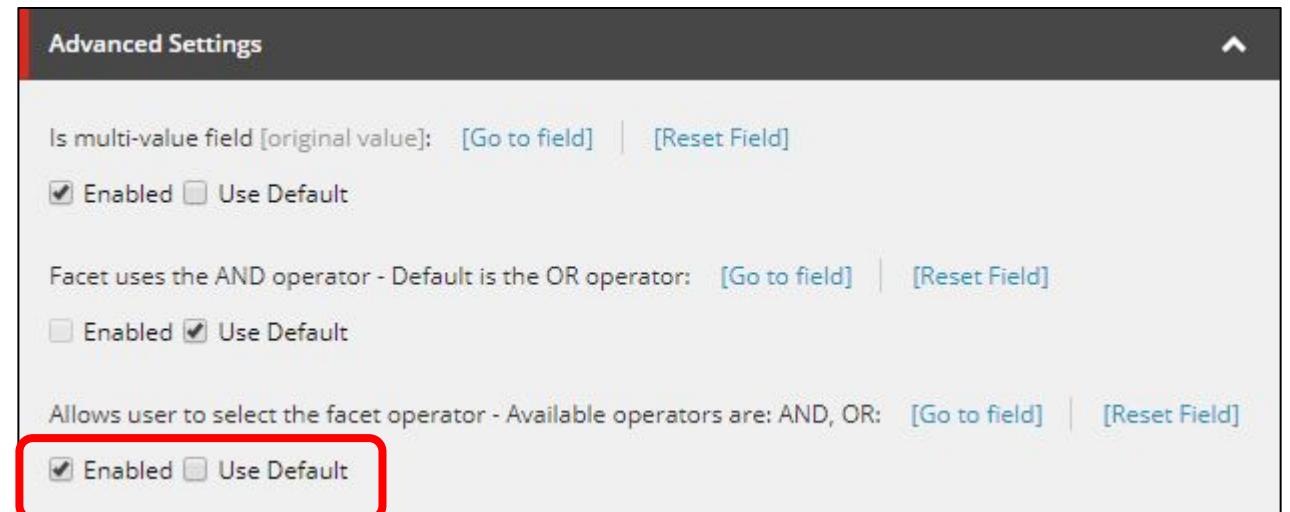
- a. Choose OK

## 10. Save the item



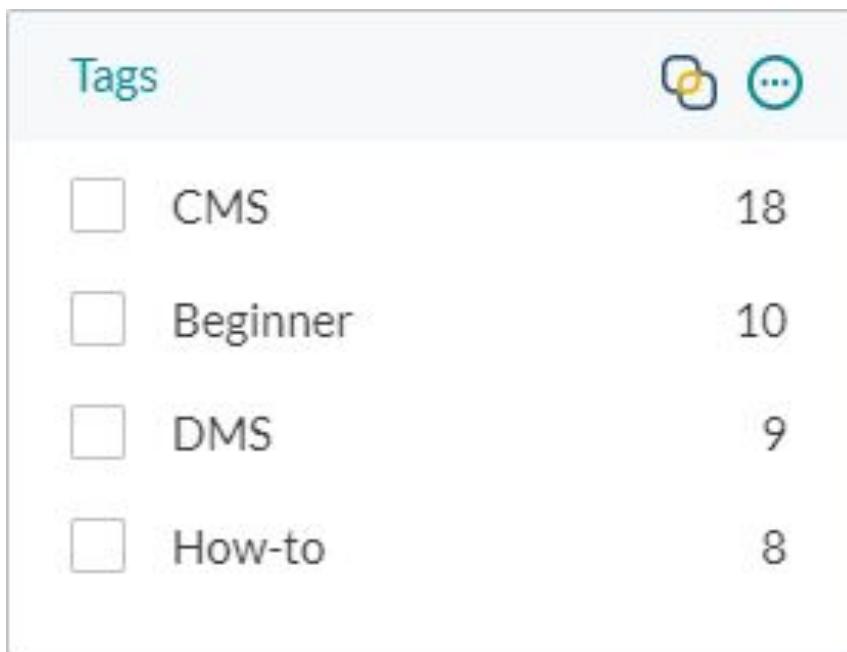
# Changing a Component Parameters for a Target Audience (4/5)

11. From the Sitecore Content Editor, select your "Home Tags Coveo Facet for Developers" item
  - a. /sitecore/Content/Home/Configuration/Coveo Hive Home Site Parameters/Global Facets/Home Tags Coveo Facet for Developers
12. In the "Allows user to select the facet operator" field (Advanced Settings)
  - a. Uncheck "Use Default", then check "Enabled"
13. Save the item



# Changing a Component Parameters for a Target Audience (5/5)

14. Back to the Experience Editor, switch to the Experience Explorer mode (Home > Other > Explore)
15. Switch for the Developer preset (4th to the left/right of Default)
16. Choose Apply
17. Validate that your Tags facet has a new operator icon for developers



# Lab

1. Modify the Tags facet behavior for developers using Sitecore personalization

# Personalizing the Search Results

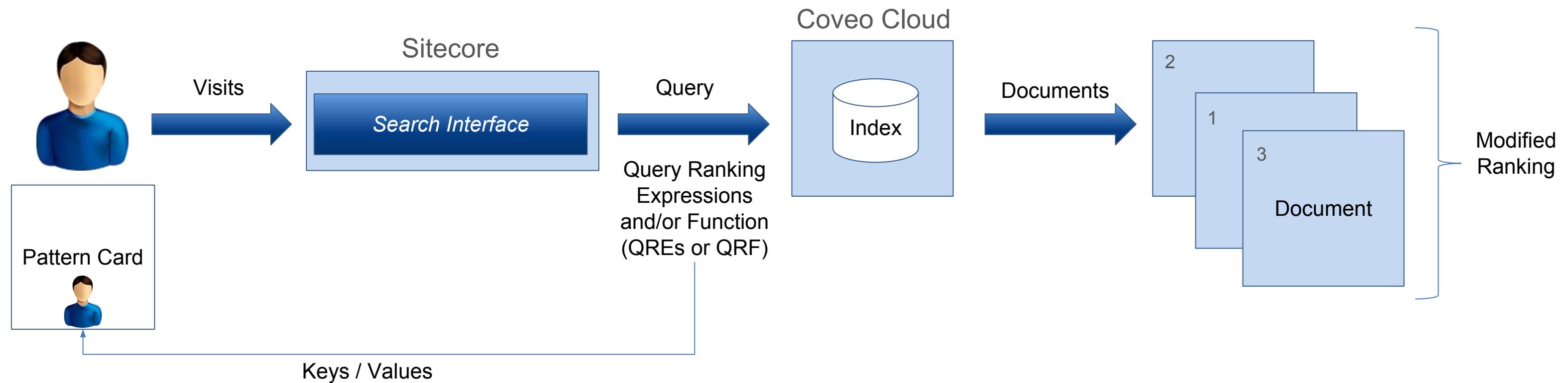
As we saw previously, Coveo for Sitecore search results can be boosted using the boosting rules.

