SharePoint 2013 Development

Architectural Analysis

This document explores SharePoint 2013 development.

2014

Heary, James

10/10/2014

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 09/23/2014 | 1.0 | Created | Jim Heary |
|  |  |  |  |
|  |  |  |  |

Table of Contents

Introduction 3

Overview 3

Common uses of SharePoint 3

Platform Architecture 3

Core Functionality 4

What’s New for Developers 5

Cloud App Model 5

Programming Model 5

Development Tools 5

Core Platform Enhancements 6

Mobility 6

Social Collaboration 6

Search 6

Workflows 6

Enterprise Content Management 6

Business Connectivity Services 7

Application Services 7

SharePoint Development 8

Capabilities 8

Programming Models 8

Apps 8

Publishing Sites 8

Farm Solutions 8

Mobile Apps 9

Reusable Components 9

Client API’s 9

Server Object Model (SSOM) 9

Client Object Models for Managed Code (CSOM) 9

JavaScript Object Model 10

REST/OData Endpoints 11

WCF Data Services Framework 11

Deprecated API Sets 11

Search 12

Architecture 12

Extensibility Points 13

Making Content Available for Search 15

Supported Search Syntax 16

Search Query with SharePoint Search Web Service (Deprecated) 16

Appendix 19

Keyword Query Language (KQL) Syntax Reference 19

Elements of a KQL Query 19

Constructing Free-Text Queries using KQL 19

Property Restriction Queries in KQL 20

KQL Operators for Complex Queries 21

SharePoint Search Web Service 23

# Introduction

## Overview

SharePoint is a web application framework and platform that integrates intranet, content management and document management, but recent versions have broader capabilities. SharePoint comprises a multipurpose set of Web technologies backed by a common technical infrastructure. By default, SharePoint has a Microsoft Office-like interface, and it is closely integrated with the Office suite. The web tools are intended for non-technical users. SharePoint can provide intranet portals, document & file management, collaboration, social networks, extranets, websites, enterprise search, and business intelligence. It also has system integration, process integration, and workflow automation capabilities.

### Common uses of SharePoint

* Intranet portals- a way to centralize access to enterprise information and applications on a corporate network. It is a tool that helps a company manage its data, applications and information more easily.
* Enterprise content and document management- store and track electronic documents or images of paper documents. It is usually also capable of keeping track of the different versions created by different users. In addition to being a platform for digital record management systems that meet government and industry compliance standards, SharePoint also provides the benefit of a central location for storing and collaborating on documents, which can significantly reduce emails and duplicated work in an organization.
* Extranet portal- can be used to provide password-protected, web-facing access to people outside an organization. Organizations often use functionality like this to integrate third parties into supply chain or business processes, or to provide a shared collaboration environment.
* Internet sites- using the 'Publishing' features, SharePoint can be used to manage larger public websites.
* Software framework- SharePoint is built upon the ASP.Net framework and provides an additional layer of services and code base to greatly reduce the amount of custom development required to provide a working application.

### Platform Architecture

The SharePoint platform is a flexible, *n*-tier service-oriented architecture (SOA). It can be scaled down to operate entirely from one machine, or scaled up to be managed across hundreds of machines.

Farms- farms are a logical grouping of SharePoint servers that share common resources. A farm typically operates stand-alone, but can also subscribe to functions from another farm, or provide functions to another farm. Each farm has its own central configuration database, which is managed through either a PowerShell interface, or a Central Administration website. Each server in the farm is able to directly interface with the central configuration database.

Web applications**-** web applications are top-level containers for content in a SharePoint farm, and are typically the interface through which a user interacts with SharePoint. A web application is associated with a set of access mappings or URLs defined in the SharePoint central management console, then automatically replicated into the IIS configuration of every server configured in the farm. Web applications are typically independent of each other, have their own application pools, and can be restarted independently in Internet Information Services.

Site collections**-** site collections are used to provide a grouping of 'SharePoint Sites'. Each web application typically has at least one site collection. Site collections may be associated with their own content databases, or they may share a content database with other site collections in the same web application. A site collection may contain one or more sites. A site collection is a group of sites that are functionally, navigationally, and administratively related to one another.

Service applications**-** service applications provide granular pieces of SharePoint functionality to other web and service applications in the farm. The Search Indexing service is an example. A service application can be turned off, exist on one server, or load-balanced across many servers in a farm. Each service application enabled on the farm typically has its own process, and typically also has its own configuration database and Active Directory service account.

### Core Functionality

#### Sites

A SharePoint Site is a collection of pages, site templates, lists, and libraries configured for the purpose of achieving an expressed goal. A site may contain sub-sites, and those sites may contain further sub-sites. Typically, sites need to be created from scratch, but sites can also be created according to packaged functionality. Examples of Site templates in SharePoint include: blogs, collaboration (team) sites, documents, and meetings.

#### Lists and libraries

Lists and libraries have the same properties. SharePoint comes with some pre-defined list and library definitions including Announcement Lists, Blogs, Contacts, Discussion Boards, Document Libraries, External Content (BCS) lists, Pages, Surveys, and Tasks.

Lists have features such as workflows, item-level or list-level permission, version history tracking, multiple content-types, and external data sources. SharePoint supports the creation of multiple views of a list or library, including Gantt chart and calendar views. Views can be personal or distributed to a group of users. Lists support item (document or record) level security permissions, where each list item can have a unique permission level.

A Library is a list where each item in the list refers to a file that is stored in SharePoint. Libraries have all the same behaviors as lists, but because libraries contain files, they have extra features. One of these is the ability to be opened and modified through a compatible WebDAV client (e.g. Windows Explorer).

#### Web-parts

Web-parts are sections that can be inserted into Pages in SharePoint sites. These sections are UI Widgets whose typical uses are displaying content defined in the web-part's settings, displaying items from Lists/Libraries, providing access to features in the SharePoint platform (e.g. Search), and providing a user interface into other products (e.g. Microsoft Reporting Services).

#### Pages

SharePoint provides three primary page content-types: Wiki page, Web-part page, and Publishing pages. The default page type is wiki page, which enables free-form editing access from the ribbon toolbar. Users can also insert web-parts into any page type.

#### Search

SharePoint Foundation contains a limited search engine that provides the ability to search within documents and, except in cloud environments, across external data sources (such as file systems).

#### Compliance, standards and integration

* Integrates with Microsoft Office 2007 and 2010.
* Uses Microsoft's OpenXML document standard for integration with Microsoft Office.
* Provides various APIs: client-side, server-side, and JavaScript with REST, SOAP and OData based interfaces.
* Can comply with many document retention, record management, document ID and discovery laws.
* Compatible with the Content Management Interoperability Standard (CMIS).
* Produces valid XHTML 1.0 that is compliant with WCAG 2.0 AA accessibility standards.
* SharePoint can use claims-based authentication, relying on SAML tokens for security assertions.

## What’s New for Developers

SharePoint 2013 is a versatile development platform for building apps and solutions with varying scopes that address a wide range of needs. Developers familiar with SharePoint know they can build server-side farm solutions that extend core SharePoint capabilities. SharePoint 2013 offers a new flexible development model—you can use SharePoint 2013 to create apps for SharePoint that take advantage of standard web technologies, such as JavaScript, OAuth, and OData. And SharePoint 2013 provides you with functionality to interact with SharePoint resources and a wide range of hosting options. The new app for SharePoint development model gives you the ability to build apps that take advantage of SharePoint capabilities and that run in the cloud instead of on your SharePoint farm. This flexible development model, along with the integration of standard web technologies, makes SharePoint development work more like other kinds of web development that you may already be doing.

### Cloud App Model

SharePoint 2013 introduces a Cloud App Model that enables you to create apps. Apps for SharePoint are self-contained pieces of functionality that extend the capabilities of a SharePoint website. An app may include SharePoint components such as lists, workflows, and site pages, but it can also surface a remote web application and remote data in SharePoint. An app has few or no dependencies on any other software on the device or platform where it is installed, other than what is built into the platform. This characteristic enables apps to be installed simply and uninstalled cleanly. Apps have no custom code that runs on the SharePoint servers. Instead, all custom logic moves "up" to the cloud or "down" to client computers. Additionally, SharePoint 2013 introduces an innovative delivery model for apps for SharePoint that includes components like the Office Store and the App Catalog.

### Programming Model

SharePoint 2013 makes it easy for any web developer, including those who work on non-Microsoft platform stacks, to create SharePoint solutions. What makes this possible is that SharePoint 2013 is based on common web standards like HTML, CSS, and JavaScript. Furthermore, implementation relies on established protocols like the Open Data protocol (OData), and OAuth.

### Development Tools

The new unified project system in Visual Studio lets you develop apps for SharePoint, apps for Office, apps for SharePoint that include apps for Office, or apps for Office that are hosted by SharePoint. In addition to the SharePoint project templates that were provided in earlier versions, Visual Studio 2012 now includes a new app project template in the Apps folder named Apps for SharePoint 2013. Several new properties have been added to the Properties window and Properties pages to support app for SharePoint projects. Other improvements include full support for development against the Cloud App Model, including OData and OAuth support, and full support for development against the Workflow Manager Client 1.0 platform.

### Core Platform Enhancements

SharePoint 2013 has been improved and enhanced to support the new cloud-based architecture and app-driven development framework. From the SharePoint APIs at the lowest level to connectivity to social media integration, SharePoint 2013 is designed and executed to support a rich application development experience. In addition to the use of Representational State Transfer (REST) endpoints for web services, there is a broad new API for both server and client development. Remote event receivers are now supported in addition to client-side rendering.

### Mobility

With SharePoint 2013, you can combine Windows Phone 7 applications with on-premises SharePoint services and applications, or with remote SharePoint services and applications that run in the cloud (such as those that use SharePoint Online), to create powerful applications that extend functionality beyond the traditional desktop or laptop and into a truly portable and much more accessible environment. The new mobility features in SharePoint 2013 are built on existing Microsoft tools and technologies, such as SharePoint, Windows Phone 7, Visual Studio, and Microsoft Silverlight. You can create SharePoint-powered mobile applications for Windows Phone using the new SharePoint phone application wizard template in Visual Studio, which lets you create simple list-based mobile applications. You can integrate new features introduced in SharePoint 2013, such as the Geolocation field type and "push" notifications from SharePoint Server, into your mobile applications.

### Social Collaboration

New and improved social and collaboration features make it easy for users to communicate and to stay engaged and informed. The improved My Site social feed helps users keep up to date with the people and content that they care about. The new Community Site feature provides a rich community experience that lets users easily find and share information and find people who have similar interests.

### Search

Search functionality in SharePoint 2013 includes several enhancements, custom content processing with the Content Enrichment web service, and a new framework for presenting search result types. Additionally, there have been significant enhancements made to the keyword query language (KQL).

### Workflows

Workflow Manager Client 1.0 is a redesigned workflow infrastructure that is built on Windows Workflow Foundation 4 and brings new power and flexibility to workflow authoring in SharePoint 2013. A fully declarative authoring environment enables information workers to use SharePoint Designer 2013 to author powerful workflows, and a new set of Visual Studio 2012 workflow project templates let developers access more sophisticated features like custom actions. Perhaps most importantly, Workflow Manager Client 1.0 is fully integrated with the model for apps for SharePoint. In addition, workflows execute in the cloud, not in SharePoint, which provides enormous flexibility in designing workflow-based apps for SharePoint.

### Enterprise Content Management

In SharePoint 2013, you can now use .NET client, Silverlight, Windows Phone, and JavaScript APIs, in addition to the newly expanded set of .NET server managed APIs, to customize Enterprise Content Management (ECM) experiences and behavior.

### Business Connectivity Services

Business Connectivity Services (BCS) enables SharePoint to access data from external data systems such as SAP, ERP, and CRM, in addition to other data-driven applications that are exposed through WCF services or OData endpoints. BCS in SharePoint 2013 has been improved and enhanced in many ways, including OData connectivity, external events, external data in apps, filtering and sorting, support for REST, and others.

### Application Services

SharePoint Server 2013 includes several services for working with data in your SharePoint sites. New for SharePoint is the Machine Translation Service, which translates sites, documents, and streams for multilingual support. SharePoint Server 2013 also includes Access Services and a new data access model. For converting files and streams to other formats, SharePoint Server 2013 has Word Automation Services and PowerPoint Automation Services (a new feature for SharePoint). SharePoint also provides data analysis tools, like PerformancePoint Services and Visio Services that enable business intelligence, and powerful new features in Excel Services.

# SharePoint Development

## Capabilities

SharePoint capabilities represent the features that you work with on SharePoint sites, such as workflows, social feeds, and search. By adding SharePoint capabilities to your app, you can help users to do things better and more quickly. You can add SharePoint capabilities to whatever kind of app you're developing: apps for SharePoint, solutions, sites, components, and mobile apps.

The following lists SharePoint capabilities that you can add to your apps:

* Workflows in SharePoint 2013- model and automate business processes.
* Social and collaboration features in SharePoint 2013- communicate and collaborate with people and track content.
* Integrating location and map functionality in SharePoint 2013- add location and mapping features.
* Search in SharePoint 2013- make search results more relevant.
* Business Connectivity Services in SharePoint 2013- access data from external data systems.
* Office 2013 and SharePoint 2013 application services- work with data, convert file formats, and translate sites.
* Work with data, convert file formats, and translate sites- work with user access, roles, rights, and claims.
* eDiscovery in SharePoint 2013- discover content in electronic format for legal purposes.

## Programming Models

You can develop applications for the SharePoint 2013 platform in many ways. These applications can be usefully categorized into the following groups based on the tools used to create them, the programming models used to develop them, the methods by which they are packaged and deployed, the methods by which they are marketed, and the devices on which they run.

### Apps

An app for SharePoint is similar to an app on a mobile device. It is a stand-alone productivity solution that does a small number of related tasks, installs easily, and uninstalls cleanly. Users can discover and download apps for SharePoint from a public SharePoint app store or from their organization's corporate app catalog. An app for SharePoint can include classic SharePoint components such as lists, custom website pages, Web Parts, workflows, and content types. But an app for SharePoint can also surface a remote web application and remote data in SharePoint. An app for SharePoint can also include both SharePoint and remote components. Apps for SharePoint are very safe applications whose custom logic is always shifted "up" to the cloud or "down" to the client computers. It never runs on the SharePoint servers.

### Publishing Sites

SharePoint publishing sites provide large-scale content publishing with a high degree of maintainability and regulation compliance. They also provide management of documents, records, taxonomy, and content types.

### Farm Solutions

SharePoint farm solutions are trusted SharePoint extensions whose custom logic calls the SharePoint server object model and runs with full trust on the SharePoint servers. These solutions are primarily for custom administrative extensions of SharePoint, such as timer jobs, custom Windows PowerShell commands, and extensions of Central Administration. Farm solutions are distributed as SharePoint solution packages that farm administrators upload to a farm-wide storage location from which they can be deployed. Components in farm solutions can have farm, web application, site collection, or website scope.

### Mobile Apps

Windows Phone apps, and apps built on non-Microsoft mobile platforms, can access SharePoint websites and data. Tools for building Windows Phone apps that interact with SharePoint 2013 are available for installation on Visual Studio 2010. A SharePoint client managed API just for use on Windows Phone devices is available. Mobile devices, including non-Microsoft devices, can also access SharePoint data through SharePoint REST/OData endpoints.

### Reusable Components

The SharePoint 2013 platform and Visual Studio 2012 enable encapsulation and reuse of application elements, including elements created with code, script, and XML markup.

## Client API’s

### Server Object Model (SSOM)

The largest set of APIs is in the server object model of managed classes. At the level of SharePoint Foundation 2013, this object model includes classes and members that enable programmatic control of the basic site and list structure of SharePoint Foundation. Most of these classes are in the [Microsoft.SharePoint](http://msdn.microsoft.com/en-us/library/office/microsoft.sharepoint(v=office.15).aspx) namespace. In addition, you can extend almost every SharePoint Foundation component by using the server object model, including workflows, alerts, Web Parts, basic search, and Microsoft Business Connectivity Services (BCS). The server object model also includes an extensive set of APIs that enable extensions of the administration and security system of SharePoint Foundation, including backup, farm health and diagnostics, logging, farm and web application management, upgrade, deployment, caching, and Windows PowerShell customization.

At the level of SharePoint Server 2013, many more classes are added to enable programming of Enterprise Content Management (ECM), user profiles, taxonomy, advanced search, and other features of SharePoint Server 2013.

You can use LINQ to Objects to query any IEnumerable collection in memory, but a LINQ to SharePoint provider enables direct querying of the lists in the SharePoint content databases. Strictly speaking, this provider is not available with any other set of APIs discussed in this topic; however, there are ways to use LINQ syntax in most of the others.

The assemblies that define the built-in server-side classes are installed to the global assembly cache of each server when SharePoint 2013 is installed. When you program against the server object model, your assemblies are installed as farm solutions to the global assembly cache.

### Client Object Models for Managed Code (CSOM)

#### .Net Client Object Model

The SharePoint 2013 object model for .NET Framework is used in .NET Framework applications that run on a non-phone Windows client. Any of the following counts as such a client: a user's computer, a server that is external to the SharePoint 2013 farm, and a web role or worker role on Microsoft Azure

Almost every class in the core site and list server object model has a corresponding class in the .NET Framework client object model. In addition, the .NET Framework client object model exposes a full set of APIs for extending other features, including some SharePoint Server 2013 features such as ECM, taxonomy, user profiles, advanced search, analytics, BCS, and others.

To improve performance, lines of code written against in the .NET Framework client object model are sent to the SharePoint server in batches where they are converted to server-side code and executed. The queried results, and the new state of all variables, are then returned to the client. You as the developer determine whether a batch runs synchronously or asynchronously.

You can use LINQ query syntax in your client code to query any IEnumerable object, including SharePoint 2013 objects that implement IEnumerable. However, when you do this, you are using LINQ to Objects, not the LINQ to SharePoint provider, so documentation of the latter is not relevant to client-side code.

The assemblies for the .NET Framework client object model must be installed on the client. They are included in a redistributable package that you can obtain on the SharePoint Client Components.

**Note**: You can also use the SharePoint REST/OData endpoints in a .NET Framework application. For a comparison of the .NET Framework client object model with the SharePoint REST/OData endpoints, see the section REST/OData endpoints later in this article.

#### Mobile Object Model

A special version of the Silverlight client object model is available for Windows Phone devices. It includes some additional APIs that are relevant only to telephones, such as APIs that enable a phone app to register for notifications from the Microsoft Push Notification Service. It supports all core SharePoint 2013 functionality; however, it does not provide support for any of the non-core extensibility areas that are supported by the other two client object models for managed code. To access those additional areas, use the SharePoint REST/OData endpoints in your mobile app.

*The assemblies for the mobile object model are persisted on every SharePoint 2013 server at %ProgramFiles%\Common Files\Microsoft Shared\web server extensions\15\TEMPLATE\LAYOUTS\ClientBin. You package them into the .xap file of your Windows Phone application.*

### JavaScript Object Model

SharePoint 2013 provides a JavaScript object model for use in either inline script or separate .js files. It includes all the same functionality as the .NET Framework and Silverlight client object models. Like the Silverlight client object model, the JavaScript object model is a useful way of including custom SharePoint code in an app, because custom server-side code is not allowed in apps for SharePoint. It also enables web developers to use their existing JavaScript skills to create SharePoint applications with a minimal learning curve.

Just like the managed-code client object models, the JavaScript infrastructure for SharePoint 2013 interacts with the farm servers in batches. These batches always run asynchronously. In addition, it is now possible to access SharePoint data across domains in JavaScript (but only data that is within the same parent site collection), which was not allowed in previous versions of SharePoint. Data is returned from the server in JavaScript Object Notation (JSON).

*The JavaScript object model is defined in a set of \*.js files located at %ProgramFiles%\Common Files\Microsoft Shared\web server extensions\15\TEMPLATE\LAYOUTS on each server.*

**Note**: You can also use the SharePoint REST/OData endpoints in a JavaScript application. For a comparison of the JavaScript client object model with SharePoint's REST/OData endpoints, see the following section, REST/OData endpoints.

### REST/OData Endpoints

For scenarios in which you need to access SharePoint entities from client technologies that do not use JavaScript and are not built on the .NET Framework or Silverlight platforms, SharePoint 2013 provides an implementation of a Representational State Transfer (REST) web service that uses the OData protocol to perform CRUD operations on SharePoint list data. In addition, almost every API in the client object models has a corresponding REST endpoint. This enables your code to interact directly with SharePoint artifacts by using any technology that supports standard HTTP requests and responses. To use the REST capabilities that are built into SharePoint 2013, your code constructs a RESTful HTTP request to an endpoint that corresponds to the desired client object model API. The client.svc web service handles the HTTP request and serves a response in either Atom or JSON format.

### WCF Data Services Framework

If you prefer to use LINQ syntax in .NET Framework or Silverlight client applications, SharePoint 2013 supports WCF Data Services as a LINQ provider. You can target either the listdata.svc (for list data only) as in earlier versions of SharePoint Foundation, or you can target the same client.svc that supports the OData interface for access to all SharePoint entities in addition to list data.

The following REST endpoints are available:

* Documents and content: <http://rgxshpntnew:9000/_api/lists>
* Team projects: <http://rgxshpntnew:9000/_api/web>
* Search: <http://rgxshpntnew:9000/_api/search>

### Deprecated API Sets

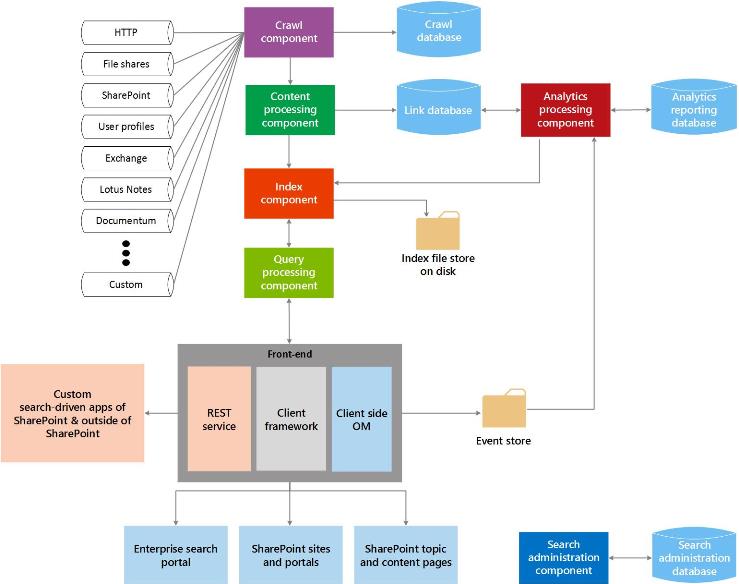
Two API sets are still supported in the SharePoint 2013 framework for backward compatibility, but we recommend that you not use them for new projects: the ASP.NET (asmx) web services, and direct Remote Procedure Calls (RPC) calls to the owssvr.dll file.

# Search

Search in SharePoint 2013 enables users to find relevant information more quickly and easily than ever before and makes it easy for Search administrators to customize the search experience. It also provides several API sets for more advanced customizations and solutions.

## Architecture

Search in SharePoint 2013 includes a wide variety of improvements and new features. With this version, Search in SharePoint 2013 is re-architected to a single enterprise search platform.



The search architecture consists of the following areas:

#### Crawl and Content Processing

The crawl and content processing architecture consists of the following:

* Crawl component- crawls content sources to collect crawled properties and metadata from crawled items and sends this information to the content processing component.
* Crawl database- contains information about crawled items, such as last crawl time, the last crawl ID, and the type of update during the last crawl.
* Content processing component- crawls content sources to collect crawled properties and metadata from crawled items and sends this information to the index component.

#### Index

The index component receives the processed items from the content processing component and writes them to the search index. This component also handles incoming queries, retrieves information from the search index, and sends back the result set to the query processing component.

#### Query Processing

The query processing component analyzes and processes search queries and results. The processed query is then submitted to the index component, which returns a set of search results for the query.

#### Search Administration

Search administration is composed of the search administration component and its corresponding database.

* Search administration component- runs the system processes for search, and adds and initializes new instances of search components.
* Search administration database- stores search configuration data.

#### Analytics

The analytics architecture consists of the analytics processing component, analytics reporting database, and link database.

* Analytics processing component- performs search analytics and usage analytics.
* Link database- stores information extracted by the content processing component and search click information.
* Analytics reporting database- stores the results of usage analytics.

## Extensibility Points

The Search in SharePoint 2013 architecture provides several extensibility points to support customization scenarios.

#### Connector Framework

The crawl component crawls content by invoking connectors or protocol handlers that interact with content sources to retrieve data. Search in SharePoint 2013 includes a connector framework that you can use to customize and build connectors to crawl new content sources.

#### Custom Content Processing

Within the content processing component, you can use the Content Enrichment web service callout to modify the managed properties of crawled items before they are added to the search index. This web service callout calls out to any external content enrichment web service that you create.

#### Query APIs

Search in SharePoint 2013 provides several query APIs, giving you lots of ways to access search results, so that you can return search results in a variety of custom solution types. The list below shows the APIs that you can use to program Search in SharePoint 2013 and where to find them.

|  |  |
| --- | --- |
| **API name** | **Class library or schema and path** |
| .NET CSOM | Microsoft.SharePoint.Client.Search.dll ProgramFiles\Common Files\Microsoft Shared\web server extensions\15\ISAPI |
| Silverlight CSOM | Microsoft.SharePoint.Client.Search.Silverlight.dll ProgramFiles\Common Files\Microsoft Shared\web server extensions\15\TEMPLATE\LAYOUTS\ClientBin |
| JavaScript CSOM | SP.search.js ProgramFiles\Common Files\Microsoft Shared\web server extensions\15\TEMPLATE\LAYOUTS |
| REST service | http://server/\_api/search/query http://server/\_api/search/suggest |
| Server object model | Microsoft.Office.Server.Search.dll ProgramFiles\Common Files\Microsoft Shared\web server extensions\15\ISAPI |

#### Analytics

To help identify and surface the content that users consider to be the most useful and relevant, the analytics processing component analyzes both the content itself, and also the way that users interact with it. These analyses are done by timer jobs that are responsible for performing analysis lifecycle tasks such as starting, stopping, pausing, and resuming an analysis job when requested. You can manipulate these timer jobs through the [Microsoft.Office.Server.Search.Analytics](http://msdn.microsoft.com/en-us/library/office/microsoft.office.server.search.analytics(v=office.15).aspx) namespace.

#### Custom Ranking Models

Search results can be ordered in various ways, one of which is by rank score. Rank scores are calculated by the search engine using ranking models. SharePoint Server 2013 provides fourteen ranking models by default. However, if you are not satisfied with the way your search results are ordered, you can use a custom ranking model.

#### Custom Security Trimming

Search in SharePoint Server 2013 performs security trimming of search results that are based on the identity of the user submitting the query, at query time, by using security information obtained from the crawl component. However, in some cases, you may need to implement custom security trimming. SharePoint Server 2013 provides two interfaces to accomplish this task: [ISecurityTrimmerPre](http://msdn.microsoft.com/en-us/library/office/microsoft.office.server.search.query.isecuritytrimmerpre(v=office.15).aspx) and [ISecurityTrimmerPost](http://msdn.microsoft.com/en-us/library/office/microsoft.office.server.search.query.isecuritytrimmerpost(v=office.15).aspx). The pre-trimmer interface (ISecurityTrimmerPre) carries out pre-query evaluation, where the search query is rewritten to add security information before the search query is matched to the search index. In contrast, the post-trimmer interface (ISecurityTrimmerPost) carries out post-query evaluation, where the search results are pruned before they are returned to the user.

#### Content Search Web Part

The Content Search Web Part is a Web Part that can display dynamic content that was previously crawled and added to the search index. Each instance of the Web Part is associated with a search query and shows the results for that particular search query. When users browse to a page that contains a Content Search Web Part, a search query is automatically issued, and the corresponding search results are returned from the search index. You can use the Content Search Web Part whenever you want to display content that is populated by automatically generated search queries. In some cases, you may want to extend the Content Search Web Part, which is exposed through the [Microsoft.Office.Server.Search.WebControls](http://msdn.microsoft.com/en-us/library/office/microsoft.office.server.search.webcontrols(v=office.15).aspx) namespace as [ContentBySearchWebPart](http://msdn.microsoft.com/en-us/library/office/microsoft.office.server.search.webcontrols.contentbysearchwebpart(v=office.15).aspx).

#### Search-driven mobile apps using the Navigation and Event Logging REST interfaces

SharePoint Server 2013 provides two new REST interfaces: Navigation and Event Logging. You can use them to create search-driven mobile apps for mobile devices, such as phones and tablets, that run on operating systems other than Windows. This feature lets you display the product catalog on a mobile device in an alternate way, instead of using a mobile channel.

## Making Content Available for Search

Search in SharePoint 2013 provides two approaches for processing queries to return search results—federated search and content crawling.

#### Federated Search

In this approach, search results are returned for content that is not crawled by your search server. The query is forwarded to an external content repository where it is processed by that repository's search engine. The repository's search engine then returns the results to the search server. The search server formats and renders the results from the external repository to display on the search results page. This approach offers the following advantages:

* You require no additional capacity requirements for the content index, as content is not crawled by Search in SharePoint 2013.
* You can take advantage of a repository's existing search engine. For example, you can federate to an Internet search engine to search the web.
* You can optimize the content repository's search engine for the repository's specific set of content, which might provide better search performance on the content set.
* You can access repositories that are secured against crawls, but which can be accessed by search queries.

#### Content Crawling

In this approach, results are returned from the Search service application's content index based on the user's query. The content index contains content that is crawled by the Search service application, and includes text content and metadata for each content item. This approach enables you to:

* Sort results by relevance.
* Control how frequently the content index is updated.
* Specify what metadata is crawled.
* Perform a single backup operation for crawled content.

#### Crawled and Managed Properties

A *crawled property* is content and metadata that is extracted from an item, such as a document or a URL, during a crawl. A crawled property can be an author, title, or subject. To include the content and metadata of crawled properties in the search index, you map crawled properties to managed properties. Managed properties can have a large number of settings, or attributes. These attributes determine how the contents are shown in search results. The search schema contains the attributes on managed properties and the mapping between crawled properties and managed properties.

#### Navigating to Managed Properties

1. Open SharePoint 2013 Central Administration
2. Select Application Management: Manage Service Applications
3. Select Search Service Application
4. Select Search Schema
5. Select Managed Properties

## Supported Search Syntax

SharePoint Server 2013 search supports Keyword Query Language (KQL) and FAST Query Language (FQL) search syntax for building search queries.

#### Keyword Query Language (KQL)

KQL is the default query language for building search queries. Using KQL, you specify the search terms or property restrictions that are passed to the SharePoint search service.

#### FAST Query Language (FQL)

FQL is a structured query language that supports advanced query operators. You can use FQL when you want to create complex queries that you want to pass programmatically to the SharePoint search service. FQL isn’t intended to be exposed to end users, and is disabled by default.

To enable FQL, you have to copy the default result source and modify the Query Transformation string {?{searchTerms} -ContentClass=urn:content-class:SPSPeople}, at one of these levels -- Search Service Application (SSA), Site Collection, or Site -- and in one of the following ways:

Remove the KQL filter, -ContentClass:urn:content-class:SPSPeople, from the Query Transformation. The resulting Query Transformation string will be: {?{searchTerms}}

Replace the Query Transformation string with an FQL equivalent, such as {?andnot({searchTerms},filter(contentclass:"urn:content-class:SPSPeople\*"))}.

## Search Query with SharePoint Search Web Service (Deprecated)

#### Search Query

The search query uses a <QueryPacket> xml document for the request. It looks like this:

<?xml version=\1.0" encoding="utf-8" ?>

<QueryPacket xmlns=”urn:Microsoft.Search.Query” revision=”1000”>

<Query domain=”ODomain”>

<QueryId />

<Context>

<QueryText language=”en-US” type=”STRING”>KQL</QueryText>

</Context>

<ResultProvider>Default</ResultProvider>

<Range><StartAt></StartAt><Count></Count></Range>

<Properties>

<Property name=”” />

</Properties>

<SortByProperties><SortByProperty /></SortByProperties>

<ImplicitAndBehavior />

<RelevanceModel />

<EnableStemming />

<TrimDuplicates />

<IncludeSpecialTermResults />

<PreQuerySuggestions />

<HighlightQuerySuggestions />

<CapitalizeFirstLetters />

<ResubmitFlags><ResubmitFlag /></ResubmitFlags>

<EnableSpellcheck />

<UserContext><UserContextData /></UserContext>

<FindSimilar><SimilarTo /><SimilarType /><SortSimilar /></FindSimilar>

<IncludeRefinementResults>

<Refiners> <Refiner /> </Refiners>

<MaxShallowRefinementHits />

</IncludeRefinementResults>

<RefinementFilters><RefinementFilter /></RefinementFilters>

<IgnoreAllNoiseQuery />

<IncludeRelevantResults />

<IncludeHighConfidenceResults />

</Query>

</QueryPacket>

<QueryText type=”type”> where type is:

* STRING- Indicates that the kind of query is keyword query (KQL).
* MSSQLFT- Indicates that the type of query is SQL Full Text Syntax query.
* FQL- Indicates that the type of query is FAST Query Language (FQL).

<ResultProvider> where provider is:

* Default- The default result provider that is configured by using the administrative interfaces.
* SharePointSearch- Specifies the SharePoint Server search provider.
* FASTSearch- Specifies the FAST Search Server for SharePoint provider.

<QueryText> uses KQL.

#### List of available content class items:

* STS\_Site –  Site Collection
* STS\_Web  –  Site (Web)
* STS\_List\_850  –  Page Library
* STS\_ListItem\_850  –  Page
* STS\_List\_DocumentLibrary  –  Document Library
* STS\_ListItem\_DocumentLibrary  –  Document Library Items
* STS\_List  –  Custom List
* STS\_ListItem  –  Custom List Item
* STS\_List\_Links  –  Links List
* STS\_ListItem\_Links  –  Links List Item
* STS\_List\_Tasks  –  Tasks List
* STS\_ListItem\_Tasks  –  Tasks List Item
* STS\_List\_Events  –  Events List
* STS\_ListItem\_Events  –  Events List Item
* STS\_List\_Announcements  –  Announcements List
* STS\_ListItem\_Announcements  –  Announcements List Item
* STS\_List\_Contacts  –  Contacts List
* STS\_ListItem\_Contacts  –  Contacts List Item
* STS\_List\_DiscussionBoard  –  Discussion List
* STS\_ListItem\_DiscussionBoard  –  Discussion List Item
* STS\_List\_IssueTracking  –  Issue Tracking List
* STS\_ListItem\_IssueTracking  –  Issue Tracking List Item
* STS\_List\_GanttTasks  –  Project Tasks List
* STS\_ListItem\_GanttTasks  –  Project Tasks List Item
* STS\_List\_Survey  –  Survey List
* STS\_ListItem\_Survey  –  Survey List Item
* STS\_List\_PictureLibrary  –  Picture Library
* STS\_ListItem\_PictureLibrary  –  Picture Library Item
* STS\_List\_WebPageLibrary  –  Web Page Library
* STS\_ListItem\_WebPageLibrary  –  Web Page Library Item
* STS\_List\_XMLForm  –  Form Library
* STS\_ListItem\_XMLForm  –  Form Library Item
* urn:content-class:SPSSearchQuery  –  Search Query
* urn:content-class:SPSListing:News  –  News Listing
* urn:content-class:SPSPeople  –  People
* urn:content-classes:SPSCategory  –  Category
* urn:content-classes:SPSListing  –  Listing
* urn:content-classes:SPSPersonListing  –  Person Listing
* urn:content-classes:SPSTextListing  –  Text Listing
* urn:content-classes:SPSSiteListing  –  Site Listing
* urn:content-classes:SPSSiteRegistry  –  Site Registry Listing

# Appendix

## Keyword Query Language (KQL) Syntax Reference

### Elements of a KQL Query

A KQL query consists of one or more of the following elements:

* Free text-keywords—words or phrases
* Property restrictions

You can combine KQL query elements with one or more of the available operators. If the KQL query contains only operators or is empty, it isn't valid. KQL queries are case-insensitive but the operators are case-sensitive (uppercase).

### Constructing Free-Text Queries using KQL

When you construct your KQL query by using free-text expressions, Search in SharePoint 2013 matches results for the terms you chose for the query based on terms stored in the full-text index. This includes managed property values where [FullTextQueriable](http://msdn.microsoft.com/en-us/library/office/microsoft.office.server.search.administration.managedproperty.fulltextqueriable(v=office.15).aspx) is set to true. Free text KQL queries are case-insensitive but the operators must be in uppercase. You can construct KQL queries by using one or more of the following as free-text expressions:

* A word (includes one or more characters without spaces or punctuation)
* A phrase (includes two or more words together, separated by spaces; however, the words must be enclosed in double quotation marks)

To construct complex queries, you can combine multiple free-text expressions with KQL query operators. If there are multiple free-text expressions without any operators in between them, the query behavior is the same as using the AND operator.

#### Using words in the free-text KQL query

When you use words in a free-text KQL query, Search in SharePoint 2013 returns results based on exact matches of your words with the terms stored in the full-text index. You can use just a part of a word, from the beginning of the word, by using the wildcard operator (\*) to enable prefix matching. In prefix matching, Search in SharePoint 2013 matches results with terms that contain the word followed by zero or more characters.

For example, the following KQL queries return content items that contain the terms "federated" and "search":

federated search   
federat\* search   
search fed\*

KQL queries don’t support suffix matching.

#### Using phrases in the free-text KQL query

When you use phrases in a free-text KQL query, Search in SharePoint 2013 returns only the items in which the words in your phrase are located next to each other. To specify a phrase in a KQL query, you must use double quotation marks. KQL queries don’t support suffix matching, so you can’t use the wildcard operator before a phrase in free-text queries. However, you can use the wildcard operator after a phrase.

### Property Restriction Queries in KQL

Using KQL, you can construct queries that use property restrictions to narrow the focus of the query to match only results based on a specified condition.

#### Specifying property restrictions

A basic property restriction consists of the following:

<Property Name><Property Operator><Property Value>

The property restriction must **not** include white space between the property name, property operator, and the property value, or the property restriction is treated as a free-text query. The length of a property restriction is limited to 2,048 characters.

#### Specifying property names for property restrictions

You must specify a valid managed property name for the property restriction. By default, Search in SharePoint 2013 includes several managed properties for documents. To specify a property restriction for a crawled property value, you must first map the crawled property to a managed property. The managed property must be [Queryable](http://msdn.microsoft.com/en-us/library/office/microsoft.office.server.search.administration.managedproperty.queryable(v=office.15).aspx) so that you can search for that managed property in a document. In addition, the managed property may be Retrievable for the managed property to be retrieved. However, the managed property doesn’t have to be Retrievable to carry out property searches.

#### Property operators that are supported in property restrictions

Search in SharePoint 2013 supports several property operators for property restrictions, as shown below.

|  |
| --- |
| : Returns results where the value specified in the property restriction is equal to the property value that is stored in the Property Store database, or matches individual terms in the property value that is stored in the full-text index. |
| = Returns search results where the property value is equal to the value specified in the property restriction. |
| < Returns results where the property value is less than the value specified in the property restriction. |
| > Returns search results where the property value is greater than the value specified in the property restriction. |
| <= Returns search results where the property value is less than or equal to the value specified in the property restriction. |
| >= Returns search results where the property value is greater than or equal to the value specified in the property restriction. |
| <> Returns search results where the property value does not equal the value specified in the property restriction. |
| .. Returns search results where the property value falls within the range specified in the property restriction. For example, the range A..B represents a set of values from A to B where both A and B are inclusive. For date ranges this means from the beginning of day A to the end of day B. |

#### Property values in the full-text index

Property values are stored in the full-text index when the FullTextQueriable property is set to true for a managed property. You can configure this only for string properties. Property values that are specified in the query are matched against individual terms that are stored in the full-text index. Use the [NoWordBreaker](http://msdn.microsoft.com/en-us/library/office/microsoft.office.server.search.administration.managedproperty.nowordbreaker(v=office.15).aspx) property to specify whether to match with the whole property value. Prefix matching is also supported. You can use the wildcard operator (\*), but isn’t required when you specify individual words. When you specify a phrase for the property value, matched results must contain the specified phrase within the property value that is stored in the full-text index. Prefix matching is also supported with phrases specified in property values, but you must use the wildcard operator (\*) in the query, and it is supported only at the end of the phrase.

#### Numerical values for properties

For numerical property values, which include the Integer, Double, and Decimal managed types, the property restriction is matched against the entire value of the property.

#### Date or time values for properties

KQL provides the datetime data type for date and time.The following ISO 8601-compatible datetime formats are supported in queries: YYYY-MM-DD, YYYY-MM-DDThh:mm:ss, YYYY-MM-DDThh:mm:ssZ, YYYY-MM-DDThh:mm:ssfrZ.

### KQL Operators for Complex Queries

KQL syntax includes several operators that you can use to construct complex queries.

#### Boolean operators

You use Boolean operators to broaden or narrow your search. You can use Boolean operators with free text expressions and property restrictions in KQL queries. Boolean operators include AND, NOT, and OR.

#### Proximity operators

You use proximity operators to match the results where the specified search terms are within close proximity to each other. Proximity operators can be used with free-text expressions only; they are not supported with property restrictions in KQL queries. There are two proximity operators: NEAR and ONEAR.

The NEAR operator matches the results where the specified search terms are within close proximity to each other, without preserving the order of the terms. The syntax for NEAR is as follows:

<expression> NEAR(n=4) <expression>

Where n is an optional parameter that indicates maximum distance between the terms. The value of n is an integer >= 0 with a default of **8**. The parameter n can be specified as n=v where v represents the value, or shortened to only v; such as NEAR(4) where v is 4.

The ONEAR operator matches the results where the specified search terms are within close proximity to each other, while preserving the order of the terms. The syntax for ONEAR is as follows, where n is an optional parameter that indicates maximum distance between the terms. The value of n is an integer >= 0 with a default of **8**.

<expression> ONEAR(n=4) <expression>

The parameter n can be specified as n=v where v represents the value, or shortened to only v; such as ONEAR(4) where v is 4.

#### Synonym operators

You use the WORDS operator to specify that the terms in the query are synonyms, and that results returned should match either of the specified terms. You can use the WORDS operator with free text expressions only; it is not supported with property restrictions in KQL queries. The following query example matches results that contain either the term "TV" or the term "television": WORDS(TV, Television)

#### Wildcard operator

You use the wildcard operator—the asterisk character ("**\***")—to enable prefix matching. You can specify part of a word, from the beginning of the word, followed by the wildcard operator, in your query, as follows. This query would match results that include terms beginning with "serv", followed by zero or more characters, such as serve, server, service, and so on:

serv\*

#### Inclusion and exclusion operators

You can specify whether the results that are returned should include or exclude content that matches the value specified in the free text expression or the property restriction by using the inclusion and exclusion operators. Operators for including and excluding content in results: Inclusion “+” and Exclusion “-“.

#### Dynamic ranking operator

You use the XRANK operator to boost the dynamic rank of items based on certain term occurrences within the match expression, without changing which items match the query. An XRANK expression contains one component that must be matched, the match expression, and one or more components that contribute only to dynamic ranking, the rank expression. At least **one** of the parameters, excluding n, must be specified for an XRANK expression to be valid. Match expressions may be any valid KQL expression, including nested XRANK expressions. Rank expressions may be any valid KQL expression without XRANK expressions. If your KQL queries have multiple XRANK operators, the final dynamic rank value is calculated as a sum of boosts across all XRANK operators.

## SharePoint Search Web Service

#### Search Query

<?xml version=\1.0" encoding="utf-8" ?>

<QueryPacket Revision="1000">

<Query>

<Context>

<QueryText language="en-US" type=”STRING">88888888\*</QueryText>

</Context>";

<ResultProvider>Default</ResultProvider>

<Range><StartAt>1</StartAt><Count>20</Count></Range>

<Properties>

<Property name="contentclass" />

<Property name="IsDocument" />

<Property name="DAV:displayname" />

<Property name="DAV:href" />

<Property name="BOLNumOWSTEXT" />

<Property name="BatchDateOWSDATE" />

<Property name="Created" />

</Properties>

</Query>

</QueryPacket>

#### Result for <http://rgxvmsp:9000/_vti_bin/search.asmx>

Search Metadata -------------------------

<SearchMetadata>

<Properties>

<Name>BatchDate</Name>

<Description />

<Type>System.DateTime</Type>

<Retrievable>true</Retrievable>

<FullTextQueriable>true</FullTextQueriable>

</Properties>

<Properties>

<Name>BOLNum</Name>

<Description>Tsort BOL Images BOLNum property</Description>

<Type>System.String</Type>

<Retrievable>true</Retrievable>

<FullTextQueriable>true</FullTextQueriable>

</Properties>

</SearchMetadata>

QueryEx -------------------------

<Results>

<RelevantResults>

<contentclass>STS\_Web</contentclass>

<IsDocument>false</IsDocument>

<DAV\_x003A\_displayname>BOL</DAV\_x003A\_displayname>

<DAV\_x003A\_href>http://rgxvmsp:9000/TSortImaging/BOL</DAV\_x003A\_href>

<Created>2006-06-14T15:18:11-04:00</Created>

</RelevantResults>

<RelevantResults>

<contentclass>STS\_Document</contentclass>

<IsDocument>false</IsDocument>

<DAV\_x003A\_displayname>default.aspx</DAV\_x003A\_displayname>

<DAV\_x003A\_href>http://rgxvmsp:9000/TSortImaging/BOL/default.aspx</DAV\_x003A\_href>

</RelevantResults>

<RelevantResults>

<contentclass>STS\_List\_DocumentLibrary</contentclass>

<IsDocument>false</IsDocument>

<DAV\_x003A\_displayname>AllItems.aspx</DAV\_x003A\_displayname>

<DAV\_x003A\_href>http://rgxvmsp:9000/TSortImaging/BOL/Document Library/Forms/AllItems.aspx</DAV\_x003A\_href>

</RelevantResults>

<RelevantResults>

<contentclass>STS\_ListItem\_DocumentLibrary</contentclass>

<IsDocument>true</IsDocument>

<DAV\_x003A\_displayname>test8.tiff</DAV\_x003A\_displayname>

<DAV\_x003A\_href>http://rgxvmsp:9000/TSortImaging/BOL/Document Library/test8.tiff</DAV\_x003A\_href>

<BOLNum>888888888</BOLNum>

<BatchDate>2012-02-21T05:00:00-05:00</BatchDate>

<Created>2012-02-28T19:19:10-05:00</Created>

</RelevantResults>

</Results>

#### Result for http://rgxshpntnew:9000/\_vti\_bin/search.asmx

Search Metadata -------------------------

<SearchMetadata>

<Properties>

<Name>BatchDateOWSDATE</Name>

<Description />

<Type>System.String</Type>

<Retrievable>true</Retrievable>

<FullTextQueriable>false</FullTextQueriable>

</Properties>

<Properties>

<Name>BOLNumOWSTEXT</Name>

<Description />

<Type>System.String</Type>

<Retrievable>true</Retrievable>

<FullTextQueriable>false</FullTextQueriable>

</Properties>

</SearchMetadata>

QueryEx -------------------------

<Results>

<RelevantResults>

<contentclass>STS\_Web</contentclass>

<IsDocument>false</IsDocument>

<DAV\_x003A\_displayname>BOL</DAV\_x003A\_displayname>

<DAV\_x003A\_href>http://rgxshpntnew:9000/tsortimaging/bol/</DAV\_x003A\_href>

<Created>2006-06-14T15:18:10-04:00</Created>

</RelevantResults>

<RelevantResults>

<contentclass>STS\_ListItem\_DocumentLibrary</contentclass>

<IsDocument>false</IsDocument>

<DAV\_x003A\_displayname>DispForm.aspx</DAV\_x003A\_displayname>

<DAV\_x003A\_href>http://rgxshpntnew:9000/tsortimaging/bol/document library/forms/dispform.aspx?id=77220/</DAV\_x003A\_href>

<BOLNumOWSTEXT>888888888</BOLNumOWSTEXT>

<BatchDateOWSDATE>2012-02-21T05:00:00Z</BatchDateOWSDATE>

<Created>2012-02-28T19:19:10-05:00</Created>

</RelevantResults>

</Results>