
Amazon Kendra

Developer Guide



Amazon Kendra: Developer Guide

Copyright © Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

.....	x
What is Amazon Kendra?	1
Benefits of Amazon Kendra	1
Amazon Kendra Developer Edition	1
Amazon Kendra Enterprise Edition	2
Pricing for Amazon Kendra	2
Are you a first-time Amazon Kendra user?	2
How Amazon Kendra works	3
Index	3
Index fields	4
Searching indexes	5
Documents	5
Types of documents	5
Document attributes	6
Data sources	7
Queries	7
Tags	8
Tagging resources	8
Tag restrictions	8
Setting up Amazon Kendra	10
Sign up for AWS	10
Regions and endpoints	10
Setting up the AWS CLI	10
Setting up the AWS SDKs	11
IAM access roles for Amazon Kendra	12
IAM roles for indexes	12
IAM Roles for the BatchPutDocument operation	14
IAM roles for data sources	15
IAM roles for Amazon S3 data sources	15
IAM role for Confluence server data sources	16
IAM roles for database data sources	17
IAM roles for Google Drive data sources	19
IAM roles for OneDrive data sources	20
IAM role for Salesforce data sources	21
IAM role for ServiceNow data sources	22
IAM roles for SharePoint data sources	23
Virtual private cloud (VPC) IAM role	24
IAM roles for web crawler data sources	25
IAM roles for Amazon WorkDocs data sources	26
IAM roles for frequently asked questions	28
IAM roles for query suggestions	29
IAM roles for principal mapping of users and groups	30
IAM roles for AWS Single Sign-On	31
Deploying Amazon Kendra	32
Overview	32
Prerequisites	32
Setting up the example	33
Main search page	33
Search component	33
Results component	33
Facets component	33
Pagination component	34
Adjusting capacity	35
Viewing capacity	35

Adding and removing capacity	35
Query suggestions capacity	36
Adaptive query bursting	36
Getting started	37
Prerequisites	37
Prerequisites for the AWS CLI and SDK	38
Getting started with an S3 bucket (console)	40
Getting started (AWS CLI)	41
Getting started (SDK for Python (Boto3))	42
Getting started (SDK for Java)	44
Getting started with Confluence (console)	47
Getting started with Google Drive (console)	48
Getting started with OneDrive (console)	50
Getting started with S3 (console)	52
Getting started with MySQL (console)	53
Getting started with Salesforce (console)	54
Getting started with ServiceNow (console)	55
Getting started with SharePoint (console)	56
Getting started with Amazon Kendra Web Crawler (console)	58
Getting started with Amazon WorkDocs (console)	60
Getting started with an AWS SSO identity source (console)	61
Changing your AWS SSO identity source	62
Creating an index	64
Controlling access to documents in an index	66
Using OpenID	67
Using a JSON Web Token (JWT) with a shared secret	68
Using a JSON Web Token (JWT) with a public key	71
Using JSON	73
Adding documents directly to an index	74
Adding documents with the API	75
Adding documents from an Amazon S3 bucket	76
Adding documents in languages other than English	78
Adding questions and answers directly to an index	80
Basic .csv file	81
Custom .csv file	81
JSON file	82
Using your FAQ file	84
Adding documents from a data source	85
Setting an update schedule	85
Setting a language	86
Using an Amazon S3 data source	86
Using a Confluence data source	90
Using a custom data source	94
Using a database data source	100
Using a Google Drive data source	102
Using a OneDrive data source	104
Using a Salesforce data source	106
Using a ServiceNow data source	109
Using a SharePoint data source	114
Using a web crawler data source	116
Using an Amazon WorkDocs data source	119
Deleting data sources	119
Creating custom document attributes	121
Adding custom attributes with the BatchPutDocument operation	122
Adding custom attributes to an Amazon S3 data source	122
Mapping data source fields	123
Configuring Amazon Kendra to use a VPC	125

Connecting to a database in a VPC	126
Deleting an index	128
Searching indexes	129
Querying an index	129
Prerequisites	130
Searching an index (console)	130
Searching an index (SDK)	130
Searching in languages	132
Filtering queries	134
Facets	134
Using document attributes to filter search results	135
Filtering a document's attributes	136
Filtering on user context	136
Filtering by user token	137
Filtering by user ID	137
Filtering by user attribute	138
User context filtering for documents added directly to an index	139
User context filtering for frequently asked questions	139
User context filtering for database data sources	140
User context filtering for Confluence data sources	140
User context filtering for Google Drive data sources	140
User context filtering for Microsoft OneDrive data sources	141
User context filtering for Amazon S3 data sources	142
User context filtering for Salesforce data sources	142
User context filtering for ServiceNow data sources	142
User context filtering for Microsoft SharePoint data sources	143
User context filtering for Amazon WorkDocs data sources	143
Query responses	143
Query suggestions	145
Tuning responses	145
Sorting responses	146
Response types	147
Answer	148
Document	149
Question and answer	149
Tuning search relevance	151
Relevance tuning at the index level	152
Relevance tuning at the query level	152
Suggesting popular search queries	154
Query suggestions settings	154
Block certain queries from suggestions	158
Clear suggestions	162
No suggestions available	162
Submitting feedback for incremental learning	163
Using the Amazon Kendra JavaScript library to submit feedback	164
Step 1: Insert a script tag into your Amazon Kendra search application	164
Step 2: Add the feedback token to search results	165
Step 3: Test the feedback script	166
Using the Amazon Kendra API to submit feedback	166
Adding synonyms to an index	168
Creating a thesaurus file	169
Adding a thesaurus to an index	170
Updating a thesaurus	173
Deleting a thesaurus	176
Toggling highlights in search results	177
Tutorial: Building an intelligent search solution	178
Prerequisites	179

Step 1: Adding documents	179
Downloading the sample dataset	180
Creating an Amazon S3 bucket	181
Creating data and metadata folders in your S3 bucket	183
Uploading the input data	185
Step 2: Detecting entities	186
Running an Amazon Comprehend entities analysis job	186
Step 3: Formatting the metadata	192
Downloading and extracting the Amazon Comprehend output	193
Uploading the output into the S3 bucket	195
Converting the output to Amazon Kendra metadata format	196
Cleaning up your Amazon S3 bucket	199
Step 4: Creating an index and ingesting the metadata	201
Creating an Amazon Kendra index	201
Updating the IAM role for Amazon S3 access	207
Creating Amazon Kendra custom search index fields	209
Adding the Amazon S3 bucket as a data source for the index	213
Syncing the Amazon Kendra index	215
Step 5: Querying the index	217
Querying your Amazon Kendra index	218
Filtering your search results	222
Step 6: Cleaning up	224
Cleaning up your files	224
.....	225
Monitoring and Logging	226
Monitoring indexes	226
Monitoring Amazon Kendra API calls with CloudTrail	229
Amazon Kendra Information in CloudTrail	230
Example: Amazon Kendra log file Entries	230
Monitoring Amazon Kendra with CloudWatch	231
Viewing Amazon Kendra metrics	231
Creating an alarm	231
CloudWatch Metrics for index synchronization Jobs	232
Metrics for Amazon Kendra data sources	233
Metrics for indexed documents	235
Monitoring Amazon Kendra with CloudWatch Logs	235
Data source log streams	236
Document log streams	237
Security	238
Data protection	238
Encryption at rest	239
Encryption in transit	239
Key management	239
VPC endpoints (AWS PrivateLink)	240
Considerations for Amazon Kendra VPC endpoints	240
Creating an interface VPC endpoint for Amazon Kendra	240
Creating a VPC endpoint policy for Amazon Kendra	241
Identity and access management	241
Audience	242
Authenticating with identities	242
Managing access using policies	244
How Amazon Kendra works with IAM	246
Identity-based policy examples	249
AWS managed policies	253
Troubleshooting	256
Logging and monitoring in Amazon Kendra	257
Compliance validation	258

Resilience	258
Infrastructure security	258
Quotas	260
Supported regions	260
Quotas	260
Troubleshooting	263
Troubleshooting data sources	263
My documents were not indexed	263
My synchronization job failed	263
My synchronization job is incomplete	264
My synchronization job succeeded but there are no indexed documents	264
Troubleshooting document search results	264
Why do I only see 100 results?	264
Why are documents I expect to see missing?	264
Why do I see documents that have an ACL policy?	265
Troubleshooting general issues	265
Document history	266
API Reference	269
Actions	269
BatchDeleteDocument	271
BatchGetDocumentStatus	274
BatchPutDocument	277
ClearQuerySuggestions	281
CreateDataSource	283
CreateFaq	292
CreateIndex	296
CreateQuerySuggestionsBlockList	301
CreateThesaurus	305
DeleteDataSource	309
DeleteFaq	311
DeleteIndex	313
DeletePrincipalMapping	315
DeleteQuerySuggestionsBlockList	318
DeleteThesaurus	320
DescribeDataSource	322
DescribeFaq	331
DescribeIndex	335
DescribePrincipalMapping	340
DescribeQuerySuggestionsBlockList	343
DescribeQuerySuggestionsConfig	347
DescribeThesaurus	350
GetQuerySuggestions	354
ListDataSources	357
ListDataSourceSyncJobs	360
ListFaqs	364
ListGroupsOlderThanOrderingId	367
ListIndices	370
ListQuerySuggestionsBlockLists	372
ListTagsForResource	375
ListThesauri	377
PutPrincipalMapping	380
Query	384
StartDataSourceSyncJob	392
StopDataSourceSyncJob	394
SubmitFeedback	396
TagResource	399
UntagResource	401

UpdateDataSource	403
UpdateIndex	411
UpdateQuerySuggestionsBlockList	415
UpdateQuerySuggestionsConfig	418
UpdateThesaurus	421
Data Types	423
AccessControlListConfiguration	426
AclConfiguration	427
AdditionalResultAttribute	428
AdditionalResultAttributeValue	429
AttributeFilter	430
AuthenticationConfiguration	432
BasicAuthenticationConfiguration	433
BatchDeleteDocumentResponseFailedDocument	434
BatchGetDocumentStatusResponseError	435
BatchPutDocumentResponseFailedDocument	436
CapacityUnitsConfiguration	437
ClickFeedback	438
ColumnConfiguration	439
ConfluenceAttachmentConfiguration	441
ConfluenceAttachmentToIndexFieldMapping	442
ConfluenceBlogConfiguration	443
ConfluenceBlogToIndexFieldMapping	444
ConfluenceConfiguration	445
ConfluencePageConfiguration	448
ConfluencePageToIndexFieldMapping	449
ConfluenceSpaceConfiguration	450
ConfluenceSpaceToIndexFieldMapping	452
ConnectionConfiguration	453
DatabaseConfiguration	455
DataSourceConfiguration	457
DataSourceGroup	459
DataSourceSummary	460
DataSourceSyncJob	462
DataSourceSyncJobMetrics	464
DataSourceSyncJobMetricTarget	466
DataSourceToIndexFieldMapping	467
DataSourceVpcConfiguration	468
Document	469
DocumentAttribute	471
DocumentAttributeValue	472
DocumentAttributeValueCountPair	473
DocumentInfo	474
DocumentMetadataConfiguration	475
DocumentRelevanceConfiguration	476
DocumentsMetadataConfiguration	477
Facet	478
FacetResult	479
FaqStatistics	480
FaqSummary	481
GoogleDriveConfiguration	483
GroupMembers	485
GroupOrderingIdSummary	486
GroupSummary	488
HierarchicalPrincipal	489
Highlight	490
IndexConfigurationSummary	491

IndexStatistics	493
JsonTokenTypeConfiguration	494
JwtTokenTypeConfiguration	495
MemberGroup	497
MemberUser	498
OneDriveConfiguration	499
OneDriveUsers	501
Principal	502
ProxyConfiguration	503
QueryResultItem	504
QuerySuggestionsBlockListSummary	506
Relevance	508
RelevanceFeedback	510
S3DataSourceConfiguration	511
S3Path	513
SalesforceChatterFeedConfiguration	514
SalesforceConfiguration	516
SalesforceCustomKnowledgeArticleTypeConfiguration	519
SalesforceKnowledgeArticleConfiguration	521
SalesforceStandardKnowledgeArticleTypeConfiguration	522
SalesforceStandardObjectAttachmentConfiguration	523
SalesforceStandardObjectConfiguration	524
ScoreAttributes	526
Search	527
SeedUrlConfiguration	528
ServerSideEncryptionConfiguration	529
ServiceNowConfiguration	530
ServiceNowKnowledgeArticleConfiguration	532
ServiceNowServiceCatalogConfiguration	534
SharePointConfiguration	536
SiteMapsConfiguration	539
SortingConfiguration	540
SqlConfiguration	542
Status	543
Suggestion	544
SuggestionHighlight	545
SuggestionTextWithHighlights	546
SuggestionValue	547
Tag	548
TextDocumentStatistics	549
TextWithHighlights	550
ThesaurusSummary	551
TimeRange	553
Urls	554
UserContext	555
UserGroupResolutionConfiguration	557
UserTokenConfiguration	558
WebCrawlerConfiguration	559
WorkDocsConfiguration	562
Common Errors	563
Common Parameters	565
AWS glossary	568

What is Amazon Kendra?

Amazon Kendra is a highly accurate intelligent search service that enables your users to search unstructured data using natural language. It returns specific answers to questions, giving users an experience that's close to interacting with a human expert. It is highly scalable and capable of meeting performance demands, tightly integrated with other AWS services such as [Amazon S3](#) and [Amazon Lex](#), and offers enterprise-grade security.

Amazon Kendra users can ask the following types of questions, or queries:

- **Factoid questions** — Simple who, what, when, or where questions, such as *Who is on duty today?* or *Where is the nearest service center to me?* Factoid questions have fact-based answers that can be returned in the form of a single word or phrase. The precise answer, however, must be explicitly stated in the ingested text content.
- **Descriptive questions** — Questions whose answer could be a sentence, passage, or an entire document. For example, *How do I connect my Echo Plus to my network?* or *How do I get tax benefits for lower income families?*.
- **Keyword searches** — Questions where the intent and scope are not clear. For example, *keynote address*. As 'address' can often have several meanings, Amazon Kendra can infer the user's intent behind the search query to return relevant information aligned with the user's intended meaning. Amazon Kendra uses deep learning models to handle this kind of query.

Benefits of Amazon Kendra

Amazon Kendra has the following benefits:

- **Accuracy** — Unlike traditional search services that use keyword searches where results are based on basic keyword matching and ranking, Amazon Kendra attempts to understand the content, the user context, and the question. Amazon Kendra searches across your data and goes beyond traditional search to return the most relevant word, snippet, or document for your query. Amazon Kendra uses machine learning to improve search results over time.
- **Simplicity** — Amazon Kendra provides a console and API for managing the documents that you want to search. You can use a simple search API to integrate Amazon Kendra into your client applications, such as websites or mobile applications.
- **Connectivity** — Amazon Kendra can connect to third-party data sources to provide search across documents managed in different environments.
- **User Access Control** — Amazon Kendra delivers highly secure enterprise search for your search applications. Your search results reflect the security model of your organization. Customers are responsible for authenticating and authorizing users to gain access to their search application.

Amazon Kendra Developer Edition

The Amazon Kendra Developer Edition provides all of the features of Amazon Kendra at a lower cost. It includes a free tier that provides 750 hours of use. The Developer Edition is ideal to explore how Amazon Kendra indexes your documents, to try out features, and to develop applications that use Amazon Kendra.

The developer edition provides the following:

- Up to 5 indexes with up to 5 data sources each.
- 10,000 documents or 3 GB of extracted text.
- Approximately 4,000 queries per day or 0.05 queries per second.
- Runs in 1 availability zone (AZ) – see [Availability Zones](#) (data centers in AWS regions)

You should not use the Developer Edition for a production application. The Developer Edition doesn't provide any guarantees of latency or availability.

Amazon Kendra Enterprise Edition

Use Amazon Kendra Enterprise Edition when you want to index your entire enterprise document library or for when your application is ready for use in a production environment.

The enterprise edition provides the following:

- Up to 5 indexes with up to 50 data sources each.
- 100,000 documents or 30 GB of extracted text.
- Approximately 8,000 queries per day or 0.1 queries per second.
- Runs in 3 availability zones (AZ) – see [Availability Zones](#) (data centers in AWS regions)

You can increase this quota using the [Service Quotas console](#).

Pricing for Amazon Kendra

You can get started for free with the Amazon Kendra Developer Edition that provides usage of up to 750 hours for the first 30 days. After your trial expires, you are charged for all provisioned Amazon Kendra indexes, even if they are empty and no queries are executed. After the trial expires, there are additional charges for scanning and syncing documents using the Amazon Kendra data sources.

For a complete list of charges and prices, see [Amazon Kendra pricing](#).

Are you a first-time Amazon Kendra user?

If you are a first-time user of Amazon Kendra, we recommend that you read the following sections in order:

1. [How Amazon Kendra works \(p. 3\)](#) – Introduces the Amazon Kendra components and describes how you use them to create a search solution.
2. [Getting started \(p. 37\)](#) – Explains how to set up your account and test the Amazon Kendra search API.
3. [Creating an index \(p. 64\)](#) – Explains how to use Amazon Kendra to create a search index and to add data sources to sync your documents.
4. [Adding documents directly to an index \(p. 74\)](#) – Explains how to add documents directly to an Amazon Kendra index.
5. [Searching indexes \(p. 129\)](#) – Explains how to use the Amazon Kendra search API to search an index.
6. [Deploying Amazon Kendra \(p. 32\)](#) – Provides a sample application you can use to deploy Amazon Kendra to your website.

How Amazon Kendra works

Amazon Kendra provides an interface for indexing and searching documents. You can use Amazon Kendra to create an updatable index of documents of a variety of types, including plain text, HTML files, Microsoft Word documents, Microsoft PowerPoint presentations, and PDF files. It has a search API that you can use from a variety of client applications, such as websites or mobile applications.

Amazon Kendra integrates with other services. For example, you can power [Amazon Lex chat bots](#) with Amazon Kendra search to provide answers to users' questions. You can use [Amazon S3 bucket](#) as a data source for your Amazon Kendra index. And you can set up [AWS Identity and Access Management](#) to control access to Amazon Kendra resources.

Amazon Kendra has the following components:

- The *index*, which provides a search API for client queries. You create the index from source documents.
- A *source repository*, which contains the documents to index.
- A *data source* that syncs the documents in your source repositories to an Amazon Kendra index. You can automatically synchronize a data source with an Amazon Kendra index so that new, updated, and deleted files in the source repository are updated in the index.
- A *document addition API*, that adds documents directly to the index.

To manage indexes and data sources, you can use the Amazon Kendra console or the API. You can create, update, and delete indexes. Deleting an index deletes all data sources and permanently deletes all of your document information from Amazon Kendra.

Topics

- [Index \(p. 3\)](#)
- [Documents \(p. 5\)](#)
- [Data sources \(p. 7\)](#)
- [Queries \(p. 7\)](#)
- [Tags \(p. 8\)](#)

Index

An *index* provides search results for documents and frequently asked questions (FAQ) that it has indexed. The way you add documents to the index depends on the type of document and where they are stored.

- If your documents are in a store, such as an S3 bucket or a Microsoft SharePoint site, you use a data source
- You use the Amazon Kendra API to add documents
- For FAQ questions and answers, which must be stored in an Amazon Simple Storage Service (Amazon S3) bucket, you upload them from the bucket

An index can contain documents that are indexed from a data source, documents that are added directly to the index, and FAQs. You can create indexes with the Amazon Kendra console, the AWS CLI, or an AWS SDK. For information about the types of documents that can be indexed, see [Types of documents \(p. 5\)](#).

For information about using a data source with an index, see [Data sources \(p. 7\)](#).

To add documents directly to an index, you call the [BatchPutDocument \(p. 277\)](#) operation. The documents are supplied as plain text, as a binary blob, or using a path to a document stored in an Amazon S3 bucket. For an example, see [Adding documents from an Amazon S3 bucket \(p. 76\)](#).

Index fields

An index contains fields that you map to the attributes of your document. Attributes could include, for example, the document title, main body text, last updated date, and other attributes contained within the structure of your documents. You can also create custom attributes such as figure title, or the business department the document is related to. Fields, which you map to your document attributes, provide the schema for your index. Amazon Kendra uses the fields to search your documents. After you map your fields to your document attributes, you can use the information in the field for searching, for display, and to create facets of the search result.

Amazon Kendra has 14 reserved fields, which you can map to your document attributes:

- `_authors` – A list of one or more authors responsible for the content of the document.
- `_category` – A category that places a document in a specific group.
- `_created_at` – The date and time in ISO 8601 format that the document was created. For example, 20120325T123010+01:00 is the ISO 8601 date-time format for March 25th 2012 at 12:30PM (plus 10 seconds) in Central European Time.
- `_data_source_id` – The identifier of the data source that contains the document.
- `_document_body` – The content of the document.
- `_document_id` – A unique identifier for the document.
- `_document_title` – The title of the document.
- `_excerpt_page_number` – The page number in a PDF file where the document excerpt appears. If your index was created before September 8, 2020, you must re-index your documents before you can use this attribute.
- `_faq_id` – If this is an FAQ question and answer, a unique identifier for them.
- `_file_type` – The file type of the document, such as pdf or doc.
- `_last_updated_at` – The date and time in ISO 8601 format that the document was last updated. For example, 20120325T123010+01:00 is the ISO 8601 date-time format for March 25th 2012 at 12:30PM (plus 10 seconds) in Central European Time.
- `_source_uri` – The URI where the document is available. For example, the URI of the document on a company website.
- `_version` – An identifier for the specific version of a document.
- `_view_count` – The number of times that the document has been viewed.

You can also create custom fields, which you can use like the reserved fields for search and display, and to create facets.

There are four types of custom fields:

- Date
- Number
- String
- String list

You create a custom field using the console or by using the [UpdateIndex \(p. 411\)](#) operation. After you create a custom field, you map it to a document attribute, just as you do with a reserved field. If you

added a document to the index with [BatchPutDocument \(p. 277\)](#) operation, you map the attributes with the API. For documents indexed from an Amazon S3 data source, you map the attributes using a metadata file that contains a JSON structure that describes the document attributes. For documents indexed with a database or SharePoint Online data source, you map attributes with the console or the data source configuration. For more information, see [Document attributes \(p. 6\)](#).

Searching indexes

After you create an index, you can use it for search queries. For more information, see [Searching indexes \(p. 129\)](#).

Documents

Amazon Kendra can index many types of documents. You can also associate attributes with documents to provide information such as the source URI and the author of a document.

Topics

- [Types of documents \(p. 5\)](#)
- [Document attributes \(p. 6\)](#)

Types of documents

An index can include both structured and unstructured text:

- Structured text
 - Frequently asked questions and answers
- Unstructured text
 - HTML files
 - Microsoft PowerPoint presentations
 - Microsoft Word documents
 - Plain text documents
 - PDFs

You can add documents directly to an index by calling the [BatchPutDocument \(p. 277\)](#) operation. You can also add documents from a data source. For information about adding files to a data source, see [Adding documents from a data source \(p. 85\)](#). For an example that shows how to add Microsoft Word documents directly to an index from an Amazon S3 bucket, see [Adding documents from an Amazon S3 bucket \(p. 76\)](#).

An index can contain multiple documents and multiple types of documents.

HTML

HTML format files. You add an HTML file to an index the same way that you add a plain text file.

Plain text

You can add plain text files to an index using the `BatchPutDocument` operation or from a data source. For an example of adding a plain text document directly to an index, see [Adding documents with the API \(p. 75\)](#).

Microsoft Word document

Microsoft Word format files can be added to an index as binary data, from an Amazon S3 bucket, or from an Amazon Kendra data source.

Microsoft PowerPoint document

Microsoft PowerPoint format files can be added to an index as binary data, from an Amazon S3 bucket, or from an Amazon Kendra data source.

Portable document format (PDF)

PDF format files can be added to an index either as binary data, from an Amazon S3 bucket, or from an Amazon Kendra data source.

Frequently asked questions and answers

Frequently asked question and answer format documents are used to answer questions such as *How tall is the Space Needle?* You can specify multiple questions that return the same answer. You specify the questions and answers in a comma-separated values (CSV) file stored in an Amazon S3 bucket.

For an example, see [Adding questions and answers directly to an index \(p. 80\)](#).

Document attributes

A document has attributes associated with it. Attributes of a document are the properties of a document or what is contained within the structure of a document. For example, each of your documents might contain title, body text, and author. You can also add your own custom attributes of your documents. Custom attributes are attributes that you specify for your own needs. For example, if your index searches tax documents, you might specify a custom attribute for the type of tax document such as W-2, 1099, and so on.

Before you can use a document attribute in a query, it must be mapped to a database field. For example, the title attribute can be mapped to the field `_document_title`. For more information, see [Index fields \(p. 4\)](#).

You can use document attributes to filter responses and to make faceted search suggestions. For example, you can filter a response to only return a specific version of a document, or you can filter searches to only return 1099 tax documents that match the search term. For more information, see [Filtering queries \(p. 134\)](#).

You can also use document attributes to manually tune the query response. For example, you can choose to increase the importance of the title field to increase the weight that Amazon Kendra assigns to the field when determining which documents to return in the response. For more information, see [Tuning search relevance \(p. 151\)](#).

Before you can add an attribute, you must create an index field to map the attribute to. You create index fields using the console or by using the [UpdateIndex \(p. 411\)](#) operation.

If you are adding a document directly to an index, you specify the attributes in the [Document \(p. 469\)](#) input parameter to the [the section called "BatchPutDocument" \(p. 277\)](#) operation. You specify the custom attribute values in a [DocumentAttribute \(p. 471\)](#) object array. If you are using a data source, the method that you use to add the document attributes depends on the data source. For more information, see [Creating custom document attributes \(p. 121\)](#).

Data sources

A *data source* is a location, such as an Amazon Simple Storage Service (Amazon S3) bucket, where you store the documents for indexing. You can automatically synchronize data sources with an Amazon Kendra index so that new, updated, or deleted documents in the data source are also added, updated, or deleted in the index for searching on.

Supported data sources are:

- Amazon S3 buckets
- Confluence instances
- Google Workspace Drives
- Amazon RDS for MySQL and Amazon RDS for PostgreSQL databases
- Confluence cloud and Confluence server
- Custom data sources
- Microsoft OneDrive for Business
- Microsoft SharePoint online and SharePoint server (versions 2013 and 2016)
- Salesforce sites
- ServiceNow instances
- Amazon Kendra web crawler
- Amazon WorkDocs

Supported document formats are: plain text, Microsoft Word, Microsoft PowerPoint, HTML, and PDF. For more information, see [Types of documents \(p. 5\)](#).

Note

To create an index, you don't need a data source. You can add documents directly to an index. For more information, see [Adding documents directly to an index \(p. 74\)](#).

To index documents using a data source.

1. [Create an index \(p. 64\)](#).
2. [Create a data source \(p. 85\)](#).

For a walkthrough with the Amazon Kendra console or with the AWS CLI, see [Getting started \(p. 37\)](#).

Queries

To get answers, users query an index. Users can use natural language in their queries. The response contains information, such as the title, a text excerpt, and the location of documents in the index that provide the best answer.

Amazon Kendra uses all of the information that you provide about your documents, not just the contents of the documents, to determine whether a document is relevant to the query. For example, if your index contains information about when documents were last updated, you can tell Amazon Kendra to assign a higher relevance to documents that were updated more recently.