Coveo created rules to use the analytics data to personalize search results:

- where the item profile card is specific profile card
- where the item specific profile key compares to specific numerical value
- where the item profile card is specific profile card and the current visit matches the specific pattern card in the specific profile

Coveo also added an automatic boosting mode to the Coveo Search component.

# How Coveo for Sitecore Uses the xDB Data



# Enabling Analytics Fields for Indexed Items (1/3)

By default, Coveo for Sitecore doesn't index the profile card and keys information to speed up queries. When using boosting based on profiles, we need to enable the indexing of those fields:

1. From the file system, browse to your Training configuration folder
  - a. <your sitecore instance folder>\Website\App\_Config\Include\Training
2. Create and open a "Analytics.config" file

# Enabling Analytics Fields for Indexed Items (2/3)

## 3. Initialize the file with this configuration for analytics fields to be indexed

```
<configuration>
  <sitecore>
    <coveo>
      <defaultIndexConfiguration>
        <indexAnalyticsFields>true</indexAnalyticsFields>
      </defaultIndexConfiguration>
    </coveo>
  </sitecore>
</configuration>
```

## 4. Save the file

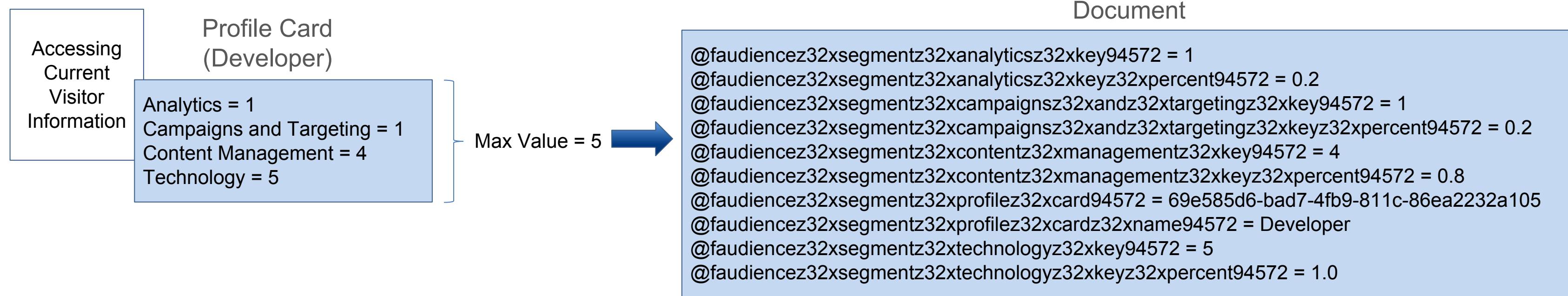
# Enabling Analytics Fields for Indexed Items (3/3)

5. Rebuild your Coveo master and web indexes from the Indexing Manager
6. Validate that your documents have the analytics fields in the Coveo Cloud Content Browser

Create a custom validator - About Launch Sitecore | Launch Sitecore x

FIELDS	PERMISSIONS	PERMISSION DETAILS	QUICK VIEW
audience			
Field	Value		
faudience32xsegmentz32xanalyticsz32xkey34976	1		
faudience32xsegmentz32xanalyticsz32xkeyz32xpercent34976	0.2		
faudience32xsegmentz32xcampaignsz32xandz32xtargetingz32xkey34976	1		
faudience32xsegmentz32xcampaignsz32xandz32xtargetingz32xkeyz32xpercent34976	0.2		
faudience32xsegmentz32xcontentz32xmanagementz32xkey34976	4		
faudience32xsegmentz32xcontentz32xmanagementz32xkeyz32xpercent34976	0.8		
faudience32xsegmentz32xprofilez32xcard34976	69e585d6-bad7-4fb9-811c-86ea2232a105		
faudience32xsegmentz32xprofilez32xcardz32xname34976	Developer		
faudience32xsegmentz32xtechnologyz32xkey34976	5		
faudience32xsegmentz32xtechnologyz32xkeyz32xpercent34976	1		

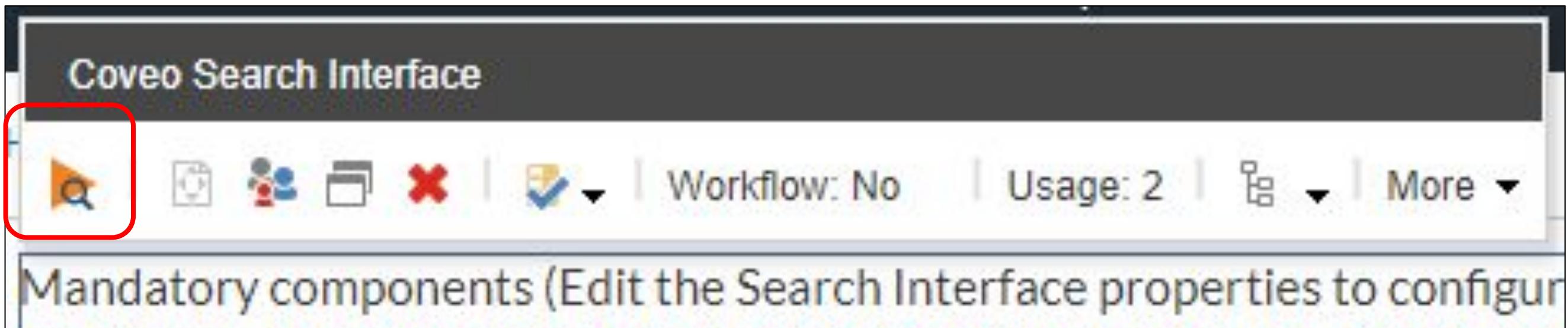
# How Coveo for Sitecore Indexes the Profile Cards



**Value = Profile Card Key Value / Max Key Value**  
Analytics Key Percent =  $1 / 5 = 0.2 = 0$   
Technology Key Percent =  $5 / 5 = 1$

# Automatic Boosting of the Search Results (1/2)

1. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
2. Access the Experience Editor (Publish > Experience Editor)
3. Select the Coveo Search Interface component
4. Click the Coveo experience button in the floating toolbar



# Automatic Boosting of the Search Results (2/2)

5. In the Control Properties dialog:

- a. Query Ranking
  - i. Check the "Enable automatic boosting" check box
  - ii. Check the "Audience Segment" profile check box
- b. Choose OK

6. Execute an iisreset

7. Switch to the Experience Explorer mode (Advanced > Other > Explore)

8. Validate that your page is displaying results in a different order for the default, Marketer, and Developer presets

- a. Select a preset
- b. Choose Apply

# How Automatic Boosting Works

Pattern Card  
(Developer)

Analytics = 1  
Campaigns and Targeting = 1  
Content Management = 4  
Technology = 5

Max Key Value = 5



Automatic boosting according to the user's pattern:

Enable automatic boosting

Audience Segment

Boosting influence (%):

Boosting weight range:

Dominant keys boosting influence (%):

Query

```
$qrf(expression:  
  exp(-(@faudiencez32xsegmentz32xanalyticsz32xkeyz32xpercent86020 - 0.2)^2 / (2 * 0.01)) * 20  
  +  
  exp(-(@faudiencez32xsegmentz32xcampaignsz32xandz32xtargetingz32xkeyz32xpercent86020 - 0.2)^2 / (2 * 0.01)) * 20  
  +  
  exp(-(@faudiencez32xsegmentz32xcontentz32xmanagementz32xkeyz32xpercent86020 - 0.8)^2 / (2 * 0.01)) * 80  
  +  
  exp(-(@faudiencez32xsegmentz32xtechnologyz32xkeyz32xpercent86020 - 1)^2 / (2 * 0.01)) * 150  
, normalizeWeight: True)
```

# Analytics Key (1 / 5)

$\exp(-(@faudiencez32xsegmentz32xanalyticsz32xkeyz32xpercent86020 - 0.2)^2 / (2 * 0.01)) * 20$

- @faudiencez32xsegmentz32xanalyticsz32xkeyz32xpercent86020
  - Items analytics key percentage value field
- - 0.2
  - Developer analytics profile key percentage (1 / 5)
- $\wedge 2$ 
  - Fixed value
- $(2 * 0.01)$ 
  - 2
    - Fixed value
  - 0.01
    - Boosting weight range (1) / 100
- $* 20$ 
  - Developer analytics profile key percentage (0.2) \* Boosting influence (100) = 20

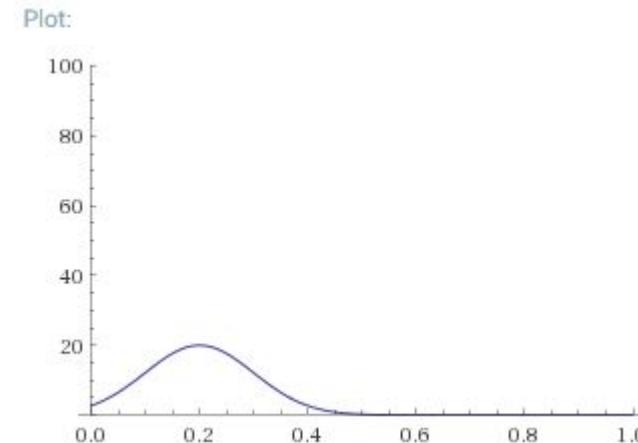
# Technology Key (5 / 5)

$\exp(-(@audiencez32xsegmentz32xtechnologyz32xkeyz32xpercent86020 - 1)^2 / (2 * 0.01)) * 150$

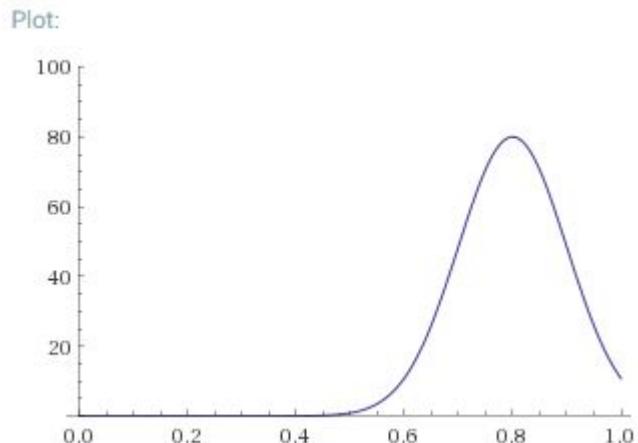
- $@audiencez32xsegmentz32xtechnologyz32xkeyz32xpercent86020$ 
  - Items technology key percentage value field
- $- 1$ 
  - Developer technology profile key percentage (5 / 5)
- $(2 * 0.01)$ 
  - 0.01
    - Boosting weight range (1) / 100
- $* 150$ 
  - Developer technology profile key percentage (1) \* Boosting influence (100) = 100
  - Plus (Developer technology profile key percentage (1) \* Boosting influence (100)) \* Dominant keys boosting influence (50) = 50
  - $100 + 50 = 150$

# How Many Points Are Documents Boosted?

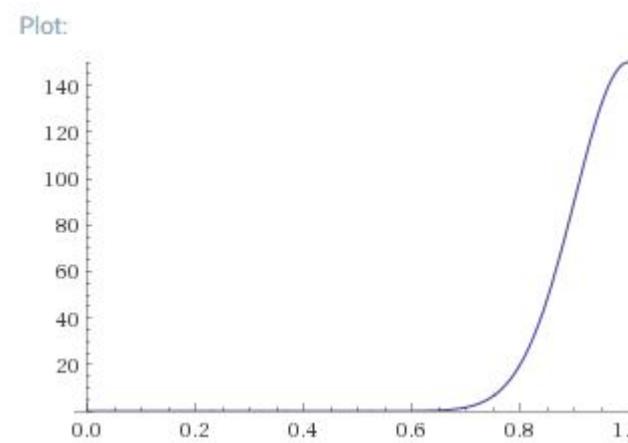
Analytics &  
Campaigns and  
Targeting Keys (1 / 5)



Content Management  
Key (4 / 5)



Technology Key  
(5 / 5)



# Lab

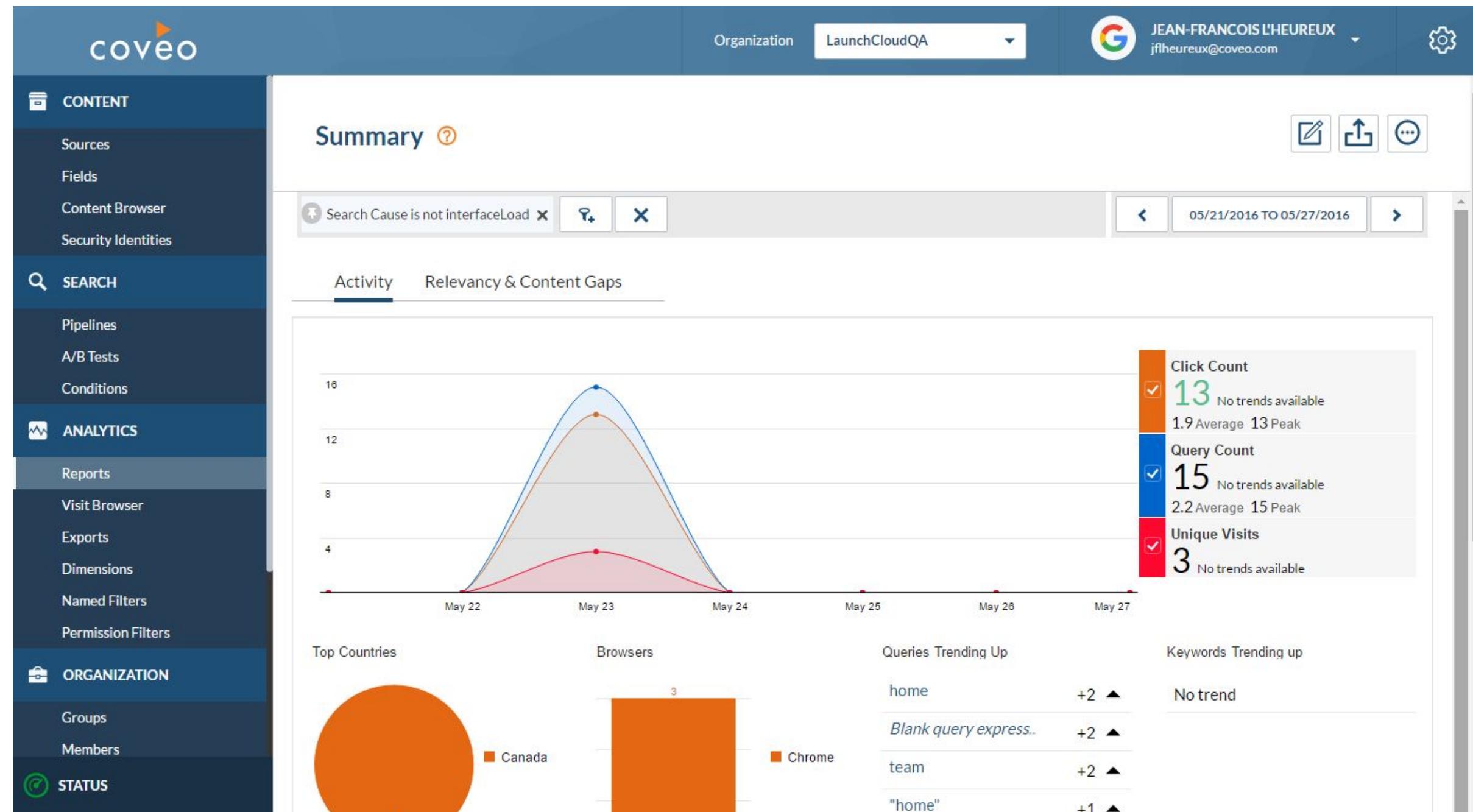
## 1. Configure automatic boosting



# Usage Analytics

# Coveo Usage Analytics

The Coveo Usage Analytics is a set of tools to help you visualize what has been recorded on your Coveo search components.



# Activating Coveo Usage Analytics

Coveo Usage Analytics feature is included in all editions but Free.

It can be activated by inserting the "Coveo For Sitecore Analytics" component in your search interface.



You can also insert the "Coveo User Context" resource component to add more information to the logged usage analytics.

Note that the logging occurs only for visits on the published site. Visits from the Experience Editor are not logged.

# Usage Analytics Events

Each user action triggers a usage analytics action

The search page will send events in 4 categories:

- Search
- Click
- PageView
- Custom

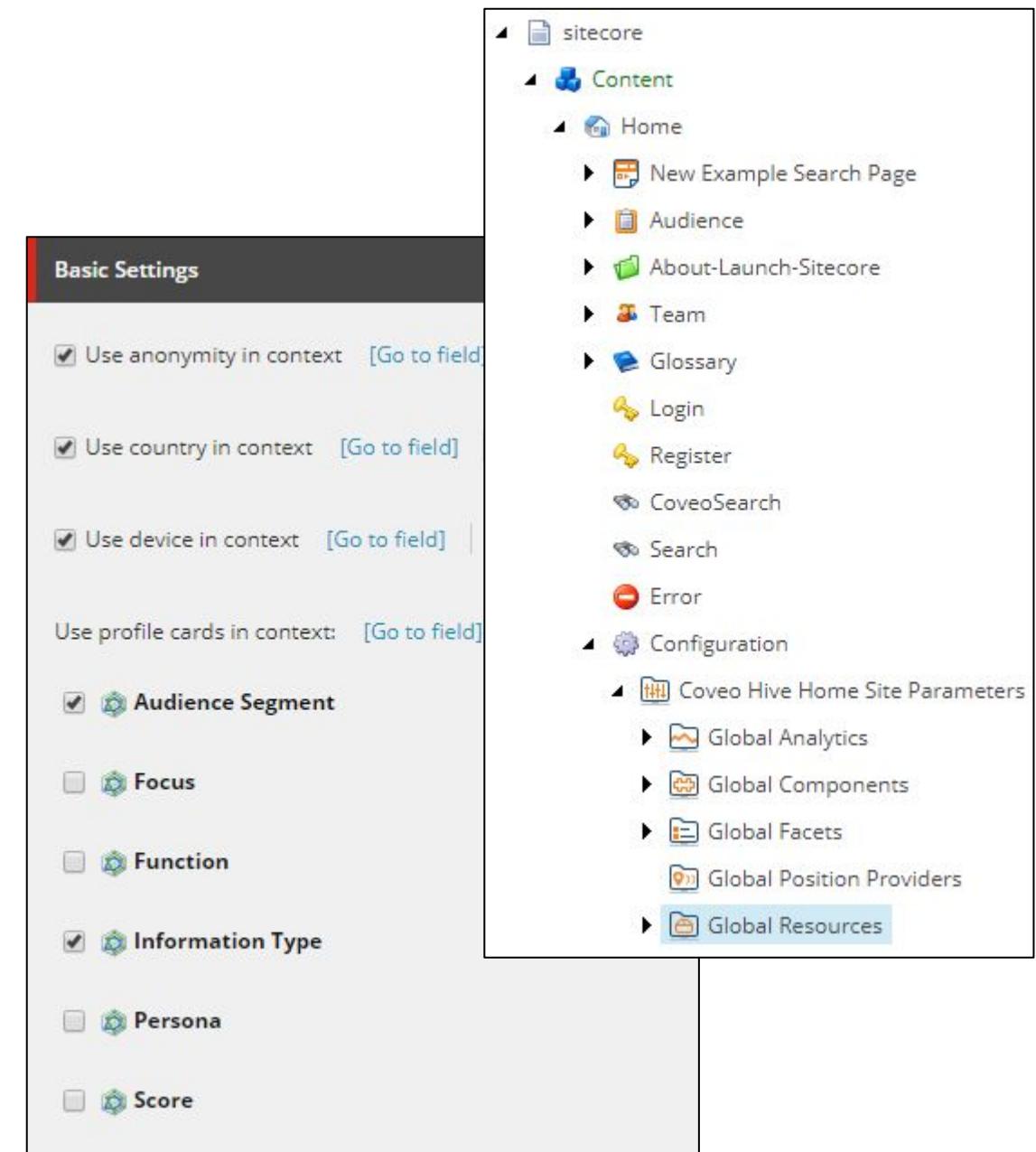
Each event contains a large amount of context metadata

# Usage Analytics Click Event Example

```
{  
  "actionCause": "documentOpen",  
  "actionType": "document",  
  "anonymous": true,  
  "device": "Chrome",  
  "mobile": false,  
  "language": "en",  
  "originLevel1": "CoveoSearch",  
  "originLevel2": "default",  
  "customData": {  
    "documentURL": "/en/Team/Patrick-Wallace",  
    "documentTitle": "Patrick Wallace - Team | Launch Sitecore",  
    "JSUIVersion": "1.0.420.31;0.0.0.0"  
  },  
  "userAgent": "Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.103 Safari/537.36",  
  "searchQueryUid": "b212446b-618d-40a5-bea8-0dc76b44ad9d",  
  "queryPipeline": "default",  
  "documentUri": "sitecore://database/web/ItemId/19903DBE-14B7-455C-AB56-6FD2EC3D3C8E/Language/en/Version/1",  
  "documentUriHash": "hs9x20ñNEy01ugZA",  
  "documentUrl": "/en/Team/Patrick-Wallace",  
  "documentTitle": "Patrick Wallace - Team | Launch Sitecore",  
  "collectionName": "default",  
  "sourceName": "Coveo_web_index - WKS-000242-sc80rev150812_Completed",  
  "documentPosition": 1,  
  "responseTime": 0,  
  "viewMethod": "documentOpen",  
  "originLevel3": "/sitecore/content/Home/CoveoSearch"  
}
```

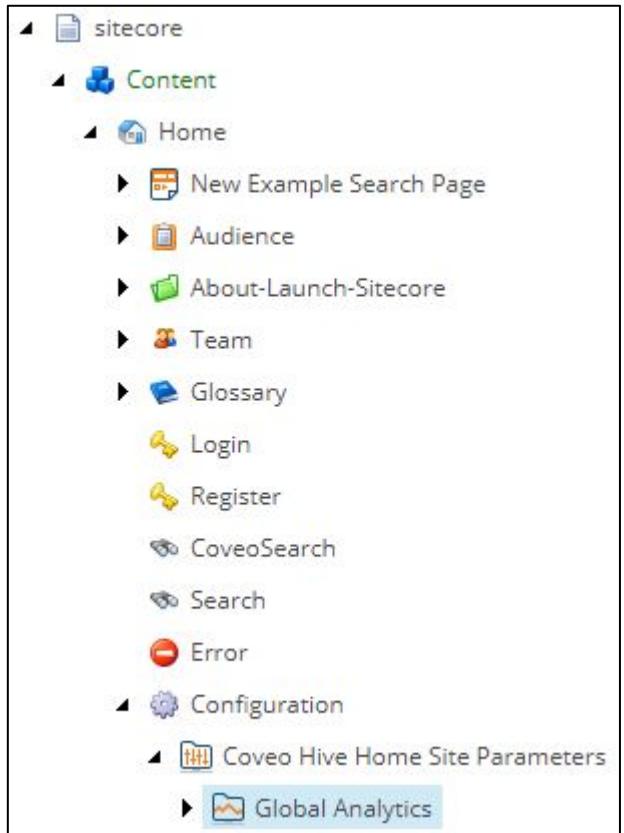
# Logging Page Views and User Context (1/4)

1. From the Sitecore Content Editor, select the Coveo Hive Global Resources folder
  - a. /sitecore/Content/Home/Configuration/Coveo Hive Home Site Parameters/Global Resources
2. Insert a new Coveo User Context (right-click > Insert > Coveo User Context)
  - a. Name: Home Coveo User Context
3. In the Use profile cards in context field (Basic Settings), select these profiles:
  - a. Audience Segment
  - b. Information Type
4. Save the item



# Logging Page Views and User Context (2/4)

5. Select the Coveo Hive Global Analytics folder
  - a. /sitecore/Content/Home/Configuration/Coveo Hive Home Site Parameters/Global Analytics
6. Insert a new Coveo Page View Analytics  
(right-click > Insert > Coveo Page View Analytics)
  - a. Name: Home Coveo Page View Analytics



# Logging Page Views and User Context (3/4)

7. From the file system, open the Launch Sitecore Main layout file
  - a. <your sitecore instance folder>\Website\Views\Shared\Main.cshtml
8. At the end of the file, before </body>, insert these two static renderings

```
<!-- Coveo Search Resources -->
@Html.Sitecore().Rendering("{41C7C6DD-C132-44BD-890A-B035056BC054}")
<!-- Coveo User Context -->
@Html.Sitecore().Rendering("{8231CD1A-3377-4FE8-A27C-F6382ACB8C52}", new { DataSource = "{ID of your Home Coveo User Context item}" })
<!-- Coveo Page View Analytics -->
@Html.Sitecore().Rendering("{DC336A78-04E4-4299-ACE8-39160A6AE863}", new { DataSource = "{ID of your Home Coveo Page View Analytics item}" })
</body>
</html>
```

9. Replace the data source IDs by the ones of your items
10. Save the file

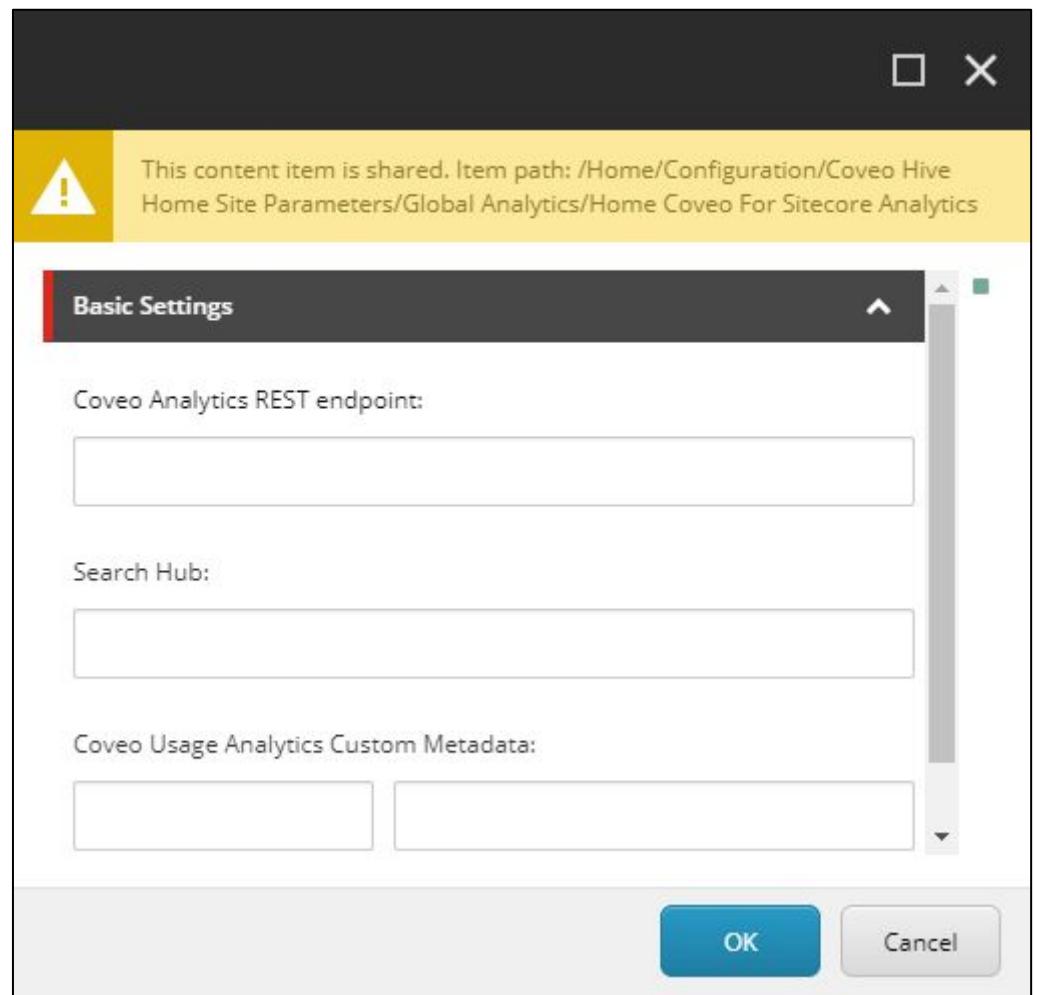
# Logging Page Views and User Context (4/4)

- 11.** From the Sitecore Content Editor, smart publish the site (Publish tab > Publish dropdown button > Publish Site option > Smart publish radiobutton > Publish button)
- 12.** By looking at the network tab of your browser developer tools, validate that visiting the published website pages logs:
  - a. Page view events
  - b. The user context in all events

# Sending Custom Metadata

Coveo for Sitecore allows you to add additional metadata to all the events happening in your search page.

You can set that in the data source item of the Coveo For Sitecore Analytics component.



# Sending Custom Events

It is also possible to send your own custom event.

# Logging in the Coveo Usage Analytics

1. Load the published version of the search page
2. Perform a couple of queries
3. Click around on documents, Quick view, facets, tabs, etc
4. Log into the Coveo Cloud Administration Console
5. Navigate to the Analytics reports (Analytics > Reports)
6. Double click the Summary dashboard
7. You'll see the new data in the dashboard **15 minutes after your first query**

The screenshot shows the Coveo Usage Analytics interface. On the left, there's a sidebar with 'CONTENT' and 'SEARCH' sections, each containing links like Sources, Fields, Content Browser, Security Identities, Search Pages, Pipelines, and A/B Tests. The main area is titled 'Reports' with a subtitle: 'Monitor and measure your Coveo Cloud search solution usage and adoption with easy-to-create and customizable analytics reports.' It features a table with four rows:

Name	Type
Summary	Dashboard
Activity	Explorer
Content Gaps	Explorer

A dropdown menu shows 'All (4)' and a 'Filter' button. At the top right, there's a user dropdown for 'jflheureux - TrainingClou...', a 'G' icon, and a gear icon.

# Coveo Usage Analytics - Demo

- Dashboards & Explorers Reports
  - Custom Reports
  - Named Filters
  - Permission Filters
- Dimensions and Metrics
  - Custom Dimensions
- Visit Browser
- Data Exports

# Sitecore Usage Analytics (xDB Events)

Coveo for Sitecore can log all the visitors actions in the global search box and search pages in the Sitecore Analytics (xDB). The actions are logged as page events for the search page visits.

This can be activated by inserting components in your search interface

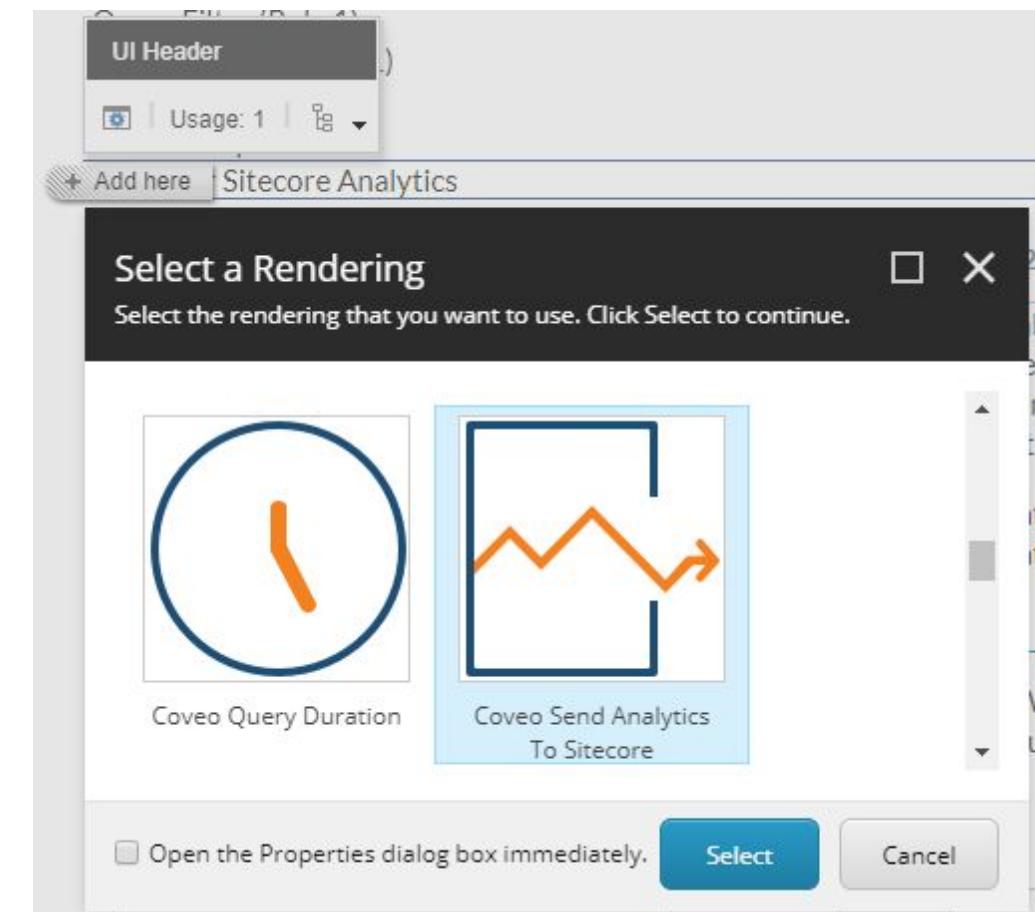
- Coveo For Sitecore Analytics
- Coveo Send Analytics To Sitecore



Note that the logging occurs only for visits on the published site. Visits from the Experience Editor are not logged.

# Enabling Sitecore Usage Analytics Logging (1/3)

1. From the Sitecore Content Editor, select your "CoveoSearch" item
  - a. /sitecore/Content/Home/CoveoSearch
2. Access the Experience Editor (Publish > Experience Editor)
3. Select the UI Header placeholder
4. Click the "Add here" button
5. In the Select a Rendering dialog
  - a. Select Coveo Send Analytics To Sitecore
  - b. Choose Select

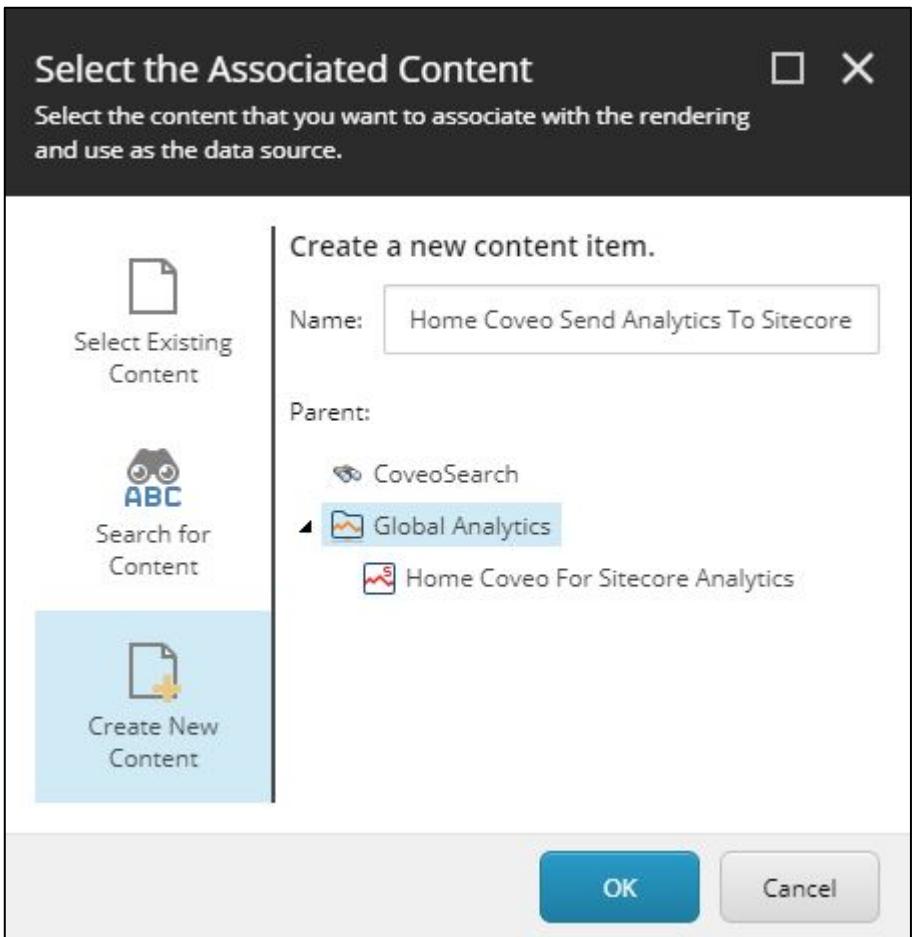


# Enabling Sitecore Usage Analytics Logging (2/3)

## 6. In the Select the Associated Content dialog

- a. In the left column, select Create New Content
- b. Name: Home Coveo Send Analytics To Sitecore
- c. Parent: Global Analytics
- d. Choose OK

## 7. Save the item



# Enabling Sitecore Usage Analytics Logging (3/3)

8. From the Sitecore Content Editor, smart publish the site (Publish tab > Publish dropdown button > Publish Site option > Smart publish radiobutton > Publish button)
9. By looking at the network tab of your browser developer tools, validate that visiting the published website pages logs events in Sitecore Usage Analytics
  - a. You should see requests to /coveo/rest/v6/analytics...

# Sitecore Experience Profile

OVERVIEW ACTIVITY PROFILING DETAILS SOCIAL

### Latest events

- Search Box Clear  
02.11.2016 13:37:30 (1D:3hr:38min)  
</coveosearch>
- Breadcrumb Click  
"Article Group" selected from facet identified by "\_EE998D0F-4B87-429E-9CE8-0DC7463C0A90".  
02.11.2016 13:37:29 (1D:3hr:38min)  
</coveosearch>
- Facet Selection  
"Article Group" selected from facet identified by "\_EE998D0F-4B87-429E-9CE8-0DC7463C0A90".  
02.11.2016 13:27:47 (1D:3hr:48min)  
</coveosearch>
- Search  
Keyword: "developing"  
02.11.2016 13:27:36 (1D:3hr:48min)  
</coveosearch>

### Best pattern matches

- Digital Marketer  
Latest match: 0 visit ago, 1D:3hr:48min
- High Level  
Latest match: 0 visit ago, 1D:3hr:48min

Best two pattern matches - see more information [here...](#)

### Recent campaigns

No campaigns found

# Sitecore Experience Profile

Coveo events and keywords are available for the Experience Profile contacts only when:

- Contact is not anonymous or contact is anonymous and:
  - Sitecore 8.2 Update-2 or older is used
  - Sitecore 8.2 Update-3 or newer is used and  
ContentSearch.Analytics.IndexAnonymousContacts setting is set to true
- ASP.Net session has expired (default timeout is 20 minutes)
- Sitecore reporting process has executed



# Query Pipelines

# Coveo Query Pipelines

The query pipelines allow you to add additional logic between your search components and the Coveo index.

It is independent from your application, so if you have multiple Sitecore, or even external Coveo search interfaces, the query pipeline can be used by all of them.

# Coveo Query Pipelines - Advanced Customization

Query pipelines are powerful and can change the dynamic of any search components in a few clicks.

Be aware that assigning pipelines can also be done through custom code, which gives a lot of flexibility to the developers. For example:

- Assign a thesaurus-free pipeline if a user clicks on a certain button
- Assign a pipeline with top results for user visiting a certain number of pages
- Assign a Coveo Machine Learning powered pipeline for users based on their personas

# A/B Testing

Each query pipeline have a different effect on the search experience, and tracking this effect is the key.

A/B testing allows you to run two pipelines in parallel. You can then create reports based on the output of each pipelines.

# Coveo Query Pipelines - Demo

- Thesaurus
- Featured Results
- Stop Words
- Ranking Expressions
- Ranking Weights
- Triggers



# Machine Learning

# Coveo Machine Learning

Although the tools and the analytics are at your fingertips, the large amount of data to analyze and decision to take can be overwhelming.

Coveo Machine Learning will learn from your analytics data and use all the Coveo tools to improve your relevance, reduce your queries without results and improve your click through rate.

Coveo Machine Learning uses a customizable ranking weight, which means that it will not override other ranking algorithms. You can use it in complement to other Coveo tools.

# Coveo Machine Learning - Features

Coveo Machine Learning is available in all Coveo for Sitecore Cloud Editions.

It offers the following features:

- Automatic Relevance Tuning (ART)
  - Reordering and injection of search results to improve relevance
- Query Suggestions
  - Suggestions based on successful searches (with a click on a result) from past visitors
- Recommendations (Enterprise Cloud Edition only)
  - Result recommendations based on past users visits

# Coveo Machine Learning - Demo

- Creating Models
  - ART
  - Query Suggestions
  - Recommendations

# 3. Personalization & Usage Analytics

## Q&A



## 4. Scaling, Upgrading & Troubleshooting

# 4. Scaling, Upgrading & Troubleshooting

## 1. Scaling

## 2. Upgrading

- a. Coveo for Sitecore
- b. Sitecore

## 3. Troubleshooting

# Index Redundancy

Coveo Cloud automatically replicates a production organization index to offer redundancy and scaling. Sandboxes organizations have only one copy of the index.

An on-premises CES7 index needs to be replicated by the use of CES mirrors and network load-balancers (NLB).

# Installing Coveo for Sitecore on Multiple CM/CD Servers

Click on the link above for a step by step guide on scaling Coveo for Sitecore

# Upgrading Coveo for Sitecore

- 1.** The latest version of the solution can be found in multiple locations:
  - a. <http://productupdate.coveo.com/?product=sitecore>
  - b. <https://developers.coveo.com/display/SitecoreV4/Downloads>
  - c. <http://www.coveo.com/en/solutions/coveo-for-sitecore/download>
- 2.** Follow the [monthly upgrade documentation pages](#)
  - a. There's a quick step by step to upgrade guide for each release of Coveo for Sitecore
  - b. If you skip releases, follow all the steps of all the skipped releases too
    - i. Example: Upgrading from April 2016 to July 2016
    - ii. Download the July release
    - iii. Read and merge in a single procedure, the upgrade documentation of April to June and of June to July
    - iv. Follow your merged procedure when installing the July release
- 3.** Keep an eye on our [best practices!](#)

# Upgrading Sitecore

1. To a new update revision
2. To a new minor or major version

In all cases:

1. Ensure Coveo for Sitecore is compatible with the desired Sitecore version
2. Backup your Sitecore instance
  - a. All files and databases
  - b. Do not forget to backup all the files you customized or created in relation to Coveo for Sitecore
3. Disable all the Coveo for Sitecore configuration files
4. Upgrade Sitecore to desired version

# Upgrading Sitecore to a New Update Revision

5. Enable all the Coveo for Sitecore configuration files
6. Rebuild the Coveo indexes
7. Verify Coveo for Sitecore is working correctly
8. Optionally upgrade Coveo for Sitecore to the latest release

# Upgrading Sitecore to a New Minor or Major Version

5. Download and install the Coveo for Sitecore package built for the new Sitecore version (same release number you had before)
6. Compare the disabled configuration files with the new ones just installed
7. Merge the disabled configuration files customizations with the new ones just installed (passwords, usernames, settings, custom indexes, fields configuration, computed fields...)
8. Enable all the new Coveo for Sitecore configuration files (removing .example extension)
9. Rebuild the Coveo indexes
10. Verify Coveo for Sitecore is working correctly
11. Optionally upgrade Coveo for Sitecore to the latest release

# Troubleshooting

1. Figure out what type of issue you are facing
  - a. Indexing
  - b. Querying
  - c. UI glitch
  - d. JavaScript error
2. Take a look at the Diagnostic Page
3. Take a look at the Sitecore logs
4. Browse/search the [online resources!](#)
5. Solution not found?
  - a. Development issue, ask a question on [answers.coveo.com](#)
  - b. Setup issue, [open a support ticket](#)

# 4. Scaling, Upgrading & Troubleshooting

## Q&A



# Advanced Configuration Topics

- Advanced upgrading advice
- Advanced Coveo Cloud Administration Console use
- Coveo for Sitecore pipelines
- Multi sites support
- Virtual users support

# Advanced Customization Topics (1/2)

- Slider and range facets
- Extending Coveo for Sitecore component properties
- Duplicating Coveo for Sitecore facets/sort components
- LINQ module
- Creating custom filtering/boosting rules
- Facet computed field (side caption. Not to confuse with index computed fields)

# Advanced Customization Topics (2/2)

- Mobile optimized search interface
- Web crawler optimized search interface
- Browser support gotchas
- Search-driven components