A query can also contain criteria for how to filter the response so that Amazon Kendra returns only documents that satisfy the filter criteria. For example, if you created an index field called *department*, you can filter the response so that only documents with the department field set to *legal* are returned. For more information, see [Filtering queries \(p. 134\)](#).

You can influence the results of a query by tuning the relevance of individual fields in the index. Tuning changes the importance of a field on the results. For example, if you raise the importance of documents with the category *new*, documents with this category are more likely to be included in the response. For more information, see [Tuning search relevance \(p. 151\)](#).

For more information about using queries, see [Searching indexes \(p. 129\)](#).

Tags

Manage your indexes, data sources, and FAQs by assigning metadata to them with *tags*. Use tags to categorize your Amazon Kendra resources in various ways, for example, by purpose, owner, or application, or any combination. Each tag consists of a *key* and a *value*, both of which you define.

Tags help you to:

- Identify and organize your AWS resources. Many AWS services support tagging, so you can assign the same tag to resources in different services to indicate that the resources are related. For example, you can tag an index and the Amazon Lex bot that uses it with the same tag.
- Allocate costs. You activate tags on the AWS Billing and Cost Management dashboard. AWS uses tags to categorize your costs and deliver a monthly cost allocation report to you. For more information, see [Cost Allocation and Tagging in About AWS Billing and Cost Management](#).
- Control access to your resources. You can use tags in AWS Identity and Access Management (IAM) policies that control access to Amazon Kendra resources. You can attach these policies to an IAM role or user to enable tag-based access control. For more information, see [Authorization based on Amazon Kendra tags \(p. 248\)](#).

You can create and manage tags using the AWS Management Console, the AWS Command Line Interface (AWS CLI), or the Amazon Kendra API.

Tagging resources

If you're using the Amazon Kendra console, you can tag resources when you create them or add them later. You can also use the console to update or remove tags.

If you're using the AWS Command Line Interface (AWS CLI) or the Amazon Kendra API, use the following operations to manage tags for your resources:

- [CreateDataSource \(p. 283\)](#) – Apply tags when you create a data source.
- [CreateFaq \(p. 292\)](#) – Apply tags when you create an FAQ.
- [CreateIndex \(p. 296\)](#) – Apply tags when you create an index.
- [ListTagsForResource \(p. 375\)](#) – View the tags associated with a resource.
- [TagResource \(p. 399\)](#) – Add and modify tags for a resource.
- [UntagResource \(p. 401\)](#) – Remove tags from a resource.

Tag restrictions

The following restrictions apply to tags on Amazon Kendra resources:

- Maximum number of tags – 50
- Maximum key length – 128 characters
- Maximum value length – 256 characters

- Valid characters for key and value – a–z, A–Z, space, and the following characters: _ . : / = + - and @
- Keys and values are case sensitive
- Don't use aws : as a prefix for keys; it's reserved for AWS use

Setting up Amazon Kendra

Before using Amazon Kendra, you must have an Amazon Web Services (AWS) account. After you have an AWS account, you can access Amazon Kendra through the Amazon Kendra console, the AWS Command Line Interface (AWS CLI), or the AWS SDKs.

This guide includes examples for AWS CLI, Java, and Python.

Topics

- [Sign up for AWS \(p. 10\)](#)
- [Regions and endpoints \(p. 10\)](#)
- [Setting up the AWS CLI \(p. 10\)](#)
- [Setting up the AWS SDKs \(p. 11\)](#)

Sign up for AWS

When you sign up for Amazon Web Services (AWS), your account is automatically signed up for all services in AWS, including Amazon Kendra. You are charged only for the services that you use.

If you have an AWS account already, skip to the next task. If you don't have an AWS account, use the following procedure to create one.

To sign up for AWS

1. Open <https://aws.amazon.com>, and then choose **Create an AWS Account**.
2. Follow the on-screen instructions to complete the account creation. Note your 12-digit AWS account number. Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.
3. Create an AWS Identity and Access Management (IAM) admin user. See [Creating Your First IAM User and Group](#) in the *AWS Identity and Access Management User Guide* for instructions.

Regions and endpoints

An endpoint is a URL that is the entry point for a web service. Each endpoint is associated with a specific AWS region. If you use a combination of the Amazon Kendra console, the AWS CLI, and the Amazon Kendra SDKs, pay attention to their default regions as all Amazon Kendra components of a given campaign (index, query, etc.) must be created in the same region. For the regions and endpoints supported by Amazon Kendra, see [Regions and Endpoints](#).

Setting up the AWS CLI

The AWS Command Line Interface (AWS CLI) is a unified developer tool for managing AWS services, including Amazon Kendra. We recommend that you install it.

1. To install the AWS CLI, follow the instructions in [Installing the AWS Command Line Interface](#) in the *AWS Command Line Interface User Guide*.

2. To configure the AWS CLI and set up a profile to call the AWS CLI, follow the instructions in [Configuring the AWS CLI](#) in the *AWS Command Line Interface User Guide*.
3. To confirm that the AWS CLI profile is configured properly, run the following command:

```
aws configure --profile default
```

If your profile has been configured correctly, you will see output similar to the following:

```
AWS Access Key ID [*****52FQ]:  
AWS Secret Access Key [*****xgyz]:  
Default region name [us-west-2]:  
Default output format [json]:
```

4. To verify that the AWS CLI is configured for use with Amazon Kendra, run the following commands:

```
aws kendra help
```

If the AWS CLI is configured correctly, you will see a list of the supported AWS CLI commands for Amazon Kendra, Amazon Kendra runtime, and Amazon Kendra events.

Setting up the AWS SDKs

Download and install the AWS SDKs that you want to use. This guide provides examples for Python. For information about other AWS SDKs, see [Tools for Amazon Web Services](#).

IAM access roles for Amazon Kendra

When you create an index, data source, or an FAQ, Amazon Kendra needs access to the AWS resources required to create the Amazon Kendra resource. You must create a AWS Identity and Access Management (IAM) policy before you create the Amazon Kendra resource. When you call the operation, you provide the Amazon Resource Name (ARN) of the role with the policy attached. For example, if you are calling the [BatchPutDocument](#) (p. 277) operation to add documents from an Amazon S3 bucket, you provide Amazon Kendra with a role with a policy that has access to the bucket.

You can create a new IAM role in the Amazon Kendra console or choose an IAM existing role to use. The console displays roles that have the string "kendra" or "Kendra" in the role name.

The following topics provide details for the required policies. If you create IAM roles using the Amazon Kendra console these policies are created for you.

Topics

- IAM roles for indexes (p. 12)
 - IAM Roles for the BatchPutDocument operation (p. 14)
 - IAM roles for data sources (p. 15)
 - IAM roles for frequently asked questions (p. 28)
 - IAM roles for query suggestions (p. 29)
 - IAM roles for principal mapping of users and groups (p. 30)
 - IAM roles for AWS Single Sign-On (p. 31)

IAM roles for indexes

When you create an index, you must provide an IAM role with permission to write to an Amazon CloudWatch Logs. You must also provide a trust policy that allows Amazon Kendra to assume the role. The following are the policies that must be provided.

A role policy to allow Amazon Kendra to access a CloudWatch log.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "cloudwatch:PutMetricData",  
            "Resource": "*",  
            "Condition": {  
                "StringEquals": {  
                    "cloudwatch:namespace": "Kendra"  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": "logs:DescribeLogGroups",  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "logs>CreateLogGroup",  
            "Resource": "*"  
        }  
    ]  
}
```

```

        "Resource": "arn:aws:logs:region:account ID:log-group:/aws/kendra/*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "logs:DescribeLogStreams",
            "logs>CreateLogStream",
            "logs:PutLogEvents"
        ],
        "Resource": "arn:aws:logs:region:account ID:log-group:/aws/kendra/*:log-
stream:*
    }
}

```

A role policy to allow Amazon Kendra to access AWS Secrets Manager. If you are using user context with Secrets Manager as a key location, you can use the following policy.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "cloudwatch:PutMetricData",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "Kendra"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "logs:DescribeLogGroups",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "logs>CreateLogGroup",
            "Resource": "arn:aws:logs:region:account ID:log-group:/aws/kendra/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogStreams",
                "logs>CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:region:account ID:log-group:/aws/kendra/*:log-stream:*
        },
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:region:account ID:secret:secret_id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],

```

```

    "Resource": [
        "arn:aws:kms:region:account ID:key/key_id"
    ],
    "Condition": {
        "StringLike": {
            "kms:ViaService": [
                "secretsmanager.*.amazonaws.com"
            ]
        }
    }
]
}

```

A trust policy to allow Amazon Kendra to assume a role.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "kendra.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
    }
}
```

IAM Roles for the BatchPutDocument operation

When you use the [BatchPutDocument \(p. 277\)](#) operation to index documents in an Amazon S3 bucket, you must provide Amazon Kendra with an IAM role with access to the bucket. You must also provide a trust policy that allows Amazon Kendra to assume the role. If the documents in the bucket are encrypted, you must provide permission to use the AWS KMS customer master key (CMK) to decrypt the documents.

A required role policy to allow Amazon Kendra to access an Amazon S3 bucket.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket name/*"
            ]
        }
    ]
}
```

A required trust policy to allow Amazon Kendra to assume a role.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Sid": "AllowKendraToAssumeAttachedRole",
        "Effect": "Allow",

```

```
    "Principal": {
        "Service": "kendra.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
}
```

An optional role policy to allow Amazon Kendra to use an AWS KMS customer master key (CMK) to decrypt documents in an Amazon S3 bucket.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:region:account ID:key/key ID"
            ]
        }
    ]
}
```

IAM roles for data sources

When you use the [CreateDataSource \(p. 283\)](#) operation, you must give Amazon Kendra an IAM role that has permission to access the database resources. The specific permissions required depend on the data source.

Topics

- [IAM roles for Amazon S3 data sources \(p. 15\)](#)
- [IAM role for Confluence server data sources \(p. 16\)](#)
- [IAM roles for database data sources \(p. 17\)](#)
- [IAM roles for Google Workspace Drive data sources \(p. 19\)](#)
- [IAM roles for Microsoft OneDrive data sources \(p. 20\)](#)
- [IAM role for Salesforce data sources \(p. 21\)](#)
- [IAM role for ServiceNow data sources \(p. 22\)](#)
- [IAM roles for Microsoft SharePoint data sources \(p. 23\)](#)
- [Virtual private cloud \(VPC\) IAM role \(p. 24\)](#)
- [IAM roles for web crawler data sources \(p. 25\)](#)
- [IAM roles for Amazon WorkDocs data sources \(p. 26\)](#)

IAM roles for Amazon S3 data sources

When you use an Amazon S3 bucket as a data source, you supply a role that has permission to access the bucket, and to use the `BatchPutDocument` and `BatchDeleteDocument` operations. If the documents in the Amazon S3 bucket are encrypted, you must provide permission to use the AWS KMS customer master key (CMK) to decrypt the documents.

A required role policy to allow Amazon Kendra to use an Amazon S3 bucket as a data source.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket name/*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "s3>ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::bucket name"
            ],
            "Effect": "Allow"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
                "kendra:BatchDeleteDocument"
            ],
            "Resource": [
                "arn:aws:kendra:region:account ID:index/index ID"
            ]
        }
    ]
}
```

An optional role policy to allow Amazon Kendra to use an AWS KMS customer master key (CMK) to decrypt documents in an Amazon S3 bucket.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:region:account ID:key/key ID"
            ]
        }
    ]
}
```

IAM role for Confluence server data sources

When you use a Confluence server as a data source, you provide a role with the following policies:

- Permission to access the AWS Secrets Manager secret that contains the credentials necessary to connect to the Confluence server. For more information about the contents of the secret, see [Using an Atlassian Confluence data source \(p. 90\)](#).
- Permission to use the AWS KMS customer master key (CMK) to decrypt the user name and password secret stored by Secrets Manager.

- Permission to use the `BatchPutDocument` and `BatchDeleteDocument` operations to update the index.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:region:account ID:secret:secret ID"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:region:account ID:key/key ID"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "secretsmanager.*.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
                "kendra:BatchDeleteDocument"
            ],
            "Resource": "arn:aws:kendra:region:account ID:index/index ID"
        }
    ]
}
```

If you are using a VPC, provide a policy that gives Amazon Kendra access to the required resources. See [Virtual private cloud \(VPC\) IAM role \(p. 24\)](#) for the required policy.

IAM roles for database data sources

When you use a database as a data source, you provide Amazon Kendra with a role that has the permissions necessary for connecting to the database. These include:

- Permission to access the AWS Secrets Manager secret that contains the user name and password for the database site. For more information about the contents of the secret, see [Using a database data source \(p. 100\)](#).
- Permission to use the AWS KMS customer master key (CMK) to decrypt the user name and password secret stored by Secrets Manager.
- Permission to use the `BatchPutDocument` and `BatchDeleteDocument` operations to update the index.
- Permission to access the Amazon S3 bucket that contains the SSL certificate used to communicate with the database site.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:region:account ID:secret:secret ID"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:region:account ID:key/key ID"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
                "kendra:BatchDeleteDocument"
            ],
            "Resource": [
                "arn:aws:kendra:region:account ID:index/index ID"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "kendra.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket name/*"
            ]
        }
    ]
}
```

There are two optional policies that you might use with a database data source.

If you have encrypted the Amazon S3 bucket that contains the SSL certificate used to communicate with the database, provide a policy to give Amazon Kendra access to the key.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:region:account ID:key/key ID"
            ]
        }
    ]
}
```

```
        ]
    }
}
```

If you are using a VPC, provide a policy that gives Amazon Kendra access to the required resources. See [Virtual private cloud \(VPC\) IAM role \(p. 24\)](#) for the required policy.

IAM roles for Google Workspace Drive data sources

When you use a Google Workspace Drive data source, you provide Amazon Kendra with a role that has the permissions necessary for connecting to the site. These include:

- Permission to get and decrypt the AWS Secrets Manager secret that contains the client account email, admin account email, and private key necessary to connect to the Google Drive site. For more information about the contents of the secret, see [Using a Google Workspace Drive data source \(p. 102\)](#).
- Permission to use the [BatchPutDocument \(p. 277\)](#) and [BatchDeleteDocument \(p. 271\)](#) operations.

The following IAM policy provides the necessary permissions:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:region:account ID:secret:secret ID"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:region:account ID:key/key ID"
      ],
      "Condition": {
        "StringLike": {
          "kms:ViaService": [
            "secretsmanager.*.amazonaws.com"
          ]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "kendra:BatchPutDocument",
        "kendra:BatchDeleteDocument"
      ],
      "Resource": "arn:aws:kendra:region:account ID:index/index ID"
    }
  ]
}
```

IAM roles for Microsoft OneDrive data sources

When you use a Microsoft OneDrive data source, you provide Amazon Kendra with a role that has the permissions necessary for connecting to the site. These include:

- Permission to get and decrypt the AWS Secrets Manager secret that contains the application ID and secret key necessary to connect to the OneDrive site. For more information about the contents of the secret, see [Using a Microsoft OneDrive data source \(p. 104\)](#).
- Permission to use the [BatchPutDocument \(p. 277\)](#) and [BatchDeleteDocument \(p. 271\)](#) operations.

The following IAM policy provides the necessary permissions:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:region:account ID:secret:secret ID"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:region:account ID:key/key ID"
      ],
      "Condition": {
        "StringLike": {
          "kms:ViaService": [
            "secretsmanager.*.amazonaws.com"
          ]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "kendra:BatchPutDocument",
        "kendra:BatchDeleteDocument"
      ],
      "Resource": "arn:aws:kendra:region:account ID:index/index ID"
    }
  ]
}
```

If you are storing the list of users to index in an S3 bucket, you must also provide permission to use the S3 `GetObject` operation. The following IAM policy provides the necessary permissions:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket name/*"
      ]
    }
  ]
}
```

```

        "secretsmanager:GetSecretValue"
    ],
    "Resource": [
        "arn:aws:secretsmanager:region:account ID:secret:secret ID"
    ]
},
{
    "Action": [
        "s3:GetObject"
    ],
    "Resource": [
        "arn:aws:s3:::input_bucket_name/*"
    ],
    "Effect": "Allow"
},
{
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt"
    ],
    "Resource": [
        "arn:aws:kms:region:account ID:key/[[key IDs]]"
    ],
    "Condition": {
        "StringLike": {
            "kms:ViaService": [
                "secretsmanager.*.amazonaws.com",
                "s3.*.amazonaws.com"
            ]
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "kendra:BatchPutDocument",
        "kendra:BatchDeleteDocument"
    ],
    "Resource": "arn:aws:kendra:region:account ID:index/index ID"
}
]
}

```

IAM role for Salesforce data sources

When you use a Salesforce as a data source, you provide a role with the following policies:

- Permission to access the AWS Secrets Manager secret that contains the user name and password for the Salesforce site. For more information about the contents of the secret, see [Using a Salesforce data source \(p. 106\)](#).
- Permission to use the AWS KMS customer master key (CMK) to decrypt the user name and password secret stored by Secrets Manager.
- Permission to use the BatchPutDocument and BatchDeleteDocument operations to update the index.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ]
        }
    ]
}
```

```

        ],
        "Resource": [
            "arn:aws:secretsmanager:region:account ID:secret:secret ID"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "kms:Decrypt"
        ],
        "Resource": [
            "arn:aws:kms:region:account ID:key/key ID"
        ],
        "Condition": {
            "StringLike": {
                "kms:ViaService": [
                    "secretsmanager.*.amazonaws.com"
                ]
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kendra:BatchPutDocument",
                "kendra:BatchDeleteDocument"
            ],
            "Resource": "arn:aws:kendra:region:account ID:index/index ID"
        }
    }
}

```

IAM role for ServiceNow data sources

When you use a ServiceNow as a data source, you provide a role with the following policies:

- Permission to access the AWS Secrets Manager secret that contains the user name and password for the ServiceNow site. For more information about the contents of the secret, see [Using a ServiceNow data source \(p. 109\)](#).
- Permission to use the AWS KMS customer master key (CMK) to decrypt the user name and password secret stored by Secrets Manager.
- Permission to use the BatchPutDocument and BatchDeleteDocument operations to update the index.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:region:account ID:secret:secret ID"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [

```

```

        "arn:aws:kms:region:account ID:key/key ID"
    ],
    "Condition": {
        "StringLike": {
            "kms:ViaService": [
                "secretsmanager.*.amazonaws.com"
            ]
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "kendra:BatchPutDocument",
        "kendra:BatchDeleteDocument"
    ],
    "Resource": "arn:aws:kendra:region:account ID:index/index ID"
}
]
}

```

IAM roles for Microsoft SharePoint data sources

For a Microsoft SharePoint data source, you provide a role with the following policies.

- Permission to access the AWS Secrets Manager secret that contains the user name and password for the SharePoint site. For more information about the contents of the secret, see [Using a Microsoft SharePoint data source \(p. 114\)](#).
- Permission to use the AWS KMS customer master key (CMK) to decrypt the user name and password secret stored by Secrets Manager.
- Permission to use the `BatchPutDocument` and `BatchDeleteDocument` operations to update the index.
- Permission to access the Amazon S3 bucket that contains the SSL certificate used to communicate with the SharePoint site.

You must also attach a trust policy that allows Amazon Kendra to assume the role.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:region:account ID:secret:secret ID"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:region:account ID:key/key ID"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [

```

```

        "kendra:BatchPutDocument",
        "kendra:BatchDeleteDocument"
    ],
    "Resource": [
        "arn:aws:kendra:region:account ID:index/index ID"
    ],
    "Condition": {
        "StringLike": {
            "kms:ViaService": [
                "kendra.amazonaws.com"
            ]
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "s3:GetObject"
    ],
    "Resource": [
        "arn:aws:s3:::bucket name/*"
    ]
}
]
}

```

You must apply the following trust policy to the role.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "kendra.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

If you have encrypted the Amazon S3 bucket that contains the SSL certificate used to communicate with the Sharepoint site, provide a policy to give Amazon Kendra access to the key.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:region:account ID:key/key ID"
            ]
        }
    ]
}
```

Virtual private cloud (VPC) IAM role

If you use a virtual private cloud (VPC) to connect to your data source, you must provide the following permissions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterfacePermission"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:AuthorizedService": "kendra.amazonaws.com"
                },
                "ArnEquals": {
                    "ec2:Subnet": [
                        "arn:aws:ec2:region:account ID:subnet/subnet IDs"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeSubnets"
            ],
            "Resource": "*"
        },
        {
            "Sid": "iamPassRole",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "kendra.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

IAM roles for web crawler data sources

When you use Amazon Kendra Web Crawler, you provide a role with the following policies:

- Permission to access the AWS Secrets Manager secret that contains the user name and password to connect to websites or a web proxy server backed by basic authentication. For more information about the contents of the secret, see [Using a web crawler data source](#).
- Permission to use the AWS KMS customer master key (CMK) to decrypt the user name and password secret stored by Secrets Manager.

- Permission to use the `BatchPutDocument` and `BatchDeleteDocument` operations to update the index.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:region:account ID:secret:secret ID"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:region:account ID:key/key ID"
      ],
      "Condition": {
        "StringLike": {
          "kms:ViaService": [
            "secretsmanager.*.amazonaws.com"
          ]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "kendra:BatchPutDocument",
        "kendra:BatchDeleteDocument"
      ],
      "Resource": "arn:aws:kendra:region:account ID:index/index ID"
    }
  ]
}
```

IAM roles for Amazon WorkDocs data sources

When you use Amazon WorkDocs, you provide a role with the following policies

- Permission to verify the directory ID (organization ID) that corresponds with your Amazon WorkDocs site repository.
- Permission to get the domain name of your Active Directory that contains your Amazon WorkDocs site directory.
- Permission to call the required public API actions for the Amazon WorkDocs connector.
- Permission to call the `BatchPutDocument` and `BatchDeleteDocument` actions to update the index.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowsKendraToGetDomainNameOfActiveDirectory",
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole"
      ],
      "Resource": "arn:aws:iam::aws:policy/AmazonWorkDocsDataSourcesAccessPolicy"
    }
  ]
}
```

```

        "Action": "ds:DescribeDirectories",
        "Resource": "*"
    },
    {
        "Sid": "AllowsKendraToCallRequiredWorkDocsAPIs",
        "Effect": "Allow",
        "Action": [
            "workdocs:GetDocumentPath",
            "workdocs:GetGroup",
            "workdocs:GetDocument",
            "workdocs:DownloadDocumentVersions",
            "workdocs:DescribeUsers",
            "workdocs:DescribeFolderContents",
            "workdocs:DescribeActivities",
            "workdocs:DescribeComments",
            "workdocs:GetFolder",
            "workdocs:DescribeResourcePermissions",
            "workdocs:GetFolderPath",
            "workdocs:DescribeInstances"
        ],
        "Resource": "*"
    },
    {
        "Sid": "iamPassRole",
        "Effect": "Allow",
        "Action": "iam:PassRole",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "iam:PassedToService": [
                    "kendra.amazonaws.com"
                ]
            }
        }
    },
    {
        "Sid": "AllowsKendraToCallBatchPutDeleteAPIs",
        "Effect": "Allow",
        "Action": [
            "kendra:BatchPutDocument",
            "kendra:BatchDeleteDocument"
        ],
        "Resource": [
            "arn:aws:kendra:{{region}}:{{account_id}}:index/${IndexId}"
        ]
    }
]
}

```

A required trust policy allow Amazon Kendra to assume a role.

```

{
    "Version": "2012-10-17",
    "Statement": {
        "Sid": "AllowsKendraToAssumeTheAttachedRole",
        "Effect": "Allow",
        "Principal": { "Service": "kendra.amazonaws.com" },
        "Action": "sts:AssumeRole"
    }
}

```

IAM roles for frequently asked questions

When you use the [CreateFaq \(p. 292\)](#) operation to load questions and answers into an index, you must provide Amazon Kendra with an IAM role with access to the Amazon S3 bucket that contains the source files. If the source files are encrypted, you must provide permission to use the AWS KMS customer master key (CMK) to decrypt the files.

A required role policy to allow Amazon Kendra to access an Amazon S3 bucket.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:GetObject"  
            ],  
            "Resource": [  
                "arn:aws:s3:::bucket name/*"  
            ]  
        }  
    ]  
}
```

A required trust policy to allow Amazon Kendra to assume a role.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Sid": "AllowKendraToAssumeAttachedRole",  
        "Effect": "Allow",  
        "Principal": {  
            "Service": "kendra.amazonaws.com"  
        },  
        "Action": "sts:AssumeRole"  
    }  
}
```

An optional role policy to allow Amazon Kendra to use an AWS KMS customer master key (CMK) to decrypt files in an Amazon S3 bucket.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "kms:Decrypt"  
            ],  
            "Resource": [  
                "arn:aws:kms:region:account ID:key ID"  
            ],  
            "Condition": {  
                "StringLike": {  
                    "kms:ViaService": [  
                        "kendra.amazonaws.com"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

```
    ]  
}
```

IAM roles for query suggestions

When you use an Amazon S3 file as a query suggestions block list, you supply a role that has permission to access the S3 file and the Amazon S3 bucket. If the block list text file (the S3 file) in the Amazon S3 bucket is encrypted, you must provide permission to use the AWS KMS customer master key (CMK) to decrypt the documents.

A required role policy to allow Amazon Kendra to use the S3 file as your query suggestions block list.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Action": [  
             "s3:GetObject"  
         ],  
         "Resource": [  
             "arn:aws:s3:::bucket name/*"  
         ]  
     }  
    ]  
}
```

A required trust policy to allow Amazon Kendra to assume a role.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Sid": "AllowKendraToAssumeAttachedRole",  
        "Effect": "Allow",  
        "Principal": {  
            "Service": "kendra.amazonaws.com"  
        },  
        "Action": "sts:AssumeRole"  
    }  
}
```

An optional role policy to allow Amazon Kendra to use an AWS KMS customer master key (CMK) to decrypt documents in an Amazon S3 bucket.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Action": [  
             "kms:Decrypt"  
         ],  
         "Resource": [  
             "arn:aws:kms:region:account ID:key/key ID"  
         ]  
     }  
    ]  
}
```

IAM roles for principal mapping of users and groups

When you use the [PutPrincipalMapping](#) operation to map users to their groups for filtering search results by user context, you need to provide a list of users or sub groups that belong to a group. If your list is more than 1000 users or sub groups for a group, you need to supply a role that has permission to access the Amazon S3 file of your list and the Amazon S3 bucket. If the text file (the S3 file) of the list in the Amazon S3 bucket is encrypted, you must provide permission to use the AWS KMS customer master key (CMK) to decrypt the documents.

A required role policy to allow Amazon Kendra to use the Amazon S3 file as your list of users and sub groups that belong to a group.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
            "Action": [  
                "s3:GetObject"  
            ],  
            "Resource": [  
                "arn:aws:s3:::bucket name/*"  
            ]  
        }  
    ]  
}
```

A required trust policy to allow Amazon Kendra to assume a role.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Sid": "AllowKendraToAssumeAttachedRole",  
        "Effect": "Allow",  
        "Principal": {  
            "Service": "kendra.amazonaws.com"  
        },  
        "Action": "sts:AssumeRole"  
    }  
}
```

An optional role policy to allow Amazon Kendra to use an AWS KMS customer master key (CMK) to decrypt documents in an Amazon S3 bucket.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
            "Action": [  
                "kms:Decrypt"  
            ],  
            "Resource": [  
                "arn:aws:kms:region:account ID:key/key ID"  
            ]  
        }  
    ]  
}
```

IAM roles for AWS Single Sign-On

When you use the [UserGroupResolutionConfiguration](#) structure to fetch access levels of groups and users from an AWS Single Sign-On identity source, you need to supply a role that has permission to access AWS SSO.

A required role policy to allow Amazon Kendra to access AWS SSO.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "sso-directory:SearchUsers",  
                "sso-directory>ListGroupsForUser",  
                "sso-directory:DescribeGroups",  
                "sso>ListDirectoryAssociations"  
            ],  
            "Resource": [  
                "*"  
            ]  
        },  
        {  
            "Sid": "iamPassRole",  
            "Effect": "Allow",  
            "Action": "iam:PassRole",  
            "Resource": "*",  
            "Condition": {  
                "StringEquals": {  
                    "iam:PassedToService": [  
                        "kendra.amazonaws.com"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

Deploying Amazon Kendra

When it comes time to deploy Amazon Kendra search to your website, we provide source code that you can use with React to get a head start on your application. The source code is provided free of charge under a modified MIT license so that you can use it as is or change it for your own needs. There are two examples:

- <https://kendrasamples.s3.amazonaws.com/kendrasamples-react-app.zip> – An example React application that provides sample data and a search page.
- <https://kendrasamples.s3.amazonaws.com/kendrasamples.zip> – A library that you can add to an existing React application.

The examples are modeled after the search page of the Amazon Kendra console. They have the same features for searching and displaying search results. You can use the whole example, or you can choose just one of the features for your own use.

To see the three components of the search page in the Amazon Kendra console, choose the code icon (</>) from the right menu. Hover your pointer over each section to see a brief description of the component and to get the URL of the component's source.

Topics

- [Overview \(p. 32\)](#)
- [Prerequisites \(p. 32\)](#)
- [Setting up the example \(p. 33\)](#)
- [Main search page \(p. 33\)](#)
- [Search component \(p. 33\)](#)
- [Results component \(p. 33\)](#)
- [Facets component \(p. 33\)](#)
- [Pagination component \(p. 34\)](#)

Overview

You add the example code to an existing React application to enable search. The search files and components are structured as follows:

- Main search page – this is the main page that contains all of the components. This is where you will integrate your application with the Amazon Kendra API.
- Search bar – this is the component where a user enters a search term and that calls the search function.
- Results – this is the component that displays the results from Amazon Kendra. It has three components: Suggested answers, FAQ results, and recommended documents.
- Facets – This is the component that shows the facets in the search results and enables you to choose a facet to limit the search.
- Pagination – this is the component that paginates the response from Amazon Kendra.

Prerequisites

Before you begin you need the following:

- An existing React Web application or the example application.
- A development environment configured with the correct libraries.
- The SDK for Java or AWS SDK for JavaScript.

Information about the required libraries and AWS SDKs is in the Readme file in the .zip files.

Setting up the example

A complete procedure for adding Amazon Kendra search to a React application is in the Readme included in the example zip files.

Main search page

The main search page contains all of the example search components. It includes the search bar component for output, the results components to display the response from the [Query \(p. 384\)](#) operation, and a pagination component that enables you to page through the response.

Search component

The search component provides a text box to enter query text. The `onSearch` function is a hook that calls the main function in `Search.tsx` to make the Amazon Kendra [Query \(p. 384\)](#) operation call.

Results component

The results component shows the response from the `Query` operation. The results are shown in three separate areas.

- Suggested answers – These are the top results returned by the `Query` operation. It contains up to three suggested answers. In the response, they have the result type `ANSWER`.
- FAQ answers – These are the frequently asked questions results returned by the response. FAQs are added to the index separately. In the response, they have the type `QUESTION_ANSWER`. For more information, see [Adding questions and answers directly to an index \(p. 80\)](#).
- Recommended documents – These are additional documents that Amazon Kendra returns in the response. In the response from the `Query` operation, they have the type `DOCUMENT`.

The results components share a set of components for features like highlighting, titles, links, etc. The shared components must be present for the result components to work.

Facets component

The facets component lists the facets available in the search results. Each facet classifies the response along a specific dimension, such as author. You can refine the search to a specific facet by choosing one from the list.

After you select a facet, the component calls the `Query` operation with an attribute filter that restricts the search to only those documents that match the facet.

Pagination component

The pagination component enables you to display the search results from the `Query` operation in multiple pages. It calls the `Query` operation with the `PageSize` and `PageNumber` parameters to get a specific page of results.

Adjusting capacity

Amazon Kendra provides resources for your index in *capacity units*. Each capacity unit provides additional resources for your index. There are separate capacity units for storage and for queries. You can only add capacity units to Amazon Kendra Enterprise indexes. You can't add capacity to a Developer edition index.

A document storage capacity unit provides the following additional storage for your index.

- 100,000 documents or 30 GB of storage.

A query capacity unit provides the following additional queries for your index.

- 0.1 queries per second or approximately 8,000 queries per day.

Each index comes with a base capacity equal to 1 capacity unit. There is an additional cost for each additional capacity unit. For details, see [Amazon Kendra pricing](#).

You can adjust capacity units up to 5 times per day to tune the capacity of your index to the expected usage. You can't reduce document storage capacity below the number of documents stored in your index. For example, if you are storing 150,000 documents, you can't reduce the storage capacity below 1 additional unit.

You can add up to 100 capacity units to your storage and query resources. If you need additional resources, [contact AWS support](#).

You can use the Amazon Kendra console or the [DescribeIndex \(p. 335\)](#) operation to view the resources that your index is using. Amazon Kendra also returns exceptions when you exceed the capacity of an index. You get a `ServiceQuotaExceededException` when the total extracted size of all the documents exceeds the limit for an index. You get a `InvalidRequest` for each document when the number of documents exceeds the limit for an index. You get a `ThrottlingException` when the number of queries per second exceeds the limit. For more information on limits, see [Quotas for Amazon Kendra](#).

Viewing capacity

View the resources that your index is using with the Amazon Kendra console. The console provides graphs that you can use to determine how much storage and query capacity your index uses. You can use this information to help you plan when to add additional capacity.

To view document storage and query use (Console)

1. Sign into the AWS Management Console and open the Amazon Kendra console at <https://console.aws.amazon.com/kendra/home>.
2. From the list of indexes, choose the index to view capacity.
3. The **Document count** and **Storage used** fields of the **Index details** show you the amount of storage that your index uses. Scroll to the the **Queries per second** chart to see a graph of the queries per second for your index.

Adding and removing capacity

If you need additional capacity for your index, you can add it using the console or the Amazon Kendra API. When you add capacity, don't add more than you need to help reduce costs.

To add or remove storage or query capacity (Console)

1. Sign into the AWS Management Console and open the Amazon Kendra console at <https://console.aws.amazon.com/kendra/home>.
2. From the list of indexes, choose the index that you want to add or remove capacity.
3. From the **Actions** menu, choose **Edit**.
4. On the **Specify index details** page, choose **Next**.
5. On the **Add additional capacity** page, choose the new query per second and document storage capacity units that you want to use for the index.
6. Choose **Update** to save your changes and update your index to the new capacity.

After you update the capacity of your index, it can take up 60 minutes for the changes to take effect.

To add or remove capacity using the Amazon Kendra API, use the `CapacityUnits` parameter [UpdateIndex \(p. 411\)](#) operation.

Query suggestions capacity

When using [query suggestions](#), there's a base query capacity of 2.5 `GetQuerySuggestions` calls per second. The `GetQuerySuggestions` capacity is five times the provisioned query capacity for an index, or the base capacity of 2.5 calls per second, whichever is higher. For example, the base capacity for an index is 0.1 queries per second, and `GetQuerySuggestions` capacity has a base of 2.5 calls per second. If you add another 0.1 queries per second to total 0.2 queries per second for an index, the `GetQuerySuggestions` capacity is 2.5 calls per second (higher than five times 0.2 queries per second).

Adaptive query bursting

Amazon Kendra has a provisioned base capacity of 1 query capacity unit. You can use up to 8,000 queries per day with a minimum throughput of 0.1 queries per second (per query capacity unit).

An adaptive approach to handling unexpected traffic beyond the provisioned throughput is Amazon Kendra's built in *adaptive query bursting*. This allows you to apply unused query capacity to handle unexpected traffic. Amazon Kendra accumulates your unused queries at your provisioned queries per second rate, every second, up to the maximum number of queries you've provisioned for your Amazon Kendra index. These accumulated queries are used for unexpected traffic above the allocated capacity. Optimal performance of adaptive query bursting can vary, depending on several factors such as your total index size, query complexity, accumulated unused queries, and overall load on your index. It is recommended that you perform your own load tests to accurately measure bursting capacity.

Adaptive query bursting is only available in the enterprise edition of Amazon Kendra.

Getting started

This section shows how to create a data source and add documents, create an index, get search results, and get code samples to help you integrate Kendra into your application. Instructions are provided for the AWS console, the AWS CLI, a Python program using the AWS SDK for Python (Boto3), and a Java program using the AWS SDK for Java.

Topics

- [Prerequisites \(p. 37\)](#)
- [Getting started with an S3 bucket \(console\) \(p. 40\)](#)
- [Getting started \(AWS CLI\) \(p. 41\)](#)
- [Getting started \(AWS SDK for Python \(Boto3\)\) \(p. 42\)](#)
- [Getting started \(AWS SDK for Java\) \(p. 44\)](#)
- [Getting started with a Confluence data source \(console\) \(p. 47\)](#)
- [Getting started with a Google Workspace Drive data source \(console\) \(p. 48\)](#)
- [Getting started with a Microsoft OneDrive for Business data source \(console\) \(p. 50\)](#)
- [Getting started with an Amazon S3 data source \(console\) \(p. 52\)](#)
- [Getting started with a MySQL database data source \(console\) \(p. 53\)](#)
- [Getting started with a Salesforce data source \(console\) \(p. 54\)](#)
- [Getting started with a ServiceNow data source \(console\) \(p. 55\)](#)
- [Getting started with a Microsoft SharePoint data source \(console\) \(p. 56\)](#)
- [Getting started with Amazon Kendra Web Crawler \(console\) \(p. 58\)](#)
- [Getting started with an Amazon WorkDocs data source \(console\) \(p. 60\)](#)
- [Getting started with an AWS Single Sign-On identity source \(console\) \(p. 61\)](#)

Prerequisites

The following steps are prerequisites for the getting started exercises. The steps show you how to set up your account, create an IAM role that gives Amazon Kendra permission to make calls on your behalf, and upload documents for indexing into an Amazon S3 bucket.

1. Create an AWS account and an AWS Identity and Access Management user, as specified in [Sign up for AWS \(p. 10\)](#).
2. If you are using an S3 bucket containing documents to test Amazon Kendra, create an S3 bucket in the same region that you are using Amazon Kendra. For instructions, see [Creating and Configuring an S3 Bucket](#) in the *Amazon Simple Storage Service User Guide*.

Upload your documents to your S3 bucket. For instructions, see [Uploading, Downloading, and Managing Objects](#) in the *Amazon Simple Storage Service User Guide*.

If you are using another data source, you must have an active site and credentials to connect to the data source.

If you are using the console to get started, do [Getting started with an S3 bucket \(console\) \(p. 40\)](#).

Prerequisites for the AWS CLI and SDK

If you are using the AWS CLI or the SDK, you need to create IAM roles and polices for Kendra to use to access resources. If you are using the console, you don't need to create IAM roles and polices. They are created as part of the console procedure.

To create an IAM policy and role for the AWS CLI and SDK that enables Kendra to access your Amazon CloudWatch Logs.

1. Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>.
2. From the left menu, choose **Policies** and then choose **Create policy**.
3. Choose **JSON** and then replace the default policy with the following:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "cloudwatch:PutMetricData"  
            ],  
            "Resource": "*",  
            "Condition": {  
                "StringEquals": {  
                    "cloudwatch:namespace": "AWS/Kendra"  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "logs:DescribeLogGroups"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "logs:CreateLogGroup"  
            ],  
            "Resource": [  
                "arn:aws:logs:region:account ID:log-group:/aws/kendra/*"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "logs:DescribeLogStreams",  
                "logs:CreateLogStream",  
                "logs:PutLogEvents"  
            ],  
            "Resource": [  
                "arn:aws:logs:region:account ID:log-group:/aws/kendra/*:log-stream:/*"  
            ]  
        }  
    ]  
}
```

4. Choose **Review policy**.
5. Name the policy "KendraPolicyForGettingStartedIndex" and then choose **Create policy**.

6. From the left menu, choose **Roles** and then choose **Create role**.
7. Choose **Another AWS account** and then type your account ID in **Account ID**. Choose **Next: Permissions**.
8. Choose the policy that you created above and then choose **Next: Tags**
9. Don't add any tags. Choose **Next: Review**.
10. Name the role "KendraRoleForGettingStartedIndex" and then choose **Create role**.
11. Find the role that you just created. Choose the role name to open the summary. Choose **Trust relationships** and then choose **Edit trust relationship**.
12. Replace the existing trust relationship with the following:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "kendra.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

13. Choose **Update trust policy**.

To create an IAM policy and role that enables Kendra to access and index your Amazon S3 bucket.

1. Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>.
2. From the left menu, choose **Policies** and then choose **Create policy**.
3. Choose **JSON** and then replace the default policy with the following:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket name/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3>ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::bucket name"
      ],
      "Effect": "Allow"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kendra:BatchPutDocument",
        "kendra:BatchDeleteDocument"
      ]
    }
  ]
}
```

```
        ],
        "Resource": "arn:aws:kendra:region:account ID:index/*"
    }
}
```

4. Choose **Review policy**.
5. Name the policy "KendraPolicyForGettingStartedDataSource" and then choose **Create policy**.
6. From the left menu, choose **Roles** and then choose **Create role**.
7. Choose **Another AWS account** and then type your account ID in **Account ID**. Choose **Next: Permissions**.
8. Choose the policy that you created above and then choose **Next: Tags**
9. Don't add any tags. Choose **Next: Review**.
10. Name the role "KendraRoleForGettingStartedDataSource" and then choose **Create role**.
11. Find the role that you just created. Choose the role name to open the summary. Choose **Trust relationships** and then choose **Edit trust relationship**.
12. Replace the existing trust relationship with the following:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "kendra.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

13. Choose **Update trust policy**.

Depending on how you want to use the Amazon Kendra API, do one of the following.

- [Getting started \(AWS CLI\) \(p. 41\)](#)
- [Getting started \(AWS SDK for Java\) \(p. 44\)](#)
- [Getting started \(AWS SDK for Python \(Boto3\)\) \(p. 42\)](#)

Getting started with an S3 bucket (console)

The following procedures show how to create and test an Amazon Kendra index by using the AWS console. In the procedures you create an index and a data source for an index. Finally, you test your index by making a search request.

Step 1: To create an index (Console)

1. Choose **Create index** to start creating a new index.
2. In **Specify index details**, give your index a name and a description.
3. In **IAM role** choose **Create a new role** and then give the role a name. The IAM role will have the prefix "AmazonKendra-".
4. Leave all of the other fields at their defaults. Choose **Next**.

5. In the **Configure user access control** page, choose **Next**.
6. **Provisioning details** page, choose **Developer edition**.
7. Choose **Create** to create your index.
8. Wait for your index to be created. Kendra provisions the hardware for your index. This operation can take some time.

Step 2: To add a data source to an index (Console)

- Create a data source that connects the Amazon Kendra index to your documents. You can choose from one of the following procedures to create a data source.
 - [Getting started with a Confluence data source \(console\) \(p. 47\)](#)
 - [Getting started with a Microsoft OneDrive for Business data source \(console\) \(p. 50\)](#)
 - [Getting started with an Amazon S3 data source \(console\) \(p. 52\)](#)
 - [Getting started with a Salesforce data source \(console\) \(p. 54\)](#)
 - [Getting started with a ServiceNow data source \(console\) \(p. 55\)](#)
 - [Getting started with a Microsoft SharePoint data source \(console\) \(p. 56\)](#)

Step 3: To search an index (Console)

1. In the navigation pane, choose **Search console**
2. Enter a search term that's appropriate for your index. The **top results** and **top document** results are shown.

Getting started (AWS CLI)

The following procedure shows how to create an Amazon Kendra index using the AWS CLI. The procedure creates a data source, index, and runs a query on the index.

To create an Amazon Kendra index (CLI)

1. Do the [Prerequisites \(p. 37\)](#).
2. Enter the following command to create an index

```
aws kendra create-index \
--name cli-getting-started-index \
--description "Index for CLI getting started guide." \
--role-arn arn:aws:iam::account id:role/KendraRoleForGettingStartedIndex
```

3. Wait for Amazon Kendra to create the index. Check the progress using the following command. When the status field is ACTIVE, go on to the next step.

```
aws kendra describe-index \
--id index id
```

4. At the command prompt, enter the following command to create a data source.

```
aws kendra create-data-source \
--index-id index id \
--name data source name \
--role-arn arn:aws:iam::account id:role/KendraRoleForGettingStartedDataSource \
--type S3 \
```

- ```
--configuration '{"S3Configuration":{"BucketName":"'${S3 bucket name}'}}'
```
5. It will take Amazon Kendra a while to create the data source. Enter the following command to check the progress. When the status is ACTIVE, go on to the next step.
- ```
aws kendra describe-data-source \
--id ${data source ID} \
--index-id ${index ID}
```
6. Enter the following command to synchronize the data source.
- ```
aws kendra start-data-source-sync-job \
--id ${data source ID} \
--index-id ${index ID}
```
7. Kendra will index your data source. The amount of time that it takes depends on the number of documents. You can check the status of the sync job using the following command. When the status is ACTIVE, go on to the next step.
- ```
aws kendra describe-data-source \
--id ${data source ID} \
--index-id ${index ID}
```
8. Enter the following command to make a query.

```
aws kendra query \
--index-id ${index ID} \
--query-text "search term"
```

The results of the search are displayed in JSON format.

Getting started (AWS SDK for Python (Boto3))

The following program is an example of using Amazon Kendra in a Python program. The program performs the following actions:

1. Creates a new index using the [CreateIndex \(p. 296\)](#) operation.
2. Waits for index creation to complete. It uses the [DescribeIndex \(p. 335\)](#) operation to monitor the status of the index.
3. Once the index is active, it creates a data source using the [CreateDataSource \(p. 283\)](#) operation.
4. Waits for data source creation to complete. It uses the [DescribeDataSource \(p. 322\)](#) operation to monitor the status of the data source.
5. When the data source is active, it synchronizes the index with the contents of the data source using the [StartDataSourceSyncJob \(p. 392\)](#) operation.

```
import boto3
from botocore.exceptions import ClientError
import pprint
import time

kendra = boto3.client("kendra")

print("Create an index")
```

```
description = "Getting started index"
index_name = "python-getting-started-index"
index_role_arn = "arn:aws:iam::${accountId}:role/KendraRoleForGettingStartedIndex"

try:
    index_response = kendra.create_index(
        Description = description,
        Name = index_name,
        RoleArn = index_role_arn
    )

    pprint.pprint(index_response)

    index_id = index_response["Id"]

    print("Wait for Kendra to create the index.")

    while True:
        # Get index description
        index_description = kendra.describe_index(
            Id = index_id
        )
        # When status is not CREATING quit.
        status = index_description["Status"]
        print("      Creating index. Status: "+status)
        time.sleep(60)
        if status != "CREATING":
            break

    print("Create an S3 data source")

    data_source_name = "python-getting-started-data-source"
    data_source_description = "Getting started data source."
    s3_bucket_name = "${bucketName}"
    data_source_type = "S3"
    data_source_role_arn = "arn:aws:iam::${accountId}:role/
KendraRoleForGettingStartedDataSource"

    configuration = {"S3Configuration":
        {
            "BucketName": s3_bucket_name
        }
    }

    data_source_response=kendra.create_data_source(
        Configuration = configuration,
        Name = data_source_name,
        Description = description,
        RoleArn = data_source_role_arn,
        Type = data_source_type,
        IndexId = index_id
    )

    pprint.pprint(data_source_response)

    data_source_id = data_source_response["Id"]

    print("Wait for Kendra to create the data source.")

    while True:
        data_source_description = kendra.describe_data_source(
            Id = data_source_id,
            IndexId = index_id
        )
        # When status is not CREATING quit.
```

```

status = data_source_description["Status"]
print("    Creating data source. Status: "+status)
time.sleep(60)
if status != "CREATING":
    break

print("Synchronize the data source.")

sync_response = kendra.start_data_source_sync_job(
    Id = data_source_id,
    IndexId = index_id
)
pprint.pprint(sync_response)

print("Wait for the data source to sync with the index.")

while True:

    jobs = kendra.list_data_source_sync_jobs(
        Id=data_source_id,
        IndexId=index_id
    )

    # There should be exactly one job item in response
    status = jobs["History"][0]["Status"]

    print("    Syncing data source. Status: "+status)
    if status != "SYNCING":
        break
    time.sleep(60)

except ClientError as e:
    print("%s" % e)

print("Program ends.")

```

Getting started (AWS SDK for Java)

The following program is an example of using Amazon Kendra in a Java program. The program performs the following actions:

1. Creates a new index using the [CreateIndex \(p. 296\)](#) operation.
2. Waits for index creation to complete. It uses the [DescribeIndex \(p. 335\)](#) operation to monitor the status of the index.
3. Once the index is active, it creates a data source using the [CreateDataSource \(p. 283\)](#) operation.
4. Waits for data source creation to complete. It uses the [DescribeDataSource \(p. 322\)](#) operation to monitor the status of the data source.
5. When the data source is active, it synchronizes the index with the contents of the data source using the [StartDataSourceSyncJob \(p. 392\)](#) operation.

```

package com.amazonaws.kendra;

import java.util.concurrent.TimeUnit;
import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.CreateDataSourceRequest;
import software.amazon.awssdk.services.kendra.model.CreateDataSourceResponse;
import software.amazon.awssdk.services.kendra.model.CreateIndexRequest;

```

```
import software.amazon.awssdk.services.kendra.model.CreateIndexResponse;
import software.amazon.awssdk.services.kendra.model.DataSourceConfiguration;
import software.amazon.awssdk.services.kendra.model.DataSourceStatus;
import software.amazon.awssdk.services.kendra.model.DataSourceSyncJob;
import software.amazon.awssdk.services.kendra.model.DataSourceSyncJobStatus;
import software.amazon.awssdk.services.kendra.model.DataSourceType;
import software.amazon.awssdk.services.kendra.model.DescribeDataSourceRequest;
import software.amazon.awssdk.services.kendra.model.DescribeDataSourceResponse;
import software.amazon.awssdk.services.kendra.model.DescribeIndexRequest;
import software.amazon.awssdk.services.kendra.model.DescribeIndexResponse;
import software.amazon.awssdk.services.kendra.model.IndexStatus;
import software.amazon.awssdk.services.kendra.model.ListDataSourceSyncJobsRequest;
import software.amazon.awssdk.services.kendra.model.ListDataSourceSyncJobsResponse;
import software.amazon.awssdk.services.kendra.model.S3DataSourceConfiguration;
import software.amazon.awssdk.services.kendra.model.StartDataSourceSyncJobRequest;
import software.amazon.awssdk.services.kendra.model.StartDataSourceSyncJobResponse;

public class CreateIndexAndDataSourceExample {

    public static void main(String[] args) throws InterruptedException {
        System.out.println("Create an index");

        String indexDescription = "Getting started index for Kendra";
        String indexName = "java-getting-started-index";
        String indexRoleArn = "arn:aws:iam::<your AWS account ID>:role/<name of an IAM role>";

        System.out.println(String.format("Creating an index named %s", indexName));
        KendraClient kendra = KendraClient.builder().build();

        CreateIndexRequest createIndexRequest = CreateIndexRequest
            .builder()
            .description(indexDescription)
            .name(indexName)
            .roleArn(indexRoleArn)
            .build();
        CreateIndexResponse createIndexResponse = kendra.createIndex(createIndexRequest);
        System.out.println(String.format("Index response %s", createIndexResponse));

        String indexId = createIndexResponse.id();

        System.out.println(String.format("Waiting until the index with index ID %s is created", indexId));
        while (true) {
            DescribeIndexRequest describeIndexRequest =
                DescribeIndexRequest.builder().id(indexId).build();
            DescribeIndexResponse describeIndexResponse =
                kendra.describeIndex(describeIndexRequest);
            IndexStatus status = describeIndexResponse.status();
            if (status != IndexStatus.CREATING) {
                break;
            }
            TimeUnit.SECONDS.sleep(60);
        }

        System.out.println("Creating an S3 data source");
        String dataSourceName = "java-getting-started-data-source";
        String dataSourceDescription = "Getting started data source";
        String s3BucketName = "an-aws-kendra-test-bucket";
        String dataSourceRoleArn = "arn:aws:iam::<your AWS account ID>:role/<name of an IAM role>";

        CreateDataSourceRequest createDataSourceRequest = CreateDataSourceRequest
            .builder()
```

```

        .indexId(indexId)
        .name(dataSourceName)
        .description(dataSourceDescription)
        .roleArn(dataSourceRoleArn)
        .type(DataSourceType.S3)
        .configuration(
            DataSourceConfiguration
            .builder()
            .s3Configuration(
                S3DataSourceConfiguration
                .builder()
                .bucketName(s3BucketName)
                .build()
            ).build()
        ).build();
    ).build();

    CreateDataSourceResponse createDataSourceResponse =
kendra.createDataSource(createDataSourceRequest);
    System.out.println(String.format("Response of creating data source: %s",
createDataSourceResponse));

    String dataSourceId = createDataSourceResponse.id();
    System.out.println(String.format("Waiting for Kendra to create the data source %s",
dataSourceId));
    DescribeDataSourceRequest describeDataSourceRequest = DescribeDataSourceRequest
        .builder()
        .indexId(indexId)
        .id(dataSourceId)
        .build();

    while (true) {
        DescribeDataSourceResponse describeDataSourceResponse =
kendra.describeDataSource(describeDataSourceRequest);

        DataSourceStatus status = describeDataSourceResponse.status();
        System.out.println(String.format("Creating data source. Status: %s", status));
        if (status != DataSourceStatus.CREATING) {
            break;
        }

        TimeUnit.SECONDS.sleep(60);
    }

    System.out.println(String.format("Synchronize the data source %s", dataSourceId));
    StartDataSourceSyncJobRequest startDataSourceSyncJobRequest =
StartDataSourceSyncJobRequest
        .builder()
        .indexId(indexId)
        .id(dataSourceId)
        .build();
    StartDataSourceSyncJobResponse startDataSourceSyncJobResponse =
kendra.startDataSourceSyncJob(startDataSourceSyncJobRequest);
    System.out.println(String.format("Waiting for the data source to sync with the
index %s for execution ID %s", indexId, startDataSourceSyncJobResponse.executionId()));

    // For this particular list, there should be just one job
    ListDataSourceSyncJobsRequest listDataSourceSyncJobsRequest =
ListDataSourceSyncJobsRequest
        .builder()
        .indexId(indexId)
        .id(dataSourceId)
        .build();

    while (true) {
        ListDataSourceSyncJobsResponse listDataSourceSyncJobsResponse =
kendra.listDataSourceSyncJobs(listDataSourceSyncJobsRequest);

```

```
        DataSourceSyncJob job = listDataSourceSyncJobsResponse.history().get(0);
        System.out.println(String.format("Syncing data source. Status: %s",
        job.status()));

        if (job.status() != DataSourceSyncJobStatus.SYNCING) {
            break;
        }

        TimeUnit.SECONDS.sleep(60);

    }

    System.out.println("Index setup is complete");
}
```

Getting started with a Confluence data source (console)

You can use the Amazon Kendra console to get started using a Confluence instance. When you use the console you specify the connection information you need to index the contents of a Confluence instance. For more information, see [Using an Atlassian Confluence data source \(p. 90\)](#).

Use the following procedure to create a basic Confluence data source using the default configuration. The procedure assumes that you have already created an index following the steps in step 1 of [Getting started with an S3 bucket \(console\) \(p. 40\)](#).

- You have an administrator account on the Confluence instance.
- You have a password for the account on a Confluence server instance or an API token for the account on a Confluence cloud instance.
- You created an index following the steps in step 1 of [Getting started with an S3 bucket \(console\) \(p. 40\)](#).

To create a Confluence data source using the Amazon Kendra console

1. Sign into the AWS Management Console and open the Amazon Kendra console at <https://console.aws.amazon.com/kendra/home>.
2. From the list of indexes, choose the index that you want to add the data source to.
3. Choose **Add data sources**.
4. From the list of data source connectors, choose **Confluence**.
5. On **Specify data source details**, give your data source a name and optionally a description. Leave the **Tags** field blank. Choose **Next** to continue.
6. Choose whether you are connecting to a Confluence cloud or Confluence server instance.
7. On **Define access and security**, in **Set source**, enter the URL of your Confluence instance.
8. In **Set authentication**, perform the following steps:
 - a. For the **Type of authentication**, choose **New**.
 - b. In **Secret name**, enter a name for the AWS Secrets Manager secret that will contain your user credentials.
 - c. In **User name**, enter the Confluence account that Amazon Kendra uses to connect to the Confluence instance. The account should have administrator permissions.

- d. For Confluence server enter the account password for the Confluence account in the password field. For Confluence cloud, enter the API token.
 - e. Choose **Save authentication** to save the Secrets Manager secret.
9. If you use Confluence Server, and your server is not publicly accessible, you need to choose a Virtual Private Cloud (VPC), otherwise you may choose **No VPC**.
 10. In **IAM role** choose **Create a new role**. Enter a name for the role in the **Role name** field. Choose **Next** to continue.
 11. Choose **Next** to continue.
 12. On **Set sync scope**, leave the defaults.
 13. In **Frequency**, choose **Run on demand** and then choose **Next**.
 14. In **Set field mappings - optional**, leave the defaults and choose **Next**.
 15. On **Review and create**, review the settings for the data source. Use the **Edit** buttons to make any changes that you need to make. When you are satisfied with the settings, choose **Create** to create your Confluence data source.

After you choose **Create**, Amazon Kendra starts creating the data source. It can take several minutes for the data source to be created. When it is finished, the status changes from **Creating** to **Active**.

After creating the index, you need to sync the Amazon Kendra index with the data source. Choose **Sync now** to start the sync process. It can take several minutes to several hours to synchronize the data source, depending on the number and size of the documents.

Getting started with a Google Workspace Drive data source (console)

You can use the Amazon Kendra console to index a Google Workspace Drive. Using the console, you can specify all of the connection information that you need to index the contents of a Google Drive. For more information, see [Using a Google Workspace Drive data source \(p. 102\)](#).

Use the following procedure to create a basic Google Drive data source using the default configuration. The procedure assumes that you have already created an index following the steps in step 1 of [Getting started with an S3 bucket \(console\) \(p. 40\)](#).

Before you can create a Google Drive data source, you must create a service account that Amazon Kendra uses to connect to your Google Drive.

Step 1: To create a Google Drive service account

1. Log into the [Google Cloud Platform console](#) with an account that has administrator privilege.
2. From the top menu, choose your project.
3. From the top-left menu, choose **IAM & Admin / Service Accounts**.
4. From the top menu, choose **CREATE SERVICE ACCOUNT**.
5. Enter a service account name and description. The service account ID, an email address, is generated automatically. Choose **Create**.
6. Skip steps 2 and 3 on the page. Choose **Done** to continue.

Step 2: Configure the service account.

Begin this procedure immediately after you finish step 1.

1. On the service account page, choose **SHOW DOMAIN-WIDE DELEGATION** to view the available options.
2. Choose **Enable G Suite Domain-wide Delegation**.
3. On the service account page, choose the three vertical dots under **Actions** and choose **Edit** to configure the service account.
4. Choose **CREATE KEY**, set the **Key Type** to **JSON**. Choose **Create**.
5. Choose **CLOSE**. A JSON file containing the client email address and private key for the service account is saved on your computer.
6. Choose the service account and copy the unique ID of the account. You'll need this in Step 3: Enable API scopes.
7. From the top-left menu, choose **APIs & Services / Library**.
8. Search for **Admin SDK** and then choose it.
9. Choose **Enable**.
10. Search for **Google Drive API** and then choose it.
11. Choose **Enable**.

Step 3: Enable Google API scopes

1. [Log in](#) to a Google account that has administrator privileges.
2. From the top-left menu, choose **Security** and then choose **Settings**.
3. Scroll down and choose **API controls**.
4. Choose **MANAGE DOMAIN-WIDE DELEGATION** in the **Domain-wide delegation** section of the page.
5. Choose **Add New**. Paste the service account unique ID from step 2 in **Client ID**.
6. Add the following read-only scopes to **OAuth scopes**, and then choose **Authorize**.

```
https://www.googleapis.com/auth/drive.readonly,  
https://www.googleapis.com/auth/drive.metadata.readonly,  
https://www.googleapis.com/auth/admin.directory.user.readonly,  
https://www.googleapis.com/auth/admin.directory.group.readonly
```

Once you have created a service account and configured it to use the Google API, you can create a Google Drive data source.

Step 4: To create a Google Drive data source using the Amazon Kendra console

1. Sign into the AWS Management Console and open the Amazon Kendra console at <https://console.aws.amazon.com/kendra/home>.
2. From the list of indexes, choose the index that you want to add the data source to.
3. Choose **Add data sources**.
4. From the list of data source connectors, choose **Google Drive**.
5. On the **Name and description** page, give your data source a name and optionally a description. Leave the **Tags** field blank. Choose **Next** to continue.
6. Under **Type of authentication**, choose **New**. Enter the following information:
 - **Secret name** – the name of the Secrets Manager secret that holds the authentication information.
 - **Admin account email** – The email of the administrator that created the service account.
 - **Client email** – The service account email address. You can find this address in `client_email` field of the JSON file you downloaded in step 1.

- **Private key** – The private key from the JSON file that you downloaded in step 1.

- Choose **Save authentication** to create a new Secrets Manager secret.
7. Under **Define access and security** choose **Create a new role (Recommended)** to create an IAM role for your data source.
 8. Under **Role name** enter a name for the IAM role.
 9. Choose **Next**.
 10. On the **Configure sync settings** page, don't change the **Set sync scope** or **Additional configuration** sections. In the **Sync run schedule** section, choose **Run on demand** for the frequency.
 11. Choose **Next**.
 12. On the **Data source fields** page, leave the defaults.
 13. Choose **Next**.
 14. On the **Review and create** page, review the settings for your data source. Once you are satisfied, choose **Create**.

After you choose **Create**, Amazon Kendra returns you to the list of data sources and starts creating the data source. It can take several minutes for the data source to be created. When it is finished, the console displays a banner and the status of the data source changes from **Creating** to **Active**.

After creating the data source, you need to sync the Amazon Kendra index with the data source. Choose **Sync now** to start the sync process. It can take several minutes to several hours to synchronize the data source, depending on the number and size of the documents.

Getting started with a Microsoft OneDrive for Business data source (console)

You can use the Amazon Kendra console to get started indexing a Microsoft OneDrive for Business site. When you use the console you can specify all of the connection information that you need to index the contents of a OneDrive site. For more information, see [Using a Microsoft OneDrive data source \(p. 104\)](#).

Use the following procedure to create a basic OneDrive data source using the default configuration. The procedure assumes that you have already created an index following the steps in step 1 of [Getting started with an S3 bucket \(console\) \(p. 40\)](#).

Before you can create a OneDrive data source, you must register an Azure Active Directory application. Use the following procedure to register the application.

Step 1: To register an Azure AD application

1. Log into the [Azure Management Portal](#).
2. Choose **Azure Active Directory** and then choose **App registrations**
3. Choose **New registration**.
4. Create the application with the following values.
 - **Name** Enter the name for your application.
 - **Application type** **Web app/API**
 - **Sign-on URL** Any valid URL. The URL doesn't need to exist.
5. Choose **Register**.

6. Choose the application name in the list of applications to open the application settings.
7. Choose **Overview** and then copy the **Application ID** value.

The application requires permission to access OneDrive data. Use the following procedure to assign permissions to the application. This procedure starts where the previous procedure left off.

Step 2: To assign permissions to the Azure AD application

1. From the application settings, choose **API Permission**.
2. Choose **Add**, and choose the **Microsoft Graph** option, and then choose **Add**.
3. From the **Application permissions**, choose the following.
 - Read files in all site collections (Files.Read.All)
 - Read all users' full profiles (User.Read.All)
 - Read directory data (Directory.Read.All)
 - Read all groups (Group.Read.All)
 - Read items in all site collections (Sites.Read.All)
4. Choose **Save** and then choose **Grant permissions**. Choose **Yes** when prompted.

Next you create an application key that Amazon Kendra uses to identify itself with the OneDrive site. This procedure starts where the previous procedure left off.

Step 3: To create an application key

1. From the **Application settings**, choose **Keys**.
2. Add the following information to the key.
 - **Description** Add a description for your key.
 - **Expires** Choose a duration based on your company policy. When the application key is rotated, you must update the key in the Amazon Kendra data source.
3. Choose **Save** to generate the key value. Copy this value to use in the next step.

Once you have created the Azure AD application and generated a secret key for the application, you are ready to create a OneDrive data source.

Step 4: To create a OneDrive data source using the Amazon Kendra console

1. Sign into the AWS Management Console and open the Amazon Kendra console at <https://console.aws.amazon.com/kendra/home>.
2. From the list of indexes, choose the index that you want to add the data source to.
3. Choose **Add data sources**.
4. From the list of data source connectors, choose **OneDrive**.
5. On the **Define attributes** page, give your data source a name and optionally a description. Leave the **Tags** field blank. Choose **Next** to continue.
6. On the **Define targets** page, enter the domain name of your OneDrive site. Enter the name without the protocol ("https://").
7. In the **IAM role** field choose **Create a new role**. Enter a name for the role in the **Role name** field. Choose **Next** to continue.
8. In the **Type of authentication** field, choose **New**. This tells the console to create a new AWS Secrets Manager secret to contain credentials for your OneDrive data source.

9. In the **New secret container name** field enter a name to identify the Secrets Manager secret.
10. In the **Application ID** and **Application password** fields enter the ID and password for the Active Directory application that you created for the data source. Choose **Save authentication** to save the credentials for your OneDrive site.
11. In the **Add OneDrive users** section, choose **Names list**. Add the email addresses of up to 10 users whose documents you want to index.
12. In the **Set sync run schedule** section, choose **Run on demand** in the **Frequency** field.
13. In the **OneDrive field mapping** section, leave the default fields selected and then choose **Next**.
14. On the **Review and create** page review the details of your OneDrive data source. If you want to make changes, choose the **Edit** button next to the item that you want to change. When you are satisfied with your choices, choose **Create** to create your OneDrive data source.

After you choose **Create**, Amazon Kendra starts creating the data source. It can take several minutes for the data source to be created. When it is finished, the status of the data source changes from **Creating** to **Active**.

After creating the data source, you need to sync the Amazon Kendra index with the data source. Choose **Sync now** to start the sync process. It can take several minutes to several hours to synchronize the data source, depending on the number and size of the documents.

Getting started with an Amazon S3 data source (console)

You can use the Amazon Kendra console to get started using an Amazon S3 bucket as a data store. When you use the console you specify all of the connection information you need to index the contents of the bucket. For more information, see [Using an Amazon S3 data source \(p. 86\)](#).

Use the following procedure to create a basic S3 bucket data source using the default configuration. The procedure assumes that you created an index following the steps in step 1 of [Getting started with an S3 bucket \(console\) \(p. 40\)](#).

To create an S3 bucket data source using the Amazon Kendra console

1. Sign into the AWS Management Console and open the Amazon Kendra console at <https://console.aws.amazon.com/kendra/home>.
2. From the list of indexes, choose the index that you want to add the data source to.
3. Choose **Add data sources**.
4. From the list of data source connectors, choose **Amazon S3**.
5. On the **Define attributes** page, give your data source a name and optionally a description. Leave the **Tags** field blank. Choose **Next** to continue.
6. In the **Enter the data source location** field, enter the name of the S3 bucket that contains your documents. You can enter the name directly, or you can browse for the name by choosing **Browse**. The bucket must be in the same Region as the index.
7. In **IAM role** choose **Create a new role** and then type a role name.
8. In the **Set sync run schedule** section, choose **Run on demand**.
9. Choose **Next** to continue.
10. On the **Review and create** page review the details of your S3 data source. If you want to make changes, choose the **Edit** button next to the item that you want to change. When you are satisfied with your choices, choose **Create** to create your S3 data source.

After you choose **Create**, Amazon Kendra starts creating the data source. It can take several minutes for the data source to be created. When it is finished, the status of the data source changes from **Creating** to **Active**.

After creating the data source, you need to sync the Amazon Kendra index with the data source. Choose **Sync now** to start the sync process. It can take several minutes to several hours to synchronize the data source, depending on the number and size of the documents.

Getting started with a MySQL database data source (console)

You can use the Amazon Kendra console to get started using a MySQL database as a data source. When you use the console you specify the connection information you need to index the contents of a MySQL database. For more information, see [Using a database data source](#).

You first need to create a MySQL database, then you can create a data source for the database.

Use the following procedure to create a basic MySQL database. The procedure assumes that you have already created an index following step 1 of [Getting started with an S3 bucket \(console\) \(p. 40\)](#).

To create a MySQL database

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. From the navigation pane, choose **Subnet groups** and then choose **Create DB Subnet Group**.
3. Name the group and choose your Virtual Private Cloud (VPC). For more information on configuring a VPC, see [Configuring Amazon Kendra to use a VPC](#).
4. Add your VPC's private subnets. Your private subnets are the ones that are not connected to your NAT. Choose **Create**.
5. From the navigation pane, choose **Databases** and then choose **Create database**.
6. Use the following parameters to create the database. Leave all of the other parameters at their defaults.
 - **Engine options – MySQL**
 - **Templates – Free tier**
 - **Credential Settings** – Enter and confirm a password
 - Under **Connectivity**, choose **Additional connectivity configuration**. Make the following choices.
 - **Subnet group** – Choose the subnet group that you created in step 4.
 - **VPC security group** – Choose the group that contains both inbound and outbound rules that you created in your VPC. For example, **DataSourceSecurityGroup**. For more information on configuring a VPC, see [Configuring Amazon Kendra to use a VPC](#).
 - Under **Additional configuration**, set the **Initial database name** to **content**.
7. Choose **Create database**.
8. From the list of databases, choose your new database. Make a note of the database endpoint.
9. After you create your database, you must create a table to hold your documents. Creating a table is outside the scope of these instructions. When you create your table, note the following:
 - Database name – **content**
 - Table name – **documents**
 - Columns – **ID**, **Title**, **Body**, and **LastUpdate**. You can include additional columns if you want.

Now that you have created your MySQL database, you can create a data source for the database.

To create a MySQL data source

1. Sign in to the AWS Management Console and open the Amazon Kendra console at <https://console.aws.amazon.com/kendra/>.
2. From the navigation pane, choose **Indexes** and then choose your index.
3. Choose **Add data sources** and then choose **Amazon RDS**.
4. Type a name and description for the data source and then choose **Next**.
5. Choose **MySQL**.
6. Under **Connection access**, enter the following information:
 - **Endpoint** – The endpoint of the database that you created earlier.
 - **Port** – The port number for the database. For MySQL, the default is 3306.
 - **Type of authentication** – Choose **New**.
 - **New secret container name** – A name for the Secrets Manager container for the database credentials.
 - **Username** – The name of a user with administrative access to the database.
 - **Password** – The password for the user, and then choose **Save authentication**.
 - **Database name – content**.
 - **Table name – documents**.
 - **IAM role** – Choose **Create a new role**, and then type a name for the role.
7. In **Column configuration** enter the following:
 - **Document ID column name – ID**
 - **Document title column name – Title**
 - **Document data column name – Body**
8. In **Column change detection** enter the following:
 - **Change detecting columns – LastUpdate**
9. In **Configure VPC & security group** provide the following:
 - In **Virtual Private Cloud (VPC)**, choose your VPC.
 - In **Subnets**, choose the private subnets that you created in your VPC.
 - In **VPC security groups**, choose the security group that contains both inbound and outbound rules that you created in your VPC for MySQL databases. For example, **DataSourceSecurityGroup**.
10. In **Set sync run schedule**, choose **Run on demand** and then choose **Next**.
11. In **Data source field mapping**, choose **Next**.
12. Review the configuration of your data source to make sure that it is correct. When you're satisfied that everything is correct, choose **Create**.

Getting started with a Salesforce data source (console)

You can use the Amazon Kendra console to get started using a Salesforce data store. When you use the console you specify the connection information you need to index the contents of a Salesforce instance. For more information, see [Using a Salesforce data source \(p. 106\)](#).

Use the following procedure to create a basic Salesforce data source using the default configuration. The procedure assumes you created an index following the steps in step 1 of [Getting started with an S3 bucket \(console\) \(p. 40\)](#).

To create a Salesforce data source using the Amazon Kendra console

1. Sign into the AWS Management Console and open the Amazon Kendra console at <https://console.aws.amazon.com/kendra/home>.
2. From the list of indexes, choose the index that you want to add the data source to.
3. Choose **Add data sources**.
4. From the list of data source connectors, choose **Salesforce**.
5. On the **Define attributes** page, give your data source a name and optionally a description. Leave the **Tags** field blank. Choose **Next** to continue.
6. On the **Define targets** page, enter the URL of your Salesforce server.
7. In the **IAM role** field choose **Create a new role**. Enter a name for the role in the **Role name** field. Choose **Next** to continue.
8. In the **Type of authentication** field, choose **New**. This tells the console to create a new AWS Secrets Manager secret to contain credentials for your Salesforce data source.
9. In the **New secret container name** field enter a name to identify the Secrets Manager secret.
10. In the **Username** field, enter the user name for your Salesforce account.
11. In the **Password** field, enter the password for your Salesforce account.
12. In the **Security token** field, enter the security token for your Salesforce account.
13. In the **Consumer key** field, enter the consumer key of the Salesforce connected app that you are using.
14. In the **Consumer secret** field, enter the secret associated with the consumer key of the Salesforce connected app that you are using.
15. In the **Authentication URL** field, enter the OAuth endpoint for Salesforce.
16. Choose **Save authentication** to save your authentication information.
17. In the **Crawl settings** section, under **Standard objects**, choose **Document**.
18. In the **Set sync run schedule** section, choose **Run on demand**.
19. Choose **Next** to continue.
20. On the **Set field mappings - optional** page, leave the defaults and choose **Next**.
21. On the **Review and create** page review the settings for the data source. Use the **Edit** buttons to make any changes that you need to make. When you are satisfied with the settings, choose **Create** to create your Salesforce data source.

After you choose **Create**, Amazon Kendra starts creating the data source. It can take several minutes for the data source to be created. When it is finished, the status changes from **Creating to Active**.

After creating the index, you need to sync the Amazon Kendra index with the data source. Choose **Sync now** to start the sync process. It can take several minutes to several hours to synchronize the data source, depending on the number and size of the documents.

Getting started with a ServiceNow data source (console)

You can use the Amazon Kendra console to get started using a ServiceNow data store. When you use the console you specify the connection information you need to index the contents of a ServiceNow instance. For more information see [Using a ServiceNow data source \(p. 109\)](#).

Use the following procedure to create a basic ServiceNow data source using the default configuration. The procedure assumes you created an index following the steps in [Getting started with an S3 bucket \(console\) \(p. 40\)](#).

To create a ServiceNow data source using the Amazon Kendra console

1. Sign into the AWS Management Console and open the Amazon Kendra console at <https://console.aws.amazon.com/kendra/home>.
2. From the list of indexes, choose the index that you want to add the data source to.
3. Choose **Add data sources**.
4. From the list of data source connectors, choose **ServiceNow**.
5. On the **Define attributes** page, give your data source a name and optionally a description. Leave the **Tags** field blank. Choose **Next** to continue.
6. For the **ServiceNow host**, enter the URL of your ServiceNow instances.
7. From the **ServiceNow version** dropdown, choose the ServiceNow version that your organization uses. You can choose **London** or all other versions.
8. In the **IAM role** field choose **Create a new role**. Enter a name for the role in the **Role name** field. Choose **Next** to continue.
9. In the **Type of authentication** field, choose **New**. This tells the console to create a new AWS Secrets Manager secret to contain credentials for your ServiceNow data source.
10. In the **New secret container name** field enter a name to identify the Secrets Manager secret.
11. Enter the username and password for your ServiceNow account, and then choose **Save authentication** to save the new secret.
12. In the **ServiceNow configuration** section leave the defaults selected.
13. In the **Set sync run schedule** section, choose **Run on demand** in the **Frequency** field.
14. Choose **Next** to continue.
15. On the **Set field mappings - optional** page, leave the defaults checked and choose **Next** to continue.
16. On the **Review and create** page review the details of your ServiceNow data source. If you want to make changes, choose the **Edit** button next to the item that you want to change. When you are satisfied with your choices, choose **Create** to create your ServiceNow data source.

After you choose **Create**, Amazon Kendra starts creating the data source. It can take several minutes for the data source to be created. When it is finished, the status of the data source changes from **Creating** to **Active**.

After creating the data source, you need to sync the Amazon Kendra index with the data source. Choose **Sync now** to start the sync process. It can take several minutes to several hours to synchronize the data source, depending on the number and size of the documents.

Getting started with a Microsoft SharePoint data source (console)

You can use the Amazon Kendra console to get started indexing a Microsoft SharePoint data source. When you use the console you specify the connection information you need to index the contents of the SharePoint data source.

Use the following procedure to create a basic SharePoint data source using the default configuration. The procedure assumes you created an index following step 1 of [Getting started with an S3 bucket \(console\) \(p. 40\)](#).

To create a SharePoint data source using the Amazon Kendra console

1. Sign into the AWS Management Console and open the Amazon Kendra console at <https://console.aws.amazon.com/kendra/home>.
 2. From the list of indexes, choose the index that you want to add the data source to.
 3. Choose **Add data sources**.
 4. From the list of data source connectors, choose **SharePoint**.
 5. On the **Specify data source details** page, do the following:
 - a. In the **Name data source** section, give your data source a name and optionally a description.
 - b. (Optional) In the **Tags** section, add tags to categorize your data source.
 - c. Choose **Next**.
 6. On the **Define access and security** page, do the following:
 - a. In the **Source** section, do one of the following:
 - Choose **SharePoint Online** to connect to your SharePoint cloud account and add the URLs of your SharePoint data repository to index.
 - Choose **SharePoint Server** to connect to your SharePoint hosted on your own server and choose your SharePoint version. Then add the URLs of your SharePoint data repository to index. (Optional) If using your own private certificate authority, you need to provide the SSL certificate location and the certificate needs to be in X.509 standard format. Amazon Kendra supports most public certificate authorities. You also need to provide permission to access the Amazon S3 bucket that contains the SSL certificate used to communicate with the SharePoint site.
 - b. In the **Authentication** section, do the following:
 - If you chose **SharePoint Online** as your source, you can select the **AWS Secrets Manager secret** dropdown to create a secret and enter your SharePoint credentials, or choose an existing secret. Your secret must contain your SharePoint user name and password, and a name for your secret. For more information on a secret that stores your authentication credentials, see [AWS Secrets Manager](#).
 - If you chose **SharePoint Server** as your source, you can select the **AWS Secrets Manager secret** dropdown to create a secret and enter your SharePoint credentials, or choose an existing secret. In addition to your SharePoint user name and password, and a name for your secret, your secret must include the server domain name. The server domain name is the NetBIOS name in your active directory provider.
 - If you want to filter on user context and you need to convert access control list (ACL) to email format, you choose whether to provide the LDAP server URL and LDAP search base, or manually enter the email domain. You can choose to keep ACL as `sAMAccountName` if you do not want to provide an email domain. `sAMAccountName` is the server domain name and the user name. For more information on LDAP server URL and LDAP search base, and manual email domain override of these, see [Using a SharePoint data source](#).
 - c. (Optional) If you chose **SharePoint Server**, in the **VPC and security group** section, do the following:
 - If you want Amazon Kendra to connect to your Amazon virtual private cloud (VPC) to index documents stored in SharePoint running in your private cloud, choose a **Virtual Private Cloud (VPC)**. You need to provide the required permissions for VPC - see [IAM role for VPC](#). To configure Amazon Kendra to use a VPC, see [Configuring a VPC](#).
 - d. In the **IAM role** section, in **IAM role**, choose an existing role that grants permission to your SharePoint and other resources. For more information about the required permissions, see [IAM access roles for Amazon Kendra](#).
 - e. Choose **Next**.
7. On the **Configure sync settings** page, do the following:

- a. (Optional) In the **Sync scope** section, do one of the following:
 - Choose **Use change logs** to sync based on identifying updated documents instead of scanning all documents.
 - Choose **Crawl document attachments** to include attachments when syncing.
 - Choose **Use local group mappings** if your SharePoint server is not in sync with your local active directory provider and you want to filter your documents. You need to provide a hash of each site name when you query the index.
 - b. Choose **Additional configuration** to use regular expression patterns to include or exclude certain files to index.
 - c. In the **Sync schedule** section, for **Frequency**, choose the frequency to sync your index with your SharePoint data source. You can sync hourly, daily, weekly, monthly, run on demand, or you can choose your own custom sync schedule.
 - d. Choose **Next**.
8. (Optional) On the **Set field mappings** page, check your SharePoint fields are mapped to Amazon Kendra index fields, such as the document ID, document title, and more. You can also add a custom field by selecting **Add field** or you can add fields directly using the API. Choose **Next**.
 9. On the **Review and Create** page, review the details of your SharePoint data source. To make changes, choose the **Edit** button next to the item that you want to change. When you are done, choose **Add data source** to add your SharePoint data source.

After you choose **Create**, Amazon Kendra starts creating the data source. It can take several minutes for the data source to be created. When it is finished, the status of the data source changes from **Creating** to **Active**.

Amazon Kendra syncs the index with SharePoint in accordance with the sync schedule you set. If you choose **Sync now** to start the sync process immediately, it can take several minutes to a few hours to synchronize, depending on the number and size of the documents.

Getting started with Amazon Kendra Web Crawler (console)

You can use the Amazon Kendra console to get started using the Amazon Kendra *Web Crawler*. When you use the console, you specify the connection information you need to index the contents of the webpages crawled using the web crawler. For more information, see [Using a web crawler data source](#).

If you want to use a web proxy server to connect to and crawl websites, you need to provide the website host name and port number. Web proxy credentials (stored in a secret in [AWS Secrets Manager](#)) are optional and you can use them to connect to a web proxy server that requires basic authentication.

If you want to use basic authentication of user name and password to access and crawl websites, you need to provide the website host name, port number, and your secret in [AWS Secrets Manager](#) that stores your authentication credentials.

The following procedure assumes that you created an index following step 1 of [Getting started with an S3 bucket \(Console\)](#).

To create Amazon Kendra Web Crawler as a data source connector (console)

1. Sign into the AWS Management Console and then open the Amazon Kendra console at <https://console.aws.amazon.com/kendra/home>.
2. From the list of indexes, choose the index that you want to add the data source to.

3. Choose **Add data sources**.
 4. From the list of data source connectors, choose **WebCrawler**.
 5. On the **Specify data source details** page, do the following:
 - a. In the **Name data source** section, give your data source a name and optionally a description.
 - b. (Optional) In the **Tags** section, add tags to categorize your data source.
 - c. Choose **Next**.
 6. On the **Define access and security page**, do the following:
 - a. In the **Source** section, do one of the following:
 - Choose **Source URLs** to enter the seed URLs of the website domains you want to crawl. Enter the seed or starting point URL, and then choose **Add new URL**. You can also add website subdomains. You can add up to ten seed URLs.
 - Choose **Source Sitemaps** to enter the website sitemap URLs you want to crawl. A sitemap includes all relevant webpages or website domains you want to crawl. Enter the sitemap URL, and then choose **Add new URL**. You can add up to three sitemap URLs.
 - Note**

You can only crawl websites that use the secure communication protocol, Hypertext Transfer Protocol Secure (HTTPS). If you receive a validation exception error when trying to crawl a website, it could be due to the website being blocked from crawling.
 - b. (Optional) In the **Web proxy** section, do the following:
 - i. In **Host name** and **Port number**, enter the website host name and port number. For example, the host name of <https://a.example.com/page1.html> is "a.example.com" and the port is 443, the standard port for HTTPS.
 - ii. Under **Web proxy credentials**, to use web proxy credentials to connect to a web proxy server that requires basic authentication, choose AWS Secrets Manager secret. For more information, see [AWS Secrets Manager](#)
 - c. (Optional) In the **Hosts with authentication** section, to connect to websites that require user authentication, choose **Add additional host with authentication**.
 - d. In the **IAM role** section, in **IAM role**, choose an existing role that grants Amazon Kendra permission to access the web crawler resources such as your index. For more information about the required permissions, see [IAM access roles for Amazon Kendra](#).
 - e. Choose **Next**.
7. On the **Configure sync settings** page, do the following:
 - a. If you selected **Source URLs** in step 6a, in **Crawl range**, do one of the following:
 - Keep **Crawl host domains only**.
 - To include subdomains, choose **Crawl host domains and their subdomains only**
 - To include subdomains and other domains the webpages link to, choose **Crawl everything**.
 - b. In **Crawl depth**, set the depth to the number of levels in a website from the seed level that you want to crawl. For example, if a website has three levels – index level or seed level in this example, sections level, and subsections level – and you are only interested in crawling information up to the sections level (levels 0 to 1), set your depth to **1**.
 - c. Choose **Advanced crawl settings** to set the maximum size (in MB) of a webpage to crawl, the maximum number of URLs on a single webpage to also crawl, and the maximum number of URLs crawled per website host per minute.
 - d. Choose **Additional configuration** to use regular expression patterns to include or exclude certain URLs to crawl.
 - e. In the **Sync schedule** section, for **Frequency**, choose the frequency to sync your index with the web crawler data source. You can sync hourly, daily, weekly, monthly, run on demand, or you can choose your own custom sync schedule.

- f. Choose **Next**.
8. On the **Review and Create** page, review the details of your web crawler data source. To make changes, choose the **Edit** button next to the item that you want to change. When you are done, choose **Add data source** to add the web crawler data source.

After you choose **Add data source**, Amazon Kendra starts web crawling. It can take several minutes to a few hours for the web crawling to complete, depending on the number and size of the webpages to crawl. When it is finished, the status changes from **Creating** to **Active**.

Amazon Kendra syncs the index with the web crawler in accordance with the sync schedule you set. If you choose **Sync now** to start the sync process immediately, it can take several minutes to a few hours to synchronize, depending on the number and size of the documents.

When selecting websites to index, you must adhere to the [Amazon Acceptable Use Policy](#) and all other Amazon terms. Remember that you must only use Amazon Kendra Web Crawler to index your own webpages, or webpages that you have authorization to index. To learn how to stop Amazon Kendra Web Crawler from indexing your website(s), please see [Stopping Amazon Kendra Web Crawler from indexing your website](#).

Getting started with an Amazon WorkDocs data source (console)

You can use the Amazon Kendra console to get started indexing an Amazon WorkDocs data source. When you use the console, you specify the connection information to index the contents of the Amazon WorkDocs data source.

Amazon WorkDocs connector is available in Oregon, North Virginia, Sydney, Singapore and Ireland regions.

Use the following procedure to create an Amazon WorkDocs data source. The procedure assumes you created an index following step 1 of [Getting started with an S3 bucket \(console\) \(p. 40\)](#).

To create an Amazon WorkDocs data source using the Amazon Kendra console

1. Sign into the AWS Management Console and open the Amazon Kendra console at <https://console.aws.amazon.com/kendra/home>.
2. From the list of indexes, choose the index that you want to add the data source to.
3. Choose **Add data sources**.
4. From the list of data source connectors, choose **Amazon WorkDocs**.
5. On the **Specify data source details** page, do the following:
 - a. In the **Name data source** section, give your data source a name and optionally a description.
 - b. (Optional) In the **Tags** section, add tags to categorize your data source.
 - c. Choose **Next**.
6. On the **Define access and security** page:
 - a. In the **Source** section, do the following:
 - Choose the organization ID, which is the directory ID corresponding to your Amazon WorkDocs site. To locate your organization ID, go to the [AWS Directory Service console](#) and select the **Active Directory** dropdown menu. In your list of **Directories**, your Amazon WorkDocs site directory has an ID, which is the organization ID. You can set up a new

directory for your Amazon WorkDocs site in the AWS Directory Service console and enable the directory in the Amazon WorkDocs console.

- b. In the **IAM role** section, do the following:
 - Choose an existing **IAM role** that grants Amazon Kendra permission to access your Amazon WorkDocs resources such as your index. For more information about the required permissions, see [IAM access roles for Amazon Kendra](#).
 - c. Choose **Next**.
7. On the **Configure sync settings** page:
- a. (Optional) In the **Sync scope** section, do any of the following:
 - Select the checkbox **Crawl document comments** to include comments when syncing. Each comment is indexed as a separate document.
 - Select the checkbox **Use change logs** to sync based on identifying updated documents instead of scanning all documents. If you are syncing your Amazon WorkDocs data source with your index for the first time, all documents are scanned.
 - Select the **Advanced configuration** to use regular expression patterns to include or exclude files in your data source.
 - b. In the **Sync schedule** section, for **Frequency**, you can sync your index with your Amazon WorkDocs data source. You can sync hourly, daily, weekly, monthly, run on demand, or you can choose your own custom sync schedule.
 - c. Choose **Next**.
8. (Optional) On the **Set field mappings** page, check that your Amazon WorkDocs fields are mapped to Amazon Kendra index fields, such as the document ID, document authors, and more. You can also add a custom field by selecting **Add field**. The custom field name must exist in the metadata of your Amazon WorkDocs documents. You can also add fields directly using the API. Choose **Next**.
9. On the **Review and Create** page, review the details of your Amazon WorkDocs data source. To make changes, choose the **Edit** button next to the item that you want to change. When you are done, choose **Add data source** to add your Amazon WorkDocs data source.

After you choose **Add data source**, Amazon Kendra starts creating the data source. It can take several minutes for the data source to be created. When it is finished, the status of the data source changes from **Creating** to **Active**.

Amazon Kendra syncs the index with Amazon WorkDocs based on the sync schedule you set. If you choose **Sync now**, it still can take several minutes to a few hours to synchronize, depending on the number and size of documents.

Getting started with an AWS Single Sign-On identity source (console)

An AWS Single Sign-On identity source contains information on access levels of groups and users. This is useful for setting up user context filtering, where Amazon Kendra filters search results for different users based on their group's access to documents.

To create an AWS SSO identity source, you must enable AWS SSO and create an organization. When you enable AWS SSO and create an organization for the first time, it automatically creates AWS SSO identity store as the default identity source. You can change to Active Directory (Amazon managed or self-managed) or an external identity provider as your identity source. You must follow the correct guidance for this – see [Changing your AWS SSO identity source](#). You can have only one identity source per organization.

In order for your groups in AWS SSO to be assigned different levels of access to documents, you need to include your groups in your Access Control List when you ingest documents into your index. This allows your groups to search for documents in Amazon Kendra in accordance with their level of access. When you issue a query, the user ID needs to be an exact match of the user name in AWS SSO.

You must also grant the required permissions to use AWS SSO with Amazon Kendra. For more information, see [IAM roles for AWS Single Sign-On](#).

To set up an AWS SSO identity source

1. Open the [AWS SSO console](#).
2. Choose **Enable AWS SSO**, and then choose **Create AWS organization**.
AWS SSO identity store is created by default, and an email is sent to you to verify the email address associated with the organization.
3. To add a group to your AWS organization, in the navigation pane, choose **Groups**.
4. On the **Groups page**, choose **Create group** and enter a group name and description in the dialog box. Choose **Create**.
5. To add a user to your AWS SSO organization, in the navigation pane, choose **Users**.
6. On the **Users page**, choose **Add user**. Under **User details**, specify all required fields. For **Password**, choose **Send an email to the user**. Choose **Next**.
7. To add a user to a group, choose **Groups** and select a group.
8. On the **Details** page, under **Group members**, choose **Add user**.
9. On the **Add users to group** page, select the user you want to add as a member of the group. You can select multiple users to add to a group.
10. To sync your list of users and groups with AWS SSO, change your identity source to Active Directory or External identity provider.

AWS SSO identity store is the default identity source and requires you to manually add your users and groups using this source if you do not have your own list managed by a provider. To change your identity source, you must follow the correct guidance for this – see [Changing your AWS SSO identity source](#).

Note

If using Active Directory or an external identity provider as your identity source, you must map the email addresses of your users to AWS SSO user names when you specify the System for Cross-domain Identity Management (SCIM) protocol. For more information, see the [AWS SSO guide on SCIM for configuring AWS SSO](#).

Once you have set up your AWS SSO identity source, you can enable this in the console when you create or edit your index. Go to the **Configure user access control page** and under **User-group expansion**, choose **AWS SSO**. You can also enable AWS SSO using the **UserGroupResolutionConfiguration** structure. You provide the **UserGroupResolutionMode** as **AWS_SSO** and create an IAM role that gives permission to call **sso:ListDirectoryAssociations**, **sso-directory:SearchUsers**, **sso-directory>ListGroupsForUser**, **sso-directory:DescribeGroups**.

Changing your AWS SSO identity source

Warning

Changing your identity source in AWS SSO **Settings** might affect the preservation of user and group information. To do this safely, it is recommended you review [Considerations for changing your identity source](#). When you change your identity source, a new identity source ID is generated. Make sure you use the correct ID when you enable AWS SSO to integrate with Amazon Kendra.

To change your AWS SSO identity source

1. Open the [AWS SSO console](#).
2. Choose **Settings**.
3. On the **Settings** page, under **Identity source**, choose **Change**.
4. On the **Change identity source** page, select your preferred identity source, and then choose **Next**.

Creating an index

You can create an index using the console, the AWS Command Line Interface (AWS CLI), or by calling the [CreateIndex \(p. 296\)](#) API operation. The following procedures show how to create an index. After you have created your index, you can add documents directly to your index or you can add them from a data source.

To create an index, you must provide the Amazon Resource Name (ARN) of an AWS Identity and Access Management (IAM) role that has permissions to any Amazon Simple Storage Service (Amazon S3) bucket that you use and to perform actions on your behalf.

To create an index (console)

1. Sign into the AWS Management Console and open the Amazon Kendra console at <https://console.aws.amazon.com/kendra/>.
2. In **Specify index details**, give your index a name and a description.
3. In **IAM role** provide an IAM role. You can either choose from roles in your account that contain the word "kendra" or you can type the name of another role. For more information about the permissions that the role requires, see [IAM roles for indexes \(p. 12\)](#).
4. Choose **Next**.
5. On the **Configure user access control** page, choose **Next**. You can update your index to use tokens for access control after you create an index. For more information, see [Controlling access to documents in an index \(p. 66\)](#).
6. On the **Provisioning details** page, choose **Create index**.
7. Creating an index can take some time. Check the list of indexes to watch the progress of creating your index. When the status of the index is **ACTIVE**, your index is ready to use.

To create an index (AWS CLI)

1. Use the following command to create an index. The `role-arn` should be the Amazon Resource Name (ARN) of a role that can run Amazon Kendra actions. For more information, see [IAM access roles for Amazon Kendra \(p. 12\)](#).

```
aws kendra create-index \
--name index name \
--description "index description" \
--role-arn arn:aws:iam::account ID:role/role name
```

2. Creating an index can take some time. To check the state of your index, use the `index ID` returned by `create-index` with the following command. When the status of the index is **ACTIVE**, your index is ready to use.

```
aws kendra describe-index \
--index-id index ID
```

To create an index (SDK)

1. You need to provide values for the following variables:
 - `description` – A description of the index that you are creating.
 - `index_name` – The name of the index that you are creating.

- **role_arn** – The Amazon Resource Name (ARN) of a role that can run Amazon Kendra operations. For more information, see [IAM access roles for Amazon Kendra \(p. 12\)](#).
2. The following examples create an index with Amazon Kendra.

Python

```

import boto3
from botocore.exceptions import ClientError
import pprint
import time

kendra = boto3.client("kendra")

print("Create an index")

description = "index description"
index_name = "index-name"
role_arn = "arn:aws:iam::${account id}:role/${role name}"

try:
    index_response = kendra.create_index(
        Description = description,
        Name = index_name,
        RoleArn = role_arn
    )

    pprint.pprint(index_response)

    index_id = index_response["Id"]

    print("Wait for Kendra to create the index.")

    while True:
        # Get index description
        index_description = kendra.describe_index(
            Id = index_id
        )
        # If status is not CREATING quit
        status = index_description["Status"]
        print("    Creating index. Status: "+status)
        if status != "CREATING":
            break
        time.sleep(60)

except ClientError as e:
    print("%s" % e)

print("Program ends.")

```

Java

```

package com.amazonaws.kendra;

import java.util.concurrent.TimeUnit;
import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.CreateIndexRequest;
import software.amazon.awssdk.services.kendra.model.CreateIndexResponse;
import software.amazon.awssdk.services.kendra.model.DescribeIndexRequest;
import software.amazon.awssdk.services.kendra.model.DescribeIndexResponse;
import software.amazon.awssdk.services.kendra.model.IndexStatus;

public class CreateIndexExample {

```

```
public static void main(String[] args) throws InterruptedException {

    String indexDescription = "Getting started index for Kendra";
    String indexName = "java-getting-started-index";
    String indexRoleArn = "arn:aws:iam::<your AWS account ID>:role/
KendraRoleForGettingStartedIndex";

    System.out.println(String.format("Creating an index named %s", indexName));
    CreateIndexRequest createIndexRequest = CreateIndexRequest
        .builder()
        .description(indexDescription)
        .name(indexName)
        .roleArn(indexRoleArn)
        .build();
    KendraClient kendra = KendraClient.builder().build();
    CreateIndexResponse createIndexResponse =
kendra.createIndex(createIndexRequest);
    System.out.println(String.format("Index response %s",
createIndexResponse));

    String indexId = createIndexResponse.id();

    System.out.println(String.format("Waiting until the index with ID %s is
created.", indexId));
    while (true) {
        DescribeIndexRequest describeIndexRequest =
DescribeIndexRequest.builder().id(indexId).build();
        DescribeIndexResponse describeIndexResponse =
kendra.describeIndex(describeIndexRequest);
        IndexStatus status = describeIndexResponse.status();
        if (status != IndexStatus.CREATING) {
            break;
        }
        TimeUnit.SECONDS.sleep(60);
    }
    System.out.println("Index creation is complete.");
}
}
```

Once you have created your index, you add documents to it. You can either add them directly or you can create a data source that automatically updates your index on a regular schedule.

Topics

- [Controlling access to documents in an index \(p. 66\)](#)
- [Adding documents directly to an index \(p. 74\)](#)
- [Adding questions and answers directly to an index \(p. 80\)](#)
- [Adding documents from a data source \(p. 85\)](#)
- [Deleting data sources \(p. 119\)](#)
- [Creating custom document attributes \(p. 121\)](#)
- [Mapping data source fields \(p. 123\)](#)
- [Configuring Amazon Kendra to use a VPC \(p. 125\)](#)

Controlling access to documents in an index

Amazon Kendra supports token-based user access control using the following token types:

- Open ID
- JWT with a shared secret
- JWT with a public key
- JSON

Amazon Kendra delivers highly secure enterprise search for your search applications. Your search results reflect the security model of your organization. Customers are responsible for authenticating and authorizing users to gain access to their search application. At search time, the Kendra service filters search results based on user ID provided by the customer's search application, and document ACLs collected by the Kendra connectors during crawl/indexing time. The search results return URLs pointing back to the original document repositories plus short excerpts. Access to the full document is still enforced by the original repository.

Topics

- [Using OpenID \(p. 67\)](#)
- [Using a JSON Web Token \(JWT\) with a shared secret \(p. 68\)](#)
- [Using a JSON Web Token \(JWT\) with a public key \(p. 71\)](#)
- [Using JSON \(p. 73\)](#)

Using OpenID

To configure a Amazon Kendra index to use an OpenID token for access control, you need the jwks url from the OpenID provider. In most cases the jwks url would be in the following format (if they are following openid discovery) <https://domain-name/.well-known/jwks.json>.

The following examples show how to use an OpenID token for user access control when you are create an index.

Console

1. Choose **Create index** to start creating a new index.
2. On the **Specify index details** page, give your index a name and a description.
3. For **IAM role**, select a role or select **Create a new role** to and specify a role name to create a new role. The IAM role will have the prefix "AmazonKendra-".
4. Leave all of the other fields at their defaults. Choose **Next**.
5. In the **Configure user access control** page, under **Access control settings**, choose **Yes** to use tokens for access control.
6. Under **Token configuration**, select **OpenID** as the **Token type**.
7. Specify a **Signing key URL**. The URL should point to a set of JSON web keys.
8. *Optional* Under **Advanced configuration**:
 - a. Specify a **Username** to use in the ACL check.
 - b. Specify one or more **Groups** to use in the ACL check.
 - c. Specify the **Issuer** that will validate the token issuer.
 - d. Specify the **Client Id(s)**. You must specify a regular expression that match the audience in the JWT.
9. In the **Provisioning details** page, choose **Developer edition**.
10. Choose **Create** to create your index.
11. Wait for your index to be created. Kendra provisions the hardware for your index. This operation can take some time.

CLI

To create an index with the AWS CLI using a JSON input file, first create a JSON file with your desired parameters:

```
{  
    "Name": "user-context",  
    "Edition": "ENTERPRISE_EDITION",  
    "RoleArn": "arn:aws:iam::account id:role:/my-role",  
    "UserTokenConfigurationList": [  
        {  
            "JwtTokenTypeConfiguration": {  
                "KeyLocation": "URL",  
                "Issuer": "optional: specify the issuer url",  
                "ClaimRegex": "optional: regex to validate claims in the token",  
                "UserNameAttributeField": "optional: user",  
                "GroupAttributeField": "optional: group",  
                "URL": "https://example.com/.well-known/jwks.json"  
            }  
        }  
    ],  
    "UserContextPolicy": "USER_TOKEN"  
}
```

You can override the default user and group field names. The default value for `UserNameAttributeField` is "user". The default value for `GroupAttributeField` is "groups".

Next, call `create-index` using the input file. For example, if the name of your JSON file is `create-index-openid.json`, you can use the following:

```
aws kendra create-index --cli-input-json file://create-index-openid.json
```

Python

```
response = client.create_index(  
    Name='user-context',  
    Edition='ENTERPRISE_EDITION',  
    RoleArn='arn:aws:iam::account id:role:/my-role',  
    UserTokenConfigurationList=[  
        {  
            "JwtTokenTypeConfiguration": {  
                "KeyLocation": "URL",  
                "Issuer": "optional: specify the issuer url",  
                "ClaimRegex": "optional: regex to validate claims in the token",  
                "UserNameAttributeField": "optional: user",  
                "GroupAttributeField": "optional: group",  
                "URL": "https://example.com/.well-known/jwks.json"  
            }  
        }  
    ],  
    UserContextPolicy='USER_TOKEN'  
)
```

Using a JSON Web Token (JWT) with a shared secret

The following examples show how to use a JSON Web Token (JWT) with a shared secret token for user access control when you are create an index.

Console

1. Choose **Create index** to start creating a new index.
2. On the **Specify index details** page, give your index a name and a description.
3. For **IAM role**, select a role or select **Create a new role** to and specify a role name to create a new role. The IAM role will have the prefix "AmazonKendra-".
4. Leave all of the other fields at their defaults. Choose **Next**.
5. In the **Configure user access control** page, under **Access control settings**, choose **Yes** to use tokens for access control.
6. Under **Token configuration**, select **JWT with shared secret** as the **Token type**.
7. Under **Parameters for signing public key**, choose the **Type of secret**. You can use an existing AWS Secrets Manager shared secret or create a new shared secret.

To create a new shared secret, choose **New** and then follow these steps:

- a. Under **New AWS Secrets Manager secret**, specify a **Secret name**. The prefix AmazonKendra- will be added when you save the public key.
 - b. Specify a **Key ID**. The key id is a hint that indicates which key was used to secure the JSON web signature of the token.
 - c. Choose the signing **Algorithm** for the token. This is the cryptographic algorithm used to secure the ID token. For more information on RSA, see [RSA Cryptography](#).
 - d. Specify a **Shared secret**. Select **Generate secret** to have a secret generated for you.
 - e. *Optional* Specify when the shared secret is valid. You can specify the date and time a secret is valid from, valid to, or both. The secret will be valid in the interval specified.
 - f. Select **Save secret** to save the new secret.
8. *Optional* Under **Advanced configuration**:
 - a. Specify a **Username** to use in the ACL check.
 - b. Specify one or more **Groups** to use in the ACL check.
 - c. Specify the **Issuer** that will validate the token issuer.
 - d. Specify the **Client Id(s)**. You must specify a regular expression that matches the audience in the JWT.
 9. In the **Provisioning details** page, choose **Developer edition**.
 10. Choose **Create** to create your index.
 11. Wait for your index to be created. Kendra provisions the hardware for your index. This operation can take some time.

CLI

You can use JWT with a shared key token inside of a AWS Secrets Manager. You need the Secrets Manager ARN, and your Amazon Kendra role must have access to `GetSecretValue` on the Secrets Manager resource. If you are encrypting the Secrets Manager resource with AWS KMS, the role must also have access to the `decrypt` action.

To create an index with the AWS CLI using a JSON input file, first create a JSON file with your desired parameters:

```
{  
    "Name": "user-context",  
    "Edition": "ENTERPRISE_EDITION",  
    "RoleArn": "arn:aws:iam::account id:role:/my-role",  
    "UserTokenConfigurationList": [  
        {
```

```

    "JwtTokenTypeConfiguration": {
        "KeyLocation": "SECRET_MANAGER",
        "Issuer": "optional: specify the issuer url",
        "ClaimRegex": "optional: regex to validate claims in the token",
        "UserNameAttributeField": "optional: user",
        "GroupAttributeField": "optional: group",
        "SecretManagerArn": "arn:aws:secretsmanager:us-west-2:account
id:secret:/my-user-context-secret"
    }
},
"UserContextPolicy": "USER_TOKEN"
}

```

You can override the default user and group field names. The default value for `UserNameAttributeField` is "user". The default value for `GroupAttributeField` is "groups".

Next, call `create-index` using the input file. For example, if the name of your JSON file is `create-index-openid.json`, you can use the following:

```
aws kendra create-index --cli-input-json file://create-index-openid.json
```

The secret must have the following format in AWS Secrets Manager:

```

{
  "keys": [
    {
      "kid": "key_id",
      "alg": "HS256|HS384|HS512",
      "kty": "OCT",
      "use": "sig", //this value can be sig only for now
      "k": "secret",
      "nbf": "ISO1806 date format"
      "exp": "ISO1806 date format"
    }
  ]
}

```

For more information about JWT, see [jwt.io](#).

Python

```

response = client.create_index(
    Name='user-context',
    Edition='ENTERPRISE_EDITION',
    RoleArn='arn:aws:iam::account id:role:/my-role',
    UserTokenConfigurationList=[
        {
            "JwtTokenTypeConfiguration": {
                "KeyLocation": "URL",
                "Issuer": "optional: specify the issuer url",
                "ClaimRegex": "optional: regex to validate claims in the token",
                "UserNameAttributeField": "optional: user",
                "GroupAttributeField": "optional: group",
                "SecretManagerArn": "arn:aws:secretsmanager:us-west-2:account
id:secret:/my-user-context-secret"
            }
        },
        UserContextPolicy='USER_TOKEN'
)

```

Using a JSON Web Token (JWT) with a public key

The following examples show how to use a JSON Web Token (JWT) with certificate token for user access control when you are create an index. For more information about JWT, see [jwt.io](#).

Console

1. Choose **Create index** to start creating a new index.
2. On the **Specify index details** page, give your index a name and a description.
3. For **IAM role**, select a role or select **Create a new role** to and specify a role name to create a new role. The IAM role will have the prefix "AmazonKendra-".
4. Leave all of the other fields at their defaults. Choose **Next**.
5. In the **Configure user access control** page, under **Access control settings**, choose **Yes** to use tokens for access control.
6. Under **Token configuration**, select **JWT with public key** as the **Token type**.
7. Under **Parameters for signing public key**, choose the **Type of secret**. You can use an existing AWS Secrets Manager secret or create a new secret.

To create a new secret, choose **New** and then follow these steps:

- a. Under **New AWS Secrets Manager secret**, specify a **Secret name**. The prefix **AmazonKendra-** will added when you save the public key.
 - b. Specify a **Key ID**. The key id is a hint that indicates which key was used to secure the JSON web signature of the token.
 - c. Choose the signing **Algorithm** for the token. This is the cryptographic algorithm used to secure the ID token. For more information on RSA, see [RSA Cryptography](#).
 - d. Under **Certificate attributes**, specify an *optional* **Certificate chain**. The certificate chain is made up of a list of certificates. It begins with a server's certificate and terminates with the root certificate.
 - e. *Optional* Specify the **Thumbprint or fingerprint**. It should be is a hash of a certificate, computed over all certificate data and its signature.
 - f. Specify the **Exponent**. This is the exponent value for the RSA public key. It is represented as a Base64urlUInt-encoded value.
 - g. Specify the **Modulus**. This is the exponent value for the RSA public key. It is represented as a Base64urlUInt-encoded value.
 - h. Select **Save key** to save the new key.
8. *Optional* Under **Advanced configuration**:
 - a. Specify a **Username** to use in the ACL check.
 - b. Specify one or more **Groups** to use in the ACL check.
 - c. Specify the **Issuer** that will validate the token issuer.
 - d. Specify the **Client Id(s)**. You must specify a regular expression that match the audience in the JWT.
 9. In the **Provisioning details** page, choose **Developer edition**.
 10. Choose **Create** to create your index.
 11. Wait for your index to be created. Kendra provisions the hardware for your index. This operation can take some time.

CLI

You can use JWT with a public key inside of a AWS Secrets Manager. You need the Secrets Manager ARN, and your Amazon Kendra role must have access to `GetSecretValue` on the Secrets Manager

resource. If you are encrypting the Secrets Manager resource with AWS KMS, the role must also have access to the decrypt action.

To create an index with the AWS CLI using a JSON input file, first create a JSON file with your desired parameters:

```
{
    "Name": "user-context",
    "Edition": "ENTERPRISE_EDITION",
    "RoleArn": "arn:aws:iam::account id:role:/my-role",
    "UserTokenConfigurationList": [
        {
            "JwtTokenTypeConfiguration": {
                "KeyLocation": "SECRET_MANAGER",
                "Issuer": "optional: specify the issuer url",
                "ClaimRegex": "optional: regex to validate claims in the token",
                "UserNameAttributeField": "optional: user",
                "GroupAttributeField": "optional: group",
                "SecretManagerArn": "arn:aws:secretsmanager:us-west-2:account
id:secret:/my-user-context-secret"
            }
        },
        "UserContextPolicy": "USER_TOKEN"
    ]
}
```

You can override the default user and group field names. The default value for `UserNameAttributeField` is "user". The default value for `GroupAttributeField` is "groups".

Next, call `create-index` using the input file. For example, if the name of your JSON file is `create-index-openid.json`, you can use the following:

```
aws kendra create-index --cli-input-json file://create-index-openid.json
```

The secret must have the following format in AWS Secrets Manager:

```
{
    "keys": [
        {
            "alg": "RS256|RS384|RS512",
            "kty": "RSA", //this can be RSA only for now
            "use": "sig", //this value can be sig only for now
            "n": "modulus of standard pem",
            "e": "exponent of standard pem",
            "kid": "key_id",
            "x5t": "certificate thumbprint for x.509 cert",
            "x5c": [
                "certificate chain"
            ]
        }
    ]
}
```

For more information about JWT, see jwt.io.

Python

```
response = client.create_index(
    Name='user-context',
    Edition='ENTERPRISE_EDITION',
    RoleArn='arn:aws:iam::account id:role:/my-role',
```

```

UserTokenConfigurationList=[
    {
        "JwtTokenTypeConfiguration": {
            "KeyLocation": "URL",
            "Issuer": "optional: specify the issuer url",
            "ClaimRegex": "optional: regex to validate claims in the token",
            "UserNameAttributeField": "optional: user",
            "GroupAttributeField": "optional: group",
            "SecretManagerArn": "arn:aws:secretsmanager:us-west-2:account
id:secret:/my-user-context-secret"
        }
    },
    UserContextPolicy='USER_TOKEN'
)

```

Using JSON

The following examples show how to use a JWT with certificate token for user access control when you are create an index.

Warning

The JSON token is a non-validated payload. This should only be used when requests to Kendra come from a trusted server and never from a browser.

Console

1. Choose **Create index** to start creating a new index.
2. On the **Specify index details** page, give your index a name and a description.
3. For **IAM role**, select a role or select **Create a new role** to and specify a role name to create a new role. The IAM role will have the prefix "AmazonKendra-".
4. Leave all of the other fields at their defaults. Choose **Next**.
5. In the **Configure user access control** page, under **Access control settings**, choose **Yes** to use tokens for access control.
6. Under **Token configuration**, select **JSON** as the **Token type**.
7. Specify a **Username** to use in the ACL check.
8. Specify one or more **Groups** to use in the ACL check.
9. Choose**Next**.
10. In the **Provisioning details** page, choose **Developer edition**.
11. Choose **Create** to create your index.
12. Wait for your index to be created. Kendra provisions the hardware for your index. This operation can take some time.

CLI

To create an index with the AWS CLI using a JSON input file, first create a JSON file with your desired parameters:

```
{
    "Name": "user-context",
    "Edition": "ENTERPRISE_EDITION",
    "RoleArn": "arn:aws:iam::account id:role:/my-role",
    "UserTokenConfigurationList": [
        {

```

```

        "JsonTokenTypeConfiguration": {
            "UserNameAttributeField": "user",
            "GroupAttributeField": "group"
        }
    ],
    "UserContextPolicy": "USER_TOKEN"
}

```

Next, call `create-index` using the input file. For example, if the name of your JSON file is `create-index-openid.json`, you can use the following:

```
aws kendra create-index --cli-input-json file://create-index-openid.json
```

If you are not using Open ID for SSO, you can send us the token in JSON format. If you do, you must specify which field in the JSON token contains the user name and which field contains the groups. The group field values must be a JSON string array. For example, if you are using SAML, your token would be similar to the following:

```
{
    "username" : "saligram",
    "groups": [
        "aws-kendra-dev",
        "aws-kendra-engg"
    ]
}
```

The `TokenConfiguration` would specify the user name and group field names:

```
{
    "UserNameAttributeField": "username",
    "GroupAttributeField": "groups"
}
```

Python

```

response = client.create_index(
    Name='user-context',
    Edition='ENTERPRISE_EDITION',
    RoleArn='arn:aws:iam::account id:role:/my-role',
    UserTokenConfigurationList=[
        {
            "JwtTokenTypeConfiguration": {
                "UserNameAttributeField": "user",
                "GroupAttributeField": "group",
            }
        }
    ],
    UserContextPolicy='USER_TOKEN'
)

```

Adding documents directly to an index

You can add documents directly to an index using the [BatchPutDocument \(p. 277\)](#) operation. You can't add documents directly using the console. When you are using the console, you use a data source to add documents.

You can add only the following types of documents with the `BatchPutDocuments` operation.

- Plain text
- HTML
- PDF
- Microsoft PowerPoint
- Microsoft Word

Documents can be added from an Amazon S3 bucket or supplied as binary data.

Adding documents to an index is an asynchronous operation. After you call the `BatchPutDocument` operation, you use the [BatchGetDocumentStatus \(p. 274\)](#) operation to monitor the progress of indexing your documents. When you call the `BatchGetDocumentStatus` operation with a list of document IDs, it returns the status of the document. When the status of the document is `INDEXED` or `FAILED`, processing of the document is complete. When the status is `FAILED`, the `BatchGetDocumentStatus` operation returns the reason that the document couldn't be indexed.

The following examples show how to add documents directly to an index.

Topics

- [Adding documents with the API \(p. 75\)](#)
- [Adding documents from an Amazon S3 bucket \(p. 76\)](#)
- [Adding documents in languages other than English \(p. 78\)](#)

Adding documents with the API

The following example adds text to an index by calling the [BatchPutDocument \(p. 277\)](#) operation.

You can use the `BatchPutDocument` operation to add documents in the following formats:

- DOC
- HTML
- PDF
- Plain text
- PPT

Files added to the index must be in a UTF-8 encoded byte stream. The following example adds UTF-8 encoded text to the index.

Python

```
import boto3

kendra = boto3.client('kendra')

index_id = '${indexID}'

title = 'Information about Amazon.com'
text = 'Amazon.com is an online retailer.'

document = {
    "Id": "1",
    "Blob": text,
    "ContentType": "PLAIN_TEXT",
```

```

        "Title": title
    }

documents = [
    document
]

result = kendra.batch_put_document(
    IndexId = index_id,
    Documents = documents
)

print(result)

```

Java

```

package com.amazonaws.kendra;

import software.amazon.awssdk.core.SdkBytes;
import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.BatchPutDocumentRequest;
import software.amazon.awssdk.services.kendra.model.BatchPutDocumentResponse;
import software.amazon.awssdk.services.kendra.model.ContentType;
import software.amazon.awssdk.services.kendra.model.Document;

public class AddDocumentsViaAPIExample {
    public static void main(String[] args) {
        KendraClient kendra = KendraClient.builder().build();

        String indexId = "yourIndexId";

        Document testDoc = Document
            .builder()
            .title("The title of your document")
            .id("a_doc_id")
            .blob(SdkBytes.fromUtf8String("your text content"))
            .contentType(ContentType.PLAIN_TEXT)
            .build();

        BatchPutDocumentRequest batchPutDocumentRequest = BatchPutDocumentRequest
            .builder()
            .indexId(indexId)
            .documents(testDoc)
            .build();

        BatchPutDocumentResponse result =
        kendra.batchPutDocument(batchPutDocumentRequest);

        System.out.println(String.format("BatchPutDocument Result: %s", result));
    }
}

```

Adding documents from an Amazon S3 bucket

You can add documents directly to your index from an Amazon S3 bucket. You can add up to 10 documents in the same call. When you use an S3 bucket, you must provide an IAM role with permission to access the bucket containing your documents. You specify the role in the `RoleArn` parameter.

Using the [BatchPutDocument \(p. 277\)](#) operation to add documents from an Amazon S3 bucket is a one-time operation. To keep an index synchronized with the contents of a bucket, create an S3 data source. For more information, see [Using an Amazon S3 data source \(p. 86\)](#).

The following example adds two Microsoft Word documents to the index using the `BatchPutDocument` operation.

Python

```
import boto3

kendra = boto3.client('kendra')

index_id = '${indexId}'
role_arn = 'arn:aws:iam::${accountID}:policy/${roleName}'

doc1_s3_file_data = {
    'Bucket': '${bucketName}',
    'Key': 'document1.docx'
}

doc1_document = {
    'S3Path': doc1_s3_file_data,
    'Title': 'Document 1 title',
    'Id': 'doc_1'
}

doc2_s3_file_data = {
    'Bucket': '${bucketName}',
    'Key': 'document2.docx'
}

doc2_document = {
    'S3Path': doc2_s3_file_data,
    'Title': 'Document 2 title',
    'Id': 'doc_2'
}

documents = [
    doc1_document,
    doc2_document
]

result = kendra.batch_put_document(
    Documents = documents,
    IndexId = index_id,
    RoleArn = role_arn
)

print(result)
```

Java

```
package com.amazonaws.kendra;

import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.BatchPutDocumentRequest;
import software.amazon.awssdk.services.kendra.model.BatchPutDocumentResponse;
import software.amazon.awssdk.services.kendra.model.Document;
import software.amazon.awssdk.services.kendra.model.S3Path;

public class AddFilesFromS3Example {
    public static void main(String[] args) {
        KendraClient kendra = KendraClient.builder().build();

        String indexId = "yourIndexId";
        String roleArn = "yourIndexRoleArn";
```

```

Document pollyDoc = Document
    .builder()
    .s3Path(
        S3Path.builder()
            .bucket("an-aws-kendra-test-bucket")
            .key("What is Amazon Polly.docx")
            .build())
    .title("What is Amazon Polly")
    .id("polly_doc_1")
    .build();

Document rekognitionDoc = Document
    .builder()
    .s3Path(
        S3Path.builder()
            .bucket("an-aws-kendra-test-bucket")
            .key("What is Amazon Rekognition.docx")
            .build())
    .title("What is Amazon rekognition")
    .id("rekognition_doc_1")
    .build();

BatchPutDocumentRequest batchPutDocumentRequest = BatchPutDocumentRequest
    .builder()
    .indexId(indexId)
    .roleArn(roleArn)
    .documents(pollyDoc, rekognitionDoc)
    .build();

BatchPutDocumentResponse result =
kendra.batchPutDocument(batchPutDocumentRequest);

System.out.println(String.format("BatchPutDocument result: %s", result));
}
}

```

Adding documents in languages other than English

You can index documents in multiple languages. If you do not specify a language, Amazon Kendra indexes documents in English by default. You include the language code for a document in the document metadata as a field - see information on [field mappings](#) and [custom attributes](#) to include `_language_code` for a document.

You can specify the language code for all your documents in your data source when you call [CreateDataSource](#). If a document does not have a language code specified in a metadata field, the document is indexed using the language code specified for all documents at the data source level. In the console, you can index documents in a supported language only at the data source level. Go to [Data sources](#), then the [Specify data source details](#) page, and choose a language from the dropdown **Language**.

You can index a FAQ in a supported language. Amazon Kendra indexes FAQs in English by default if you do not specify a language. You specify the language code when you call the [CreateFaq](#) operation or you can include the language code for a FAQ in the FAQ metadata as a field. If a FAQ does not have a language code in its metadata specified in a metadata field, the FAQ is indexed using the language code specified when you call the [CreateFAQ](#) operation. To index a FAQ document in a supported language in the console, go to [FAQs](#) and select **Add FAQ**. You choose a language from the dropdown **Languages** and select **Add**.

The following languages and their codes are supported (English or `en` is supported by default if you do not specify a language):

Language name	Language code
Spanish	es
French	fr
German	de
Portuguese	pt
Japanese	ja
Korean	ko
Chinese	zh
Italian	it
Hindi	hi
Arabic	ar
Armenian	hy
Basque	eu
Bengali	bn
Brazilian	pt-BR
Bulgarian	bg
Catalan	ca
Czech	cs
Danish	da
Dutch	nl
Finnish	fi
Galician	gl
Greek	el
Hungarian	hu
Indonesian	id
Irish	ga
Latvian	lv
Lithuanian	lt
Norwegian	no
Persian	fa
Romanian	ro
Russian	ru

Language name	Language code
Sorani	ckb
Swedish	sv
Turkish	tr

Not all Amazon Kendra features are currently available for languages other than English. The following features are not available for non-English indexes:

- Semantic search of [FAQs](#) and extracted answers from documents. Keyword search is used for retrieving relevant FAQs and for document ranking.
- [Custom synonyms](#) for domain-specific, business-specific, or specialized terms.
- [Query suggestions](#) of popular queries relevant to a search.
- [Confidence scores](#) in the search results.

Adding questions and answers directly to an index

You can add questions and answers (FAQs) directly to your index using the console or the [CreateFaq \(p. 292\)](#) operation. You put the data for the FAQ in a file that you store in an Amazon Simple Storage Service (Amazon S3) bucket. You can use comma-separated values (.csv) files or JSON files as input for your FAQ, as follows:

- Basic .csv – A .csv file where each line contains a question, answer, and an optional URL with more information about the answer.
- Custom .csv – A .csv file that contains questions, answers, and a header that defines custom attributes that you can use to facet, display, or sort FAQ responses. You can also define access control attributes to limit the FAQ response to certain users and groups.
- JSON – A JSON file that contains questions, answers, and, optionally, custom and access control attributes. You can define attributes to facet, display, and sort FAQ responses, or access control attributes that limit the FAQ response to certain users and groups.

For example, the following is a basic .csv file that provides answers to questions about free clinics in Spokane, Washington USA and Mountain View, Missouri USA.

```
How many free clinics are in Spokane WA?, 13, https://www.freeclinics.com/
How many free clinics are there in Mountain View, Missouri?, 7, https://
www.freeclinics.com/
```

When you use a custom .csv or JSON file for input, you can declare custom attributes for your FAQ questions. For example, you can create a custom attribute that assigns each FAQ question a department. When the FAQ is returned in a response, you can use the department as a facet to narrow the search.

A custom attribute must map to an index field. You can use a built-in field, or you can specify a custom index field. When you use the console, you use the [Facet definition](#) page to create an index field. When you use the API, you must first create an index field using the [UpdateIndex \(p. 411\)](#) operation.

The attribute type in the FAQ file must match the type of the associated index field. For example, the `_authors` field is a `STRING_LIST` type field, so you must provide values for the `_authors` field as a string list in your FAQ file. You can check the type of index fields using the [Facet definition](#) page in the console or by using the [DescribeIndex \(p. 335\)](#) operation.

When you create an index field that maps to a custom attribute, you can mark it displayable, facetable, or sortable. You can't make a custom attribute searchable.

In addition to custom attributes that you provide, you can also use the following built-in attributes in a custom .csv or JSON file:

- `_authors` (String list) – A list of authors of the answers to the FAQ question.
- `_category` (String) – A category that groups the answers to FAQ questions with other similar documents.
- `_created_at` (ISO 8601-encoded string) – The date and time that the FAQ question was created. The date and time should be ISO 8601-encoded strings.

It is important for the time zone to be included in the ISO 8601 date-time format. For example, 20120325T123010+01:00 is the ISO 8601 date-time format for March 25th 2012 at 12:30PM (plus 10 seconds) in Central European Time.

- `_last_updated_at` (ISO 8601-encoded string) – The date and time that the FAQ question was updated. The date and time should be ISO 8601-encoded strings.

It is important for the time zone to be included in the ISO 8601 date-time format. For example, 20120325T123010+01:00 is the ISO 8601 date-time format for March 25th 2012 at 12:30PM (plus 10 seconds) in Central European Time.

- `_source_uri` (String) – A URL for a document with more information about the FAQ answer.
- `_version` (String) – The version of the FAQ question.
- `_view_count` (Long) – The number of times that the FAQ question has been viewed in search results.

Basic .csv file

Use a basic .csv file when you want to use a simple structure for your FAQs. In a basic .csv file, each line has two or three fields: question, answer, and an optional source URI that points to a document with more information.

The contents of the file should follow the [RFC 4180 Common Format and MIME Type for Comma-Separated Values \(CSV\) Files](#).

The following is a FAQ file in basic .csv format.

```
How many free clinics are in Spokane WA?, 13, https://www.freeclinics.com/
How many free clinics are there in Mountain View, Missouri?, 7, https://
www.freeclinics.com/
```

Custom .csv file

Use a custom .csv file when you want to add custom attributes to your FAQ questions. You use a header row in your .csv file to define the additional attributes.

The .csv file must contain the following two required attributes:

- `_question` – The frequently asked question
- `_answer` – The answer to the frequently asked question

Your file can contain built-in and custom attributes. The following is an example of a custom .csv file.

```
_question,_answer,_last_updated_at,custom_string
```

```
How many free clinics are in Spokane WA?, 13, 20120325T123010+01:00, Note: Some free
clinics require you to meet certain criteria in order to use their services
How many free clinics are there in Mountain View, Missouri?, 7, 20120325T123010+01:00,
Note: Some free clinics require you to meet certain criteria in order to use their
services
```

The contents of the custom attributes file should follow the [RFC 4180 Common Format and MIME Type for Comma-Separated Values \(CSV\) Files](#).

There are four types of custom attributes:

- Date – ISO 8601-encoded date and time values.

It is important for the time zone to be included in the ISO 8601 date-time format. For example, 20120325T123010+01:00 is the ISO 8601 date-time format for March 25th 2012 at 12:30PM (plus 10 seconds) in Central European Time.

- Long – Numbers, such as 1234.
- String – String values. If your string contains commas, enclose the value in double quotation marks (""). For example, "custom attribute, and more".
- String list – A list of string values. List the values in a comma-separated list that is enclosed in quotation marks. For example, "item1, item2, item3". If the list contains only a single entry, you can omit the quotation marks. For example, item1.

A custom .csv file can contain user context fields, which you can use to limit access to the FAQ to certain users and groups. To filter on user context, the user must provide user and group information in the query. Otherwise, all relevant FAQs are returned. For more information, see [Filtering on user context \(p. 136\)](#).

There are four user context filters for FAQs:

- `_acl_user_allow` – Users in the allow list can see the FAQ in the query response. The FAQ isn't returned to other users.
- `_acl_user_deny` – Users in the deny list can't see the FAQ in the query response. The FAQ is returned to all other users when it is relevant to the query.
- `_acl_group_allow` – Users that are members of an allowed group can see the FAQ in the query response. The FAQ isn't returned to users that are members of another group.
- `_acl_group_deny` – Users that are members of a denied group can't see the FAQ in the query response. The FAQ is returned to other groups when it is relevant to the query.

Provide the values for the allow and deny lists in comma-separated lists enclosed in quotation marks. For example, "user1,user2,user3". You can include a user or a group in either an allow list or a deny list, but not both. If you include a user or group in both, you receive an error.

The following is an example of a custom .csv file with user context information.

```
_question, _answer, _acl_user_allow, _acl_user_deny, _acl_group_allow, _acl_group_deny
How many free clinics are in Spokane WA?, 13, userID4565, "userID6201, userID7552",
groupBasicRate, "groupBasicPlusRate,groupPremiumRate"
```

JSON file

You can use a JSON file to provide questions, answers, and attributes for your index. You can add any of the built-in attributes and custom attributes to the FAQ. This is the schema schema for the JSON file.

```
{
```

```

    "SchemaVersion": 1,
    "FaqDocuments": [
        {
            "Question": string,
            "Answer": string,
            "Attributes": {
                string: object
                additional attributes
            },
            "AccessControlList": [
                {
                    "Name": string,
                    "Type": enum( "GROUP" | "USER" ),
                    "Access": enum( "ALLOW" | "DENY" )
                },
                additional user context
            ]
        },
        additional FAQ documents
    ]
}

```

This example JSON file shows two FAQ documents, one with just the required question and answer, and one with additional attributes and user context information.

```

{
    "SchemaVersion": 1,
    "FaqDocuments": [
        {
            "Question": "How many free clinics are in Spokane WA?",
            "Answer": "13"
        },
        {
            "Question": "How many free clinics are there in Mountain View, Missouri?",
            "Answer": "7",
            "Attributes": {
                "_source_uri": "https://www.freeclinics.com",
                "_category": "Charitable Clinics"
            },
            "AccessControlList": [
                {
                    "Name": "user@amazon.com",
                    "Type": "USER",
                    "Access": "ALLOW"
                },
                {
                    "Name": "Admin",
                    "Type": "GROUP",
                    "Access": "ALLOW"
                }
            ]
        }
    ]
}

```

There are four types of custom attributes for JSON files:

- Date – A JSON string value with ISO 8601-encoded date and time values.

It is important for the time zone to be included in the ISO 8601 date-time format. For example, 20120325T123010+01:00 is the ISO 8601 date-time format for March 25th 2012 at 12:30PM (plus 10 seconds) in Central European Time.

- Long – A JSON number value, such as 1234.

- String – A JSON string value. For example, "custom attribute".
- String list – A JSON array of string values. For example, ["item1,item2,item3"].

In addition to built-in and custom attributes, you can provide user context information for the FAQ in a JSON file. You can provide user context information to limit access to the FAQ content based on users and groups. You can include a user or a group in either an allow list or a deny list, but not both. If you include a user or group in both, you receive an error. To filter on user context, you must provide user and group information in the query. Otherwise, all relevant FAQs are returned. For more information, see [Filtering on user context \(p. 136\)](#).

This JSON example adds user context to a FAQ.

```
"AccessControlList": [
    {
        "Name": "group or user name",
        "Type": "GROUP | USER",
        "Access": "ALLOW | DENY"
    },
    additional user context
]
```

Using your FAQ file

After you store your FAQ input file in an Amazon S3 bucket, you use the console or the `CreateFaq` operation to put the questions and answers into your index. If you want to update a FAQ, you need to delete the FAQ and create it again. You use the `DeleteFaq` operation to delete a FAQ.

You must provide an IAM role that has access to the S3 bucket containing your source files. You specify the role in the console or in the `RoleArn` parameter. The following is an example of a program that adds a FAQ file to an index.

Python

```
import boto3

kendra = boto3.client('kendra')

index_id = '${indexId}'
role_arn = 'arn:aws:iam::${accountId}:role/${roleName}'

faq_path = {
    'Bucket': '${bucketName}',
    'Key': 'FreeClinicsUSA.csv'
}

response = kendra.create_faq(
    S3Path = faq_path,
    Name = 'FreeClinicsUSA',
    IndexId = index_id,
    RoleArn = role_arn
)

print(response)
```

Java

```
package com.amazonaws.kendra;
```

```

import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.CreateFaqRequest;
import software.amazon.awssdk.services.kendra.model.CreateFaqResponse;
import software.amazon.awssdk.services.kendra.model.S3Path;

public class AddFaqExample {
    public static void main(String[] args) {
        KendraClient kendra = KendraClient.builder().build();

        String indexId = "yourIndexId";
        String roleArn = "your role for accessing S3 files";

        CreateFaqRequest createFaqRequest = CreateFaqRequest
            .builder()
            .indexId(indexId)
            .name("FreeClinicsUSA")
            .roleArn(roleArn)
            .s3Path(
                S3Path
                    .builder()
                    .bucket("an-aws-kendra-test-bucket")
                    .key("FreeClinicsUSA.csv")
                    .build())
            .build();

        CreateFaqResponse response = kendra.createFaq(createFaqRequest);

        System.out.println(String.format("The result of creating FAQ: %s", response));
    }
}

```

Adding documents from a data source

When you create a data source, you give Amazon Kendra the location of documents that it should index. Unlike adding documents directly to an index, you can periodically scan the data source to update the index.

For example, say that you have a repository of tax instruction stored in an Amazon S3 bucket. Existing documents are changed and new documents are added to the repository from time to time. If you add the repository to Amazon Kendra as a data source, you can keep your index up to date by periodically updating your index.

You can update the index manually using console or the [StartDataSourceSyncJob \(p. 392\)](#) operation, or you can set up a schedule to update the index.

An index can have more than one data source. Each data source can have its own update schedule. For example, you might update the index of your working documents daily, or even hourly, while updating your archived documents manually whenever the archive changes.

Setting an update schedule

Configure your data source to periodically update with the console or by using the `Schedule` parameter when you create or update a data source. The content of the parameter is a string that holds either a cron-format schedule string or an empty string to indicate that the index should be updated on demand. For the format of a cron expression, see [Schedule Expressions for Rules](#) in the *Amazon CloudWatch Events User Guide*. Amazon Kendra supports only cron expressions. It does not support rate expressions.

Setting a language

You can index all your documents in a data source in a supported language. You specify the language code for all your documents in your data source when you call [CreateDataSource](#). If a document does not have a language code specified in a metadata field, the document is indexed using the language code specified for all documents at the data source level. If you do not specify a language, Amazon Kendra indexes documents in a data source in English by default. For more information on supported languages, including their codes, see [Adding documents in languages other than English](#).

You index all your documents in a data source in a supported language using the console. Go to **Data sources** or **Add data sources** if you are adding a new data source. On the **Specify data source details** page, choose a language from the dropdown **Language**. You select **Update** or continue to enter the configuration information to connect to your data source.

- [Using an Amazon S3 data source \(p. 86\)](#)
- [Using an Atlassian Confluence data source \(p. 90\)](#)
- [Using a custom data source \(p. 94\)](#)
- [Using a database data source \(p. 100\)](#)
- [Using a Google Workspace Drive data source \(p. 102\)](#)
- [Using a Microsoft OneDrive data source \(p. 104\)](#)
- [Using a Salesforce data source \(p. 106\)](#)
- [Using a ServiceNow data source \(p. 109\)](#)
- [Using a Microsoft SharePoint data source \(p. 114\)](#)
- [Using a web crawler data source \(p. 116\)](#)

Using an Amazon S3 data source

Use an Amazon S3 data source when your document repository is an S3 bucket. For a walk-through of how to use an S3 bucket in the console, see [Getting started with an S3 bucket \(console\)](#).

You must create an index before you create a data source. You provide the index identifier as a parameter to the [CreateDataSource \(p. 283\)](#) operation.

The S3 bucket must be in the same Region as the index and Amazon Kendra must have permission to access the S3 bucket that contains your documents. You provide the Amazon Resource Name (ARN) of a role that has access when you create the data source using the `RoleARN` parameter.

The following examples demonstrate creating an S3 data source. The examples assume that you have already created an index and an IAM role with permission to read the data from the index. For more information about the IAM role, see [IAM roles for Amazon S3 data sources \(p. 15\)](#). For more information about creating an index, see [Creating an index \(p. 64\)](#).

CLI

```
aws kendra create-data-source \
--index-id index ID \
--name example-data-source \
--type S3 \
--configuration '{"S3Configuration":{"BucketName":"bucket name"}}' \
--role-arn 'arn:aws:iam::account id:role/role name'
```

Python

The following snippet of Python code creates an S3 data source. For the complete example, see [Getting started \(AWS SDK for Python \(Boto3\)\) \(p. 42\)](#).

```
print("Create an S3 data source")

name = "getting-started-data-source"
description = "Getting started data source."
s3_bucket_name = "${bucketName}"
type = "S3"
role_arn = "arn:aws:iam::${accountID}:role/${roleName}"

configuration = {"S3DataSourceConfiguration":
{
    "BucketName": s3_bucket_name
}
}

data_source_response=kendra.create_data_source(
    Configuration = configuration,
    Name = name,
    Description = description,
    RoleArn = role_arn,
    Type = type,
    IndexId = index_id
)
```

It can take some time to create your data source. You can monitor the progress by using the [DescribeDataSource \(p. 322\)](#) operation. When the data source status is ACTIVE the data source is ready to use.

The following examples demonstrate getting the status of a data source.

CLI

```
aws kendra describe-data-source \
--index-id index ID \
--id data source ID
```

Python

The following snippet of Python code gets information about an S3 data source. For the complete example, see [Getting started \(AWS SDK for Python \(Boto3\)\) \(p. 42\)](#).

```
print("    Wait for Kendra to create the data source.")

while True:
    data_source_description = kendra.describe_data_source(
        Id = "data source ID",
        IndexId = "index ID"
    )
    status = data_source_description["Status"]
    print("Creating data source. Status: "+status)
    time.sleep(60)
    if status != "CREATING":
        break
```

This data source doesn't have a schedule, so it does not run automatically. To index the data source, you call the [StartDataSourceSyncJob \(p. 392\)](#) operation to synchronize the index with the data source.

The following examples demonstrate synchronizing a data source.

CLI

```
aws kendra start-data-source-sync-job \
--index-id index ID \
--id data source ID
```

Python

The following snippet of Python code synchronizes an S3 data source. For the complete example, see [Getting started \(AWS SDK for Python \(Boto3\)\) \(p. 42\)](#).

```
print("Synchronize the data source.")

sync_response = kendra.start_data_source_sync_job(
    Id = "data source ID",
    IndexId = "index ID"
)
```

S3 document metadata

You can add metadata, additional information about a document, to documents in an Amazon S3 bucket using a metadata file. Each metadata file is associated with an indexed document.

Your metadata files must be stored in the same bucket as your indexed files. You can specify a location within the bucket for your metadata files using the console or the `S3Prefix` field of the `DocumentsMetadataConfiguration` parameter when you create an S3 data source. If you don't specify an S3 prefix, your metadata files must be stored in the same location as your indexed documents.

If you specify an S3 prefix for your metadata files, they live in a directory structure parallel to your indexed documents. Amazon Kendra looks only in the specified directory for your metadata. If the metadata isn't read, check that the directory location matches the location of your metadata.

The following examples show how the indexed document location maps to the metadata file location. Note that the document's S3 key is appended to the metadata's S3 prefix and then suffixed with `.metadata.json` to form the metadata file's S3 path. The combined S3 key, with the metadata's S3 prefix and `.metadata.json` suffix must be no more than a total of 1024 characters. It is recommended that you keep your S3 key below 1000 characters to account for additional characters when combining your key with the prefix and suffix.

```
Bucket name:
  s3://bucketName
Document path:
  documents
Metadata path:
  none
File mapping
  s3://bucketName/documents/file.txt ->
    s3://bucketName/documents/file.txt.metadata.json
```

```
Bucket name:
  s3://bucketName
Document path:
  documents/legal
Metadata path:
  metadata
File mapping
  s3://bucketName/documents/legal/file.txt ->
    s3://bucketName/metadata/documents/legal/file.txt.metadata.json
```

Your document metadata is defined in a JSON file. The file must be a UTF-8 text file without a BOM marker. The file name of the JSON file should be `document.extension.metadata.json`, where "document" is the name of the document that the metadata applies to and "extension" is the file extension for the document.

The content of the JSON file follows this template. All of the attributes are optional. If you don't specify the `_source_uri`, then the links returned by Amazon Kendra in search results point to the S3 bucket that contains the document.

```
{
    "DocumentId": "document ID",
    "Attributes": {
        "_category": "document category",
        "_created_at": "ISO 8601 encoded string",
        "_last_updated_at": "ISO 8601 encoded string",
        "_source_uri": "document URI",
        "_version": "file version",
        "_view_count": number of times document has been viewed,
        "custom attribute key": "custom attribute value",
        additional custom attributes
    },
    "AccessControlList": [
        {
            "Name": "user name",
            "Type": "GROUP | USER",
            "Access": "ALLOW | DENY"
        }
    ],
    "Title": "document title",
    "ContentType": "HTML | MS_WORD | PDF | PLAIN_TEXT | PPT"
}
```

The `_created_at` and `_last_updated_at` metadata fields are ISO 8601 encoded dates. For example, 20120325T123010+01:00 is the ISO 8601 date-time format for March 25th 2012 at 12:30PM (plus 10 seconds) in Central European Time.

You can add additional information to the `Attributes` field about a document that you use to filter queries or to group query responses. For more information, see [Creating custom document attributes \(p. 121\)](#).

The `AccessControlList` field enables you to filter the response from a query so that only certain users and groups have access to documents. For more information, see [Filtering on user context \(p. 136\)](#).

Access control for S3 data sources

You can control access to documents in an S3 data source using a configuration file. You specify the file in the console or as the `AccessControlListConfiguration` parameter when you call the [CreateDataSource \(p. 283\)](#) or [UpdateDataSource \(p. 403\)](#) operation.

The configuration file contains a JSON structure that identifies an S3 prefix and lists the access settings for the prefix. The prefix can be a path, or it can be an individual file. If the prefix is a path, the access settings apply to all of the files in that path.

You can specify both users and groups in the access settings. When you query the index, you specify user and group information. For more information, see [Filtering by user attribute \(p. 138\)](#).

The JSON structure for the configuration file should be in the following format:

```
[  
    {
```

```
    "keyPrefix": "s3://prefix1",
    "aclEntries": [
        {
            "Name": "user1",
            "Type": "USER",
            "Access": "ALLOW"
        },
        {
            "Name": "group1",
            "Type": "GROUP",
            "Access": "DENY"
        }
    ],
    {
        "keyPrefix": "s3://prefix2",
        "aclEntries": [
            {
                "Name": "user2",
                "Type": "USER",
                "Access": "ALLOW"
            },
            {
                "Name": "user1",
                "Type": "USER",
                "Access": "DENY"
            },
            {
                "Name": "group1",
                "Type": "GROUP",
                "Access": "DENY"
            }
        ]
    }
]
```

Using an Atlassian Confluence data source

You can use Amazon Kendra to index blogs, pages, and attachments to either blogs and pages that are hosted on an Atlassian Confluence cloud or Atlassian Confluence server instance. For a walk-through of how to use Confluence in the console, see [Getting started with a Confluence data source \(console\)](#).

By default, Amazon Kendra indexes regular spaces, you can configure Amazon Kendra to index personal and archived spaces. To exclude or include specific blogs, pages, and spaces from indexing, you use a list.

Amazon Kendra indexes pages and blogs by default. You can restrict the pages and blogs that are crawled using inclusion and exclusion patterns. When you use an inclusion pattern, only those pages and blogs that match the inclusion pattern are indexed. If you specify an exclusion pattern, pages that match the pattern are not indexed.

When you create the data source, you can use the console or the [CreateDataSource \(p. 283\)](#) operation to specify whether Amazon Kendra indexes the attachments to pages and blogs. If you choose to index attachments, only attachments to the indexed pages and blogs are indexed.

Before you can index the documents on a Confluence instance, you must create an account with administrative permission that Amazon Kendra can use to connect to the server. The account must grant Amazon Kendra permission to view all of the content within your Confluence instance. You can grant the account these permissions by making it a member of the `confluence-administrators` group.

You store the Confluence credentials in an AWS Secrets Manager secret. If you are using the Amazon Kendra console to create your data source, you can create the secret while creating the data source. Or

you can use an existing Secrets Manager secret. If you are using the API to create your data source, you must provide the Amazon Resource Name (ARN) of an existing secret.

The secret must contain the following information:

- User name
- Password or API token

The credentials are stored as a JSON string in the Secrets Manager secret.

```
{  
    "username": "user-name",  
    "password": "password or API token"  
}
```

For Atlassian Confluence cloud, you must generate an API token . For more information, see [API tokens](#) on the Atlassian website.

The data source IAM role must have permission to access the secret and the AWS Key Management Service key used to decrypt it. For more information see [IAM role for Confluence server data sources \(p. 16\)](#).

You must create an index before you create the Confluence data source. For more information, see [Creating an index \(p. 64\)](#). You provide the ID of the index when you create the data source.

You specify connection and other information in the console or in an instance of the [ConfluenceConfiguration \(p. 445\)](#) data type. You must provide the following information:

- Credentials to connect to the Confluence instance.
- The URL of the Confluence instance that contains your documents.
- The version of Confluence that you are using.

Indexing spaces

Amazon Kendra includes information from a space in the index. A space may be included in the results of a query based on this information. The Confluence account used for the data source must have permission to access the space in order to index it.

By default, Amazon Kendra doesn't index Confluence archive and personal spaces. You can choose to index them when you create the data source. If you don't want Amazon Kendra to index a space, mark it private in Confluence .

You can restrict access to the contents of a space by specifying view permissions. If a query includes user information, Amazon Kendra reads these permissions and uses them for user context filtering. For more information, see [Filtering on user context \(p. 136\)](#).

If you use the Amazon Kendra console to create a Confluence data source, Amazon Kendra creates index fields for you when you specify a field mapping. If you use the API, you must first create the index field using the [UpdateIndex \(p. 411\)](#) operation. To map the Confluence fields to Amazon Kendra index fields, see the following table.

Confluence field	Suggested Amazon Kendra field
DISPLAY_URL	_source_uri

Confluence field	Suggested Amazon Kendra field
ITEM_TYPE	_category
SPACE_KEY	cf_space_key
URL	cf_url

Indexing pages

Amazon Kendra indexes all pages, including nested pages, in a space unless they are filtered out by an inclusion or exclusion pattern.

The Confluence security model uses nested permissions. To have access to a page, the account that you use to connect to the Confluence instance. If a query includes user information, Amazon Kendra reads these permissions and uses them for user context filtering. For more information, see [Filtering on user context \(p. 136\)](#).

If you use the console to create a Confluence data source, Amazon Kendra creates the index fields for you when you specify a field mapping. If you use the API, you must first create the index field using the [UpdateIndex \(p. 411\)](#) operation. To map the Confluence fields to Amazon Kendra index fields, see the following table.

Confluence field	Suggested Amazon Kendra field
AUTHOR	cf_author
CONTENT_STATUS	cf_page_content_status
CREATED_DATE	_created_at
DISPLAY_URL	_source_uri
ITEM_TYPE	_category
LABELS	cf_labels
MODIFIED_DATE	_last_updated_at
PARENT_ID	cf_parent_id
SPACE_KEY	cf_space_key
SPACE_NAME	cf_space_name
URL	cf_url
VERSION	cf_version

Blogs

Amazon Kendra indexes all blogs in a space unless they are filtered from indexing by an inclusion or an exclusion pattern.

The Confluence security model uses nested permissions. To index a blog, the user that you use to connect to the Confluence instance must have access to the blog and the space that contains it. If a

query includes user information, Amazon Kendra reads these permissions and uses them for user context filtering. For more information, see [Filtering on user context \(p. 136\)](#).

If you use the console to index a Confluence data source, Amazon Kendra creates the index fields for you when you specify a field mapping. If you use the API, you must first create the index field using the [UpdateIndex \(p. 411\)](#) operation. To map the Confluence data source fields to Amazon Kendra index fields, see the following table.

Confluence field	Suggested Amazon Kendra field
AUTHOR	cf_author
DISPLAY_URL	_source_uri
ITEM_TYPE	_category
LABELS	cf_labels
PUBLISH_DATE	_created_at
SPACE_KEY	cf_space_key
SPACE_NAME	cf_space_name
URL	cf_url
VERSION	cf_version

Attachments

Confluence enables you to create attachments to pages and blog posts. By default, attachments aren't indexed. You can configure Amazon Kendra to include attachments in the index. Amazon Kendra includes only attachments to indexed pages and blogs in the index.

Amazon Kendra indexes only the following supported documents types:

- Microsoft Word
- Microsoft PowerPoint
- HTML
- PDF
- Plain text

The Confluence security model uses nested permissions. To index to an attachment, the account that you use to connect the Confluence instance must have access to the blog or page and the space that contains it. If a query includes user information, Amazon Kendra reads these permissions and uses them for user context filtering. For more information, see [Filtering on user context \(p. 136\)](#).

If you use the console, Amazon Kendra creates index fields for you when you specify a field mapping. If you use the API, you must first create the index field using the [UpdateIndex \(p. 411\)](#) operation. To map the Confluence fields to Amazon Kendra fields, see the following table.

Confluence field	Suggested Amazon Kendra field
AUTHOR	cf_author

Confluence field	Suggested Amazon Kendra field
CONTENT_TYPE	cf_attachment_content_type
CREATED_DATE	_created_at
DISPLAY_URL	_source_uri
FILE_SIZE	cf_attachment_file_size
ITEM_TYPE	_category
LABELS	cf_labels
PARENT_ID	cf_parent_id
SPACE_KEY	cf_space_key
SPACE_NAME	cf_space_name
URL	cf_url
VERSION	cf_version

Using a custom data source

Use a custom data source when you have a repository that Amazon Kendra doesn't yet provide a data source connector for. It enables you to see the same run history metrics that Amazon Kendra data sources provide even when you can't use Amazon Kendra's data sources to sync your repositories. Use this to create a consistent sync monitoring experience between Amazon Kendra data sources and custom ones. Specifically, use a custom data source to see sync metrics for a data source connector that you created using the [BatchPutDocument \(p. 277\)](#) and [BatchDeleteDocument \(p. 271\)](#) operations.

When you create a custom data source, you have complete control over how the documents to index are selected. Amazon Kendra only provides metric information that you can use to monitor your data source sync jobs. You must create and run the crawler that determines the documents your data source indexes.

You create an identifier for your custom data source using the console or by using the [CreateDataSource \(p. 283\)](#) operation. To use the console, give your data source a name, and optionally a description and resource tags. After the data source is created, a data source ID is shown. Copy this ID to use when you synchronize the data source with the index.

Specify data source details

Name data source

Data source name

Maximum of 1000 alphanumeric characters. Can include hyphens (-), but not spaces.

Description - *optional*

Tags (0) - optional Info

A tag is an administrative label that you assign to AWS resources to make it easier to manage them. Each tag consists of a key and an optional value. Use tags to search and filter your resources or track your AWS costs.

This resource has no tags

[Add new tag](#)

You can add up to 50 more tags.

[Cancel](#) [Next](#)

You can also create a custom data source using the [CreateDataSource](#) operation. The operation returns an ID to use when you synchronize the data source. When you use the [CreateDataSource](#) operation to create a custom data source, you can't set the Configuration, RoleArn or Schedule parameters. If you set these parameters, Amazon Kendra returns a [ValidationException](#) exception.

To use a custom data source, create an application that is responsible for updating the Amazon Kendra index. The application depends on a crawler that you create. The crawler reads the documents in your repository and determines which should be sent to Amazon Kendra. Your application should perform the following steps:

1. Crawl your repository and make a list of the documents in your repository that have been added, updated, or deleted.
2. Call the [StartDataSourceSyncJob](#) (p. 392) operation to signal that a sync job is starting. You provide a data source ID to identify the data source that is synchronizing. Amazon Kendra returns a execution ID to identify a particular sync job.
3. Call the [BatchPutDocument](#) (p. 277) and [BatchDeleteDocument](#) (p. 271) operations to add, update, and remove documents from the index. You provide the data source ID and execution ID to identify the data source that is synchronizing and the job that this update is associated with.
4. Call the [StopDataSourceSyncJob](#) (p. 394) operation to signal the end of the sync job. Once you call the [StopDataSourceSyncJob](#) operation, the associated execution ID is no longer valid.
5. Call the [ListDataSourceSyncJobs](#) (p. 360) operation with the index and data source identifiers to list the sync jobs for the data source and to see metrics for the sync jobs.

After you end a sync job, you can start a new synchronization job. There can be a period of time before all of the submitted documents are added to the index. Use the [ListDataSourceSyncJobs](#) operation

to see the status of the sync job. If the `Status` returned for the sync job is `SYNCING_INDEXING`, some documents are still being indexed. You can start a new sync job when the status of the previous job is `FAILED`, `SUCCEEDED`, or `SYNCING_INDEX`.

After you call the `StopDataSourceSyncJob` operation, you can't use a sync job identifier in a call to the `BatchPutDocument` or `BatchDeleteDocument` operations. If you do, all of the documents submitted are returned in the `FailedDocuments` response message from the operation.

Required attributes

When you submit a document to Amazon Kendra using the `BatchPutDocument` operation, each document requires two attributes to identify the data source and synchronization run that it belongs to. You must provide the following two attributes:

- `_data_source_id` – The identifier of the data source. This is returned when you create the data source with the console or the `CreateDataSource` operation.
- `_data_source_sync_job_execution_id` – The identifier of the sync run. This is returned when you start the index synchronization with the `StartDataSourceSyncJob` operation.

The following is the JSON required to index a document using a custom data source.

```
{  
    "Documents": [  
        {  
            "Attributes": [  
                {  
                    "Key": "_data_source_id",  
                    "Value": {  
                        "StringValue": "data source identifier"  
                    }  
                },  
                {  
                    "Key": "_data_source_sync_job_execution_id",  
                    "Value": {  
                        "StringValue": "sync job identifier"  
                    }  
                }  
            ],  
            "Blob": "document content",  
            "ContentType": "content type",  
            "Id": "document identifier",  
            "Title": "document title"  
        }  
    ],  
    "IndexId": "index identifier",  
    "RoleArn": "IAM role ARN"  
}
```

When you remove a document from the index using the `BatchDeleteDocument` operation, you need to specify the following two fields in the `DataSourceSyncJobMetricTarget` parameter:

- `DataSourceId` – The identifier of the data source. This is returned when you create the data source with the console or the `CreateDataSource` operation.
- `DataSourceSyncJobId` – The identifier of the sync run. This is returned when you start the index synchronization with the `StartDataSourceSyncJob` operation.

The following is the JSON required to delete a document from the index using the `BatchDeleteDocument` operation.

```
{  
    "DataSourceSyncJobMetricTarget": {  
        "DataSourceId": "data source identifier",  
        "DataSourceSyncJobId": "sync job identifier"  
    },  
    "DocumentIdList": [  
        "document identifier"  
    ],  
    "IndexId": "index identifier"  
}
```

Viewing metrics

After a sync job is finished, you can use the [ListDataSourceSyncJobs \(p. 360\)](#) operation to get the metrics associated with the sync job. Use this to monitor your custom data source syncs.

If you submit the same document multiple times, either as part of the `BatchPutDocument` operation, the `BatchDeleteDocument` operation, or if the document is submitted for both addition and deletion, the document is only counted once in the metrics.

- `DocumentsAdded` – The number of documents submitted with a `BatchPutDocument` operation associated with this sync job added to the index for the first time. If a document is submitted for addition more than once in a sync, the document is only counted once in the metrics.
- `DocumentsDeleted` – The number of documents submitted with a `BatchDeleteDocument` operation associated with this sync job deleted from the index. If a document is submitted for deletion more than once in a sync, the document is only counted once in the metrics.
- `DocumentsFailed` – The number of documents associated with this sync job that failed indexing. These are documents that were accepted by Amazon Kendra for indexing but could not be indexed or deleted. If a document is not accepted by Amazon Kendra, the identifier for the document is returned in the `FailedDocuments` response property of the `BatchPutDocument` and `BatchDeleteDocument` operations
- `DocumentsModified` – The number of modified documents submitted with a `BatchPutDocument` operation associated with this sync job that were modified in the Amazon Kendra index.

Amazon Kendra also emits Amazon CloudWatch metrics while indexing documents. For more information, see [Metrics for indexed documents \(p. 235\)](#).

Amazon Kendra doesn't return the `DocumentsScanned` metric for custom data sources. It also emit the CloudWatch metrics listed in [Metrics for Amazon Kendra data sources \(p. 233\)](#).

Custom data source (Java)

The following code provides a sample implementation of a custom data source using Java. The program first creates a custom data source and then uses that data source to add and remove documents from an index. Finally, it gets the metrics of the data source synchronization run.

Note

The following code demonstrates creating and using a custom data source in the same sample. When you are using a custom data source in your application it isn't necessary to create a new data source each time that you synchronize your index.

```
import com.amazonaws.services.kendra.AWSkendra;  
import com.amazonaws.services.kendra.AWSkendraClientBuilder;  
import com.amazonaws.services.kendra.model.BatchDeleteDocumentRequest;  
import com.amazonaws.services.kendra.model.BatchDeleteDocumentResult;  
import com.amazonaws.services.kendra.model.BatchPutDocumentRequest;  
import com.amazonaws.services.kendra.model.BatchPutDocumentResult;  
import com.amazonaws.services.kendra.model.CreateDataSourceRequest;
```

```
import com.amazonaws.services.kendra.model.CreateDataSourceResult;
import com.amazonaws.services.kendra.model.DataSourceSyncJobMetricTarget;
import com.amazonaws.services.kendra.model.DataSourceType;
import com.amazonaws.services.kendra.model.Document;
import com.amazonaws.services.kendra.model.DocumentAttribute;
import com.amazonaws.services.kendra.model.DocumentAttributeValue;
import com.amazonaws.services.kendra.model.ListDataSourceSyncJobsRequest;
import com.amazonaws.services.kendra.model.ListDataSourceSyncJobsResult;
import com.amazonaws.services.kendra.model.StartDataSourceSyncJobRequest;
import com.amazonaws.services.kendra.model.StartDataSourceSyncJobResult;
import com.amazonaws.services.kendra.model.StopDataSourceSyncJobRequest;
import com.amazonaws.services.kendra.model.StopDataSourceSyncJobResult;

public class SampleSyncForCustomDataSource {

    public static void main(String[] args) {

        final AWSkendra awskendraClient = AWSkendraClientBuilder.standard().build();

        final String indexId = "Amazon Kendra index ID";

        // Create custom data source.
        final CreateDataSourceRequest createDataSourceRequest = new CreateDataSourceRequest()
            .withName("sample-custom-data-source")
            .withType(DataSourceType.CUSTOM)
            .withDescription("description of sample-custom-data-source")
            .withIndexId(indexId);
        final CreateDataSourceResult createDataSourceResult =
        awskendraClient.createDataSource(createDataSourceRequest);

        // Get the data source id from createDataSourceResult.
        final String datasourceId = createDataSourceResult.getId();

        // Wait for the custom data source to become active.
        // You can use the DescribeDataSource operation to check the status
        .
        .
        .

        // Start the sync by calling StartDataSourceSync and providing your index id
        // and your custom data source id
        final StartDataSourceSyncJobResult startDataSourceSyncJobResult =
        awskendraClient.startDataSourceSyncJob(
            new StartDataSourceSyncJobRequest()
                .withIndexId(indexId)
                .withId(datasourceId)
        );
        final String executionId = startDataSourceSyncJobResult.getExecutionId();

        // To associate documents with an synchronization run, add the data source ID and
        // execution ID as attributes to the BatchPutDocument request. The key for the
        // data source ID is "_data_source_id" and the key for the execution run ID is
        // "_data_source_sync_job_execution_id".
        final BatchPutDocumentRequest batchPutDocumentRequest = new BatchPutDocumentRequest()
            .withIndexId(indexId)
            .withDocuments(
                new Document()
                    .withAttributes(
                        new DocumentAttribute()
                            .withKey("_data_source_id")
                            .withValue(
                                new DocumentAttributeValue()
                                    .withStringValue(datasourceId)
                            )
                    ),
                new DocumentAttribute()
            )
    }
}
```

```

        .withKey("_data_source_sync_job_execution_id")
        .WithValue(
            new DocumentAttributeValue()
                .WithValue(executionId)
        )
    ).withId("first_document_id")
    .withBlob(..)
    .
    .
    ,
    new Document()
        .withAttributes(
            new DocumentAttribute()
                .withKey("_data_source_id")
                .WithValue(
                    new DocumentAttributeValue()
                        .WithValue(datasourceId)
                ),
            new DocumentAttribute()
                .withKey("_data_source_sync_job_execution_id")
                .WithValue(
                    new DocumentAttributeValue()
                        .WithValue(executionId)
                )
        )
    .
    .
    .
    ,
    ....More documents
    );
// Call the BatchPutRequest operation.
final BatchPutDocumentResult batchPutDocumentResult =
awsKendraClient.batchPutDocument(batchPutDocumentRequest);

// To delete documents, provide you custom data source ID and job execution ID in the
// DataSourceSyncJobMetrics parameter in the BatchDeleteDocument request.
BatchDeleteDocumentRequest batchDeleteDocumentRequest = new
BatchDeleteDocumentRequest()
    .withDocumentIdList(
        "id_of_first_document_to_delete",
        "id_of_second_document_to_delete",
        "id_of_third_document_to_delete",
        .
        .
        .
    )
    .withDataSourceSyncJobMetricTarget(
        new DataSourceSyncJobMetricTarget()
            .withDataSourceSyncJobId(executionId)
            .withDataSourceId(datasourceId)
    )
    .withIndexId(indexId);
// Make BatchPutRequest call.
final BatchDeleteDocumentResult batchDeleteDocumentResult =
awsKendraClient.batchDeleteDocument(batchDeleteDocumentRequest);

// Repeat BatchPutDocument and BatchDeleteDocument requests for all the documents in
your
// repository to sync with Amazon Kendra.

// After you are finished, call the StopDataSourceSyncJob operation to signal the end
of the sync job.
final StopDataSourceSyncJobResult stopDataSourceSyncJobResult =
awsKendraClient.stopDataSourceSyncJob(
    new StopDataSourceSyncJobRequest()

```

```
        .withIndexId(indexId)
        .withId(datasourceId)
    );

    // After you call the StopDataSourceSyncJob operation, you can start another sync job.
    // You can't use the BatchPutDocument or BatchDeleteDocument operation requests with a
    // stopped job execution ID.

    // It can take time to index all of the documents submitted. Use the
    ListDataSourceSyncJobs
    // operation to get the status of a sync job and number of documents added, modified,
    failed
    // or deleted as part of this sync with your Amazon Kendra index.

    // If the sync job status is SYNCING_INDEXING, documents are still being indexed.

    // Jobs are sorted in reverse order of their start time with the most recent first.
    final ListDataSourceSyncJobsResult listDataSourceSyncJobsResult =
awsKendraClient.listDataSourceSyncJobs(
    new ListDataSourceSyncJobsRequest()
        .withIndexId(indexId)
        .withId(datasourceId)
    );
}

}
```

Using a database data source

You can index documents stored in a database using a database data source. After you provided connection information for the database, Amazon Kendra connects and indexes documents. Amazon Kendra supports the following databases:

- Amazon Aurora MySQL
- Amazon Aurora PostgreSQL
- Amazon RDS for MySQL
- Amazon RDS for PostgreSQL

Note, serverless Aurora databases are not supported.

Before you create a database data source, you need to create an index and create custom fields in the index for the data from the database. For more information, see [Creating an index \(p. 64\)](#) and [Mapping data source fields \(p. 123\)](#).

To use a database data source, you need to identify the following:

- Connection information such as credentials for the database stored in AWS Secrets Manager, the host name, port, and name of the data table that contains the document data. For PostgreSQL the data table must be a public table.
- Column information such as the names of the columns in the data table that contain the document data and document ID, one to five columns to detect if a document has changed, and optional data table columns that map to custom index fields. You can map any of the Amazon Kendra reserved field names to a table column.
- Optionally, VPC information to connect to the database server. For more information about using a VPC, see [Configuring Amazon Kendra to use a VPC \(p. 125\)](#). If you are using a database data source with a VPC, the subnets provided in the VPC configuration must be in one of the following Availability Zone IDs:
 - US West (Oregon) – usw2-az1, usw2-az2, usw2-az3

- US East (N. Virginia) – use1-az1, use1-az2, use1-az4
- EU (Ireland) – euw1-az1, uew1-az2, euw1-az3

Database configuration provides the information required to connect to your database server. The host and port tell Amazon Kendra where to find the database server on the internet. The database name and table name tell Amazon Kendra where to find the document data on the database server.

To enable Amazon Kendra to access your documents, specify a user that has read access to the table that contains the documents. Amazon Kendra requires credentials for the user to access the database. You provide these credentials using AWS Secrets Manager. Once you have created the secret, you provide the Amazon Resource Name (ARN) of the secret to Amazon Kendra. The secret must contain the user name and password that Amazon Kendra uses to access the database in a JSON structure. The secret may contain additional information, but Amazon Kendra uses only the user name and password. The following is the minimum JSON structure that must be in the secret:

```
{  
    "username": "user_name",  
    "password": "password"  
}
```

The secret can contain more information. However, Amazon Kendra ignores other fields. For more information, see [What Is AWS Secrets Manager](#) in the *AWS Secrets Manager User Guide*.

The following example shows a database configuration.

```
"DatabaseConfiguration": {  
    "ConnectionConfiguration": {  
        "DatabaseHost": "host.subdomain.domain.tld",  
        "DatabaseName": "DocumentDatabase",  
        "DatabasePort": 3306,  
        "SecretArn": "arn:aws:secretmanager:region:account ID:secret/secret name",  
        "TableName": "DocumentTable"  
    }  
}
```

Note

The `DatabaseHost` field should be the Amazon Relational Database Service (Amazon RDS) instance endpoint for the database. Don't use the cluster endpoint.

By default, Amazon Kendra uses SQL identifiers—such as database name, table name, and column names—exactly as set in the database data source configuration. Amazon Kendra doesn't enclose identifiers in quotation marks or change the case.

A PostgreSQL database always changes unquoted table and column names to lowercase. For example, if Amazon Kendra is configured to use the table name `SAMPLE_TABLE`, PostgreSQL converts it internally to `sample_table`. If a table or column name contains uppercase letters, the SQL query won't match the correct columns or table because PostgreSQL internally changes them to lowercase.

To configure Amazon Kendra to enclose the SQL identifiers for table and column names in quotation marks ("), set the `QueryIdentifiersEnclosingOption` field to `DOUBLE_QUOTES` inside the [SqlConfiguration](#) (p. 542) parameter of the [CreateDataSource](#) (p. 283) operation. When you set this parameter, the SQL identifiers sent to databases are enclosed in quotation marks so that PostgreSQL doesn't change SQL identifiers to lowercase. If you enclose identifiers in quotation marks when you are using MySQL, you must set the `ansi_quotes` option in the MySQL database.

You add document table information to an index by mapping table columns to index fields. There are two types of information that you add. The first is one to five columns that Amazon Kendra uses to determine if a document has changed since the last time that an index update was run. For example, if

you have columns in your table named `LastUpdateDate` and `LastUpdateTime`, you can tell Amazon Kendra to use them to determine if a document was updated.

The second type of information about columns is to map some or all of the columns in your table to index fields. For example, you can map a column that contains the document abstract to an index field. If you mark the field searchable, Amazon Kendra uses the contents of the field when determining if a document matches the query. For more information about the attributes that you can assign a custom field, see [Mapping data source fields \(p. 123\)](#).

After you map the columns, you can also use the index fields as custom attributes to filter the results of a query. For more information, see [Filtering queries \(p. 134\)](#).

Specify the `DocumentDataColumnName` and `DocumentIdColumnName` fields. The column mapped to the `DocumentIdColumnName` field must be an integer column.

The following example shows a simple column configuration for a database data source.

```
"ColumnConfiguration": {  
    "ChangeDetectingColumns": [  
        "LastUpdateDate",  
        "LastUpdateTime"  
    ],  
    "DocumentDataColumnName": "TextColumn",  
    "DocumentIdColumnName": "IdentifierColumn",  
    "DocoumentTitleColumnName": "TitleColumn",  
    "FieldMappings": [  
        {  
            "DataSourceFieldName": "AbstractColumn",  
            "IndexFieldName": "Abstract"  
        }  
    ]  
}
```

Using a Google Workspace Drive data source

You can use an Amazon Kendra data source to connect to Google Workspace Drive to index the documents stored there. For a walk-through of how to use Google Workspace Drive in the console, see [Getting started with a Google Workspace Drive data source \(console\)](#).

Amazon Kendra indexes the documents stored in shared drives as well as the documents stored in user My Drives. By default, Amazon Kendra indexes all documents in your Google Drive. You can exclude documents from the index based on the ID of a shared drive, the user that owns the document, the MIME type of the document, or their path.

The Google Drive data source indexes Google Workspace documents as well as the documents listed in [Types of documents \(p. 5\)](#). The supported Google Workspace document types are:

- Google Docs
- Google Slides

To enable Amazon Kendra to connect to a Google Drive, you must first use an administrator account to create a service account. The service account must have read-only permission for the user and shared drives that you want to index. The account needs the following permissions:

- <https://www.googleapis.com/auth/drive.readonly>
- <https://www.googleapis.com/auth/drive.metadata.readonly>
- <https://www.googleapis.com/auth/admin.directory.user.readonly>
- <https://www.googleapis.com/auth/admin.directory.group.readonly>

Once you have created the service account, download the key file to your computer. You send this key to Amazon Kendra when you create the data source.

You store the Google Drive credentials in an AWS Secrets Manager secret. If you are using the Amazon Kendra console to create your data source, you can create the secret while creating the data source, or you can use an existing Secrets Manager secret. If you are using the API to create your data source, you must provide the Amazon Resource Name (ARN) of an existing secret.

The secret must contain the following information:

- Client (service) account
- Admin account
- Private key

The credentials are stored in a JSON string in the Secrets Manager secret as shown in the following example.

```
{  
    "clientAccount": "account email",  
    "adminAccount": "account email",  
    "privateKey": "private key"  
}
```

The data source IAM role must have permissions to access the AWS Key Management Service key used to decrypt it. For more information, see [IAM roles for Google Workspace Drive data sources \(p. 19\)](#).

You must create an index before you create the Google Drive data source. For more information, see [Creating an index \(p. 64\)](#). You provide the ID of the index when you create the data source.

You specify connection and other information in the console or in an instance of the [GoogleDriveConfiguration \(p. 483\)](#) data type. You must provide the following information:

- The Amazon Resource Name (ARN) of a AWS Secrets Manager secret that contains the credentials required for Amazon Kendra to connect.

After you create the data source, you can:

- Modify the Secrets Manager secret containing the credentials required to access Google Workspace.
- Modify the list of user accounts to exclude from indexing.
- Modify the list of shared drives to exclude from indexing.
- Modify the include and exclude regular expressions.
- Modify the MIME types to exclude from indexing.

After you sync the data source, you can't remove the field mappings. You can map additional fields.

By default, Amazon Kendra indexes all supported documents stored on your Google Workspace Drive and any My Drives for your users. You can exclude documents using the console or by setting the following fields of the [GoogleDriveConfiguration \(p. 483\)](#) parameter when you create the data source. Exclusions are combined with a logical AND, so if a file matches any of the exclusions it is not included in the index.

- **ExcludeMimeTypes** – One or more MIME types of the documents to exclude. For example, if you specify the MIME type for Microsoft Word documents, those documents aren't indexed.
- **ExcludeSharedDrives** – One or more shared drive identifiers to exclude. None of the files on the shared drive are indexed.

- **ExcludeUserAccounts** – One or more email addresses of user accounts to exclude from the index. None of the files in the My Drive owned by the account are indexed. Files shared with the user are indexed unless the owner of the file is also excluded.
- **ExclusionPatterns** – One or more regular expressions. Files that match the pattern aren't indexed.
- **InclusionPatterns** – One or more regular expressions. Files that match the pattern are indexed, all other files are excluded from the index. Any file that matches another exclusion isn't indexed even if it matches the inclusion pattern.

You can map Google Drive properties to Amazon Kendra index fields. The following table shows the Google Drive properties that can be mapped and a suggested Amazon Kendra index field.

Google Drive property name	Suggested Amazon Kendra field name
createdTime	_created_at
dataSize	gd_data_size
displayUrl	gd_source_url
fileExtension	_file_type
id	_document_id
imeType	gd_mime_type
modifiedTime	_last_updated_at
name	_document_title
owner	gd_owner
version	gd_version

The Google Drive API enables you to create custom file properties using key/value pairs. You can map these custom properties to Amazon Kendra index fields. You must prefix the name of the field with the string "property." when you specify the field name. For example, a custom property called "author" is specified as "property.author".

Using a Microsoft OneDrive data source

Amazon Kendra can use a data source to connect to Microsoft OneDrive sites to index the documents that your users create. For a walk-through of how to use OneDrive in the console, see [Getting started with a Microsoft OneDrive for Business data source \(console\)](#).

When you use a OneDrive data source to connect Amazon Kendra to your OneDrive site, you choose the users whose documents are indexed. You can optionally provide inclusion and exclusion patterns to specify the documents to index.

To create a OneDrive data source, you must first create an Azure Active Directory (AD) application that Amazon Kendra connects to. You must grant the application the following permissions on the Microsoft Graph option:

- Read files in all site collections (File.Read.All)
- Read all users' full profile (User.Read.All)
- Read directory data (Directory.Read.All)
- Read all groups (Group.Read.All)

- Read items in all site collections (Site.Read.All)

When you create the AD application, it is assigned an application ID. You must use the AD site to register a secret key for the application. Amazon Kendra uses the ID and key as credentials to authenticate when it connects to the OneDrive site. You store the ID and key in an AWS Secrets Manager secret. If you are using the console to create your OneDrive data source, you can enter the credentials there to create a Secrets Manager secret. Or you can choose an existing Secrets Manager secret. If you are using the API, you must provide the Amazon Resource Name (ARN) of an existing secret.

The secret must contain the application ID and secret key that Amazon Kendra uses to access the site in a JSON structure. The following is the minimum JSON structure that must be stored in the secret:

```
{  
    "username": "application ID",  
    "password": "secret key"  
}
```

The data source AWS Identity and Access Management (IAM) role must have permission to access the secret. For more information, see [IAM roles for Microsoft OneDrive data sources \(p. 20\)](#).

The secret can contain more information, but Amazon Kendra ignores it. For more information, see [What is AWS Secrets Manager](#) in the *AWS Secrets Manager User Guide*.

You must create an index before you create the OneDrive data source. For information, see [Creating an index \(p. 64\)](#). You provide the ID of the index when you create the data source.

You specify connection and other information in the console or using an instance of the [OneDriveConfiguration \(p. 499\)](#) data type. You must provide the following information:

- The credentials required to log in to the OneDrive site.
- The tenant domain that contains the OneDrive site.
- A list of users whose documents should be indexed. You can provide a list of user names, or you can provide the user names in a file stored in an Amazon Simple Storage Service (Amazon S3) bucket. If you store the list of user names in an S3 bucket, the IAM policy for the data source must provide access to the bucket and access to the key that the bucket was encrypted with, if any.

After you create a data source, you can:

- Modify the list of users.
- Change from a list of users to a list stored in an S3 bucket.
- Change the S3 bucket location of a list of users. If you change the bucket location, you must also update the IAM role for the data source so that it has access to the bucket.
- Change the content of a user list stored in an S3 bucket.

You can optionally provide the following information:

- A list of inclusion and exclusion regular expressions that filter the documents that are included in the index. The regular expressions are applied to the file name of the document.
- Field mappings that map fields from your OneDrive site to Amazon Kendra index fields. For information, see [Mapping data source fields \(p. 123\)](#).

After you sync the data source, you can't change the inclusion and exclusion patterns or the remove field mapping. You can map additional fields.

You can map OneDrive properties to Amazon Kendra index fields. The following table shows the OneDrive properties that can be mapped and a suggested Amazon Kendra index field.

OneDrive field name	Suggested Amazon Kendra field name
body	_document_body
createdDateTime	_created_at
name	_document_title
webUrl	_document_id
createdBy.displayName	od_createdBy_displayName
createdBy.id	od_createdBy_id
createdBy.email	oc_createdBy_email
cTag	od_ctag
eTag	od_etag
fileSystemInfo.createdDateTime	od_fileSystemInfo_createdDateTime
fileSystemInfo.lastAccessedDateTime	od_fileSystemInfo_lastAccessedDateTime
fileSystemInfo.lastModifiedDateTime	od_fileSystemInfo_lastModifiedDateTime
file.mimeType	od_file_mimeType
lastModifiedDateTime	_last_updated_at
lastModifiedBy.displayName	od_lastModifiedBy_displayName
lastModifiedBy.id	od_lastModifiedBy_id
lastModifiedBy.email	od_lastModifiedBy_email
size	od_size
webDavUrl	oe_webDavUrl

Using a Salesforce data source

Amazon Kendra can connect to your Salesforce server to index your customer relationship information. For a walk-through of how to use Salesforce in the console, see [Getting started with a Salesforce data source \(console\)](#).

When you use Amazon Kendra to index your Salesforce server, you can choose to index up to 17 of the standard Salesforce objects. You can also index knowledge articles, chatter feeds, and attachments.

Amazon Kendra uses the Salesforce API version 48. The Salesforce API limits the number of requests that you can make per day. If Amazon Kendra exceeds those requests, it retries until it is able to continue.

Before you can connect Amazon Kendra to your Salesforce server, you must create a Salesforce connected app with OAuth enabled so that Amazon Kendra can connect. When you create an app, it is assigned a consumer key and a consumer secret that Amazon Kendra uses to connect to the app.

You must provide Amazon Kendra with credentials to access your Salesforce server. These credentials identify the user making the connection and the Salesforce connected app that Amazon Kendra connects to.

The credentials should be for a user with read-only access to Salesforce. To create permissions for the user, clone the ReadOnly profile and then add the View All Data and Manage Articles permissions.

You store the credentials in an AWS Secrets Manager secret. If you are using the console to create your data source, you can create the secret there, or you can use an existing Secrets Manager secret. If you are using the API, you must provide the Amazon Resource Name (ARN) of an existing secret.

The secret must contain the following information:

- `authenticationUrl` – The URL of the OAuth authentication server used to authenticate with Salesforce. Typically, this is `https://login.salesforce.com/services/oauth2/token`.
- `consumerKey` – The consumer key, also called the client ID, of the Salesforce Connected App that is used to index the server. The app must have permission that allows access to the REST API.
- `consumerSecret` – The consumer secret, also called the client secret, of the Salesforce Connected App used to index the server.
- `securityToken` – The Salesforce security token associated with the account used to connect to Salesforce.
- `password` – The password associated with the account used to connect to Salesforce.
- `username` – The user name of the account used to connect to Salesforce. The account must have read access to the objects and fields that you want to index.

The credentials are stored as a JSON string in the Secrets Manager secret. The following is the minimum JSON structure that must be in the secret:

```
{  
    "username": "user_name",  
    "password": "password",  
    "securityToken": "token",  
    "consumerKey": "key",  
    "consumerSecret": "secret",  
    "authenticationUrl": "https://login.salesforce.com/services/oauth2/token"  
}
```

The data source IAM role must have permission to access the secret. For more information, see [IAM role for Salesforce data sources \(p. 21\)](#).

The secret can contain more information, however, Amazon Kendra ignores other fields. For more information, see [What is AWS Secrets Manager](#) in the *AWS Secrets Manager User Guide*.

You must create an index before you create the Salesforce data source. For more information, see [Creating an index \(p. 64\)](#). You provide the ID of the index when you create the data source.

You specify connection and other information in the console or using an instance of the [SalesforceConfiguration \(p. 516\)](#) data type. You must provide the following information:

- The URL of the Salesforce server that contains the information to index.
- The credentials required to connect to the Salesforce server.

You must provide configuration information for indexing at least one of the following:

- Salesforce objects
- Salesforce knowledge articles

- Salesforce chatter feeds

You can optionally:

- Provide configuration information for indexing attachments.
- Indicate whether Amazon Kendra should gather access control information for user context filtering.

Standard objects

Salesforce provides an extensive list of standard objects that contain information about your customer relations. You can choose to index any of these standard objects:

- Account
- Campaign
- Case
- Contact
- Contract
- Chatter
- Document
- Group
- Idea
- Lead
- Opportunity
- Partner
- Pricebook
- Product
- Profile
- Solution
- Task
- User

For each object, you must map an object field to the Amazon Kendra built-in `_body` field so that Amazon Kendra knows where to find the object content to index. You can map additional object fields to custom Amazon Kendra fields.

Salesforce enables you to add custom fields to standard objects. To use the custom field with Amazon Kendra, you must use the internal Salesforce field name. The internal name is the name of the field followed by `"_c"` (two underscores and the character c). For example, if you have a custom field named `AccountOriginalOwner`, the internal name is `AccountOriginalOwner__c`.

You can map fields from multiple objects to a single Amazon Kendra field. For example, you can map the `Account` object `Name` field and the `Partner` object `Name` field to the same Amazon Kendra custom field.

Once you save the mapping between an Amazon Kendra field and a Salesforce object field, you can't change the mapping. However, you can add more mappings between Amazon Kendra and Salesforce.

For more information, see [Mapping data source fields \(p. 123\)](#).

Knowledge articles

You can use Amazon Kendra to index the contents of standard knowledge articles or custom knowledge articles.

When you index standard knowledge articles, Amazon Kendra will index every article on your server, including the standard fields of custom knowledge articles. If you index custom knowledge articles, Amazon Kendra indexes only articles of that type. It won't index the contents of standard knowledge articles.

You configure indexing of knowledge articles using the console or the [SalesforceKnowledgeArticleConfiguration \(p. 521\)](#) object. You can indicate the status of the articles that you want to index, you can tell Amazon Kendra to index draft, published, or archived articles.

For custom knowledge articles, you must specify the name of the custom article type. You must specify the internal name of the article type, which is the name of the type plus "__kav" (two underscores followed by the characters kav). For example, if you have a customer article type called CustomKnowledgeArticleForTech, the internal name is CustomKnowledgeArticleForTech__kav. You can specify up to 10 article types.

For both custom and standard knowledge articles, you must specify the name of the field that contains the content of the article. You can optionally specify the field that contains the title. You can map additional article fields to custom Amazon Kendra fields using the console or the [DataSourceToIndexFieldMapping \(p. 467\)](#) object.

Chatter feeds

You can index the contents of your Salesforce chatter feeds. You configure indexing using the console or the [SalesforceChatterFeedConfiguration \(p. 514\)](#) object.

You must specify the field in the Salesforce FeedItem table that contains the content of the item. Typically this is the "Body" column. You have the option of specifying the title of the item. Typically, this is the "Title" column of the FeedItem table. You can map additional fields to custom Amazon Kendra fields using the console or the [DataSourceToIndexFieldMapping \(p. 467\)](#) object.

By default, Amazon Kendra indexes all items on the chatter feed. You can use the console or the `IncludeFilterType` field of the [SalesforceChatterFeedConfiguration](#) object to limit indexing to only those items that are from standard Salesforce users or from active user accounts.

You can map additional fields to custom Amazon Kendra fields using the console or the [DataSourceToIndexFieldMapping \(p. 467\)](#) object.

Attachments

You can choose to have Amazon Kendra index attachments to standard objects, knowledge articles, and chatter feeds. You can use the console or the `CrawlAttachments` option on the [SalesforceConfiguration \(p. 516\)](#) structure to indicate whether attachments should be indexed.

By default, Amazon Kendra indexes all attachments. You can use the console or the API to filter attachments from the list that is indexed. To filter an attachment, you use a regular expression that is evaluated against the file name of the attachment. For example, to remove JSON files from the list of indexed files, use a regular expression that filters out files that end with ".json".

You can also restrict indexed documents by specifying the attachments to index. For example, to index only Microsoft Word files, specify a regular expression that selects files that end with ".doc" or ".docx."

Using a ServiceNow data source

Amazon Kendra can connect to your ServiceNow instance to index your knowledge bases and a service catalog. For a walk-through of how to use ServiceNow in the console, see [Getting started with a ServiceNow data source \(console\)](#).

When you use Amazon Kendra to index a ServiceNow instance, you can choose to index public knowledge bases and public service catalogs, or you can use a query to index specific knowledge bases,

including private ones. You can connect to a ServiceNow instance using basic authentication and a ServiceNow administrative user, or you can use an OAuth 2.0 token and a user with read permission on instance tables.

For public knowledge bases in your ServiceNow instance, Amazon Kendra indexes only public articles. A public knowledge base must have the public role under **Can Read**, and **Cannot Read** must be null or not set.

Authentication

There are two forms of authentication that you can use with ServiceNow. The first, basic authentication, enables Amazon Kendra to connect to the ServiceNow instance using a user name and password. The user must have administrative permissions to the ServiceNow instance.

The second, OAuth, uses a token that follows the OAuth 2.0 authentication specification to identify Amazon Kendra and a user name and password. The user name and password must provide access to the ServiceNow knowledge base and service catalog.

Basic authentication

When you use basic authentication, you provide the user name and password of an administrative user of your ServiceNow instance. Amazon Kendra uses these credentials to connect to ServiceNow.

The user name and password for the ServiceNow instance must be stored in an AWS Secrets Manager secret. If you use the Amazon Kendra console, you can enter the ServiceNow credentials there to create a new secret, or you can choose an existing secret. If you use the Amazon Kendra API, you must provide the Amazon Resource Name (ARN) of an existing secret that contains your ServiceNow user name and password.

The secret must contain the username and password of the ServiceNow account that you want Amazon Kendra to use to access ServiceNow. The following is the minimum JSON structure that must be stored in the secret.

```
{  
    "username": "user-name",  
    "password": "password"  
}
```

The secret can contain other information, but Amazon Kendra ignores it. For more information, see [What is Secrets Manager](#) in the *AWS Secrets Manager User Guide*.

When you create the ServiceNow data source, you specify an IAM role that grants Amazon Kendra permission to access resources required to index your ServiceNow instance. The data source IAM role must have permission to access the secret and to use the AWS Key Management Service (AWS KMS) key to decrypt it. For more information, see [IAM role for ServiceNow data sources \(p. 22\)](#).

OAuth authentication

When you use OAuth authentication to connect to ServiceNow, you provide the client ID and secret that identifies Amazon Kendra to ServiceNow. You also provide a user name and password that is used to access the knowledge bases and service catalog.

The client ID, client secret, user name, and password must be stored in an AWS Secrets Manager secret. If you use the Amazon Kendra console, you can enter the ServiceNow credentials there to create an Secrets Manager secret, or you can choose an existing secret. If you use the Amazon Kendra API, you must provide the Amazon Resource Name (ARN) of an existing Secrets Manager secret that contains your ServiceNow credentials.

The secret must contain the user name, password, client ID and client secret for your ServiceNow instance. The following is the minimum JSON structure that must be stored in the Secrets Manager secret.

```
{  
    "username": "user-name",  
    "password": "password",  
    "clientId": "client-id",  
    "clientSecret": "client-secret"  
}
```

The secret can contain other information, but Amazon Kendra ignores it. For more information, see [What is Secrets Manager](#) in the *AWS Secrets Manager User Guide*.

Generating the client ID and secret

You generate the OAuth client ID and secret using the ServiceNow console and then copy them to the Amazon Kendra console. Create the client ID and secret using the following procedure.

To create a ServiceNow client ID and secret

1. Log in to the ServiceNow console.
2. From the left menu, choose **System OAuth** and then choose **Application Registry**.
3. Choose **New** to create a new registry.
4. For the type of OAuth application, choose **Create an OAuth application endpoint for external clients**.
5. Enter a name.
6. You can enter your own client secret, or you can have ServiceNow generate one for you. Leave the defaults for the other fields.
7. Choose **Submit** to create the registry containing the OAuth client secret and ID.
8. After the registry is created, choose it from the list of registries.
9. Copy the client secret and ID. You'll need them when you create the ServiceNow data source.

Table permissions

When you use OAuth authentication, you provide a user name and password. Amazon Kendra sends the user name and password to ServiceNow. ServiceNow uses them to determine the access that Amazon Kendra has to the ServiceNow instance. To index a knowledge base and service catalog, the user must have read permission on the following tables.

- kb_category
- kb_knowledge
- kb_knowledge_base
- kb_uc_cannot_read_mtom
- kb_uc_can_read_mtom
- sc_catalog
- sc_category
- sc_cat_item
- sys_attachment
- sys_attachment_doc
- sys_user_role

Connecting to a ServiceNow instance

You provide ServiceNow connection information in the Amazon Kendra console or by using an instance of the [ServiceNowConfiguration \(p. 530\)](#) data type. You must provide the following information:

- The ARN of the Secrets Manager secret that contains the credentials required to access the ServiceNow instance.
- The version of the ServiceNow instance. For Amazon Kendra, this is LONDON for the London version and OTHERS for all other versions.
- The ServiceNow instance host. For example, if the URL of the instance is `https://your-domain.service-now.com`, the host is `your-domain.service-now.com`.
- Whether to index knowledge articles, service catalogs, or both. You must also provide the name of the ServiceNow field that contains the document body.

You can optionally provide the following information:

- A ServiceNow query that selects documents from one or more knowledge bases. The knowledge bases can be public or private. For more information, see [Specifying documents to index with a query \(p. 114\)](#).
- The name of the ServiceNow field that contains the document title. This is typically the ServiceNow title field. If you don't specify a document title field, Amazon Kendra uses the document ID as the title.
- Whether Amazon Kendra should index attachments to knowledge base or catalog items. If you choose to index attachments, you can specify the file type of attachments to exclude from the index.
- Field mappings that map fields in your ServiceNow instance to fields in your Amazon Kendra index. For more information, see [Mapping data source fields \(p. 123\)](#).
- An inclusion pattern to specify the file type of document attachments to include in the index. If you specify an inclusion pattern, any attachment that doesn't match the pattern isn't indexed. If the pattern includes file types that Amazon Kendra does not support, those files won't be included. For a list of supported file types, see [Types of documents \(p. 5\)](#).
- An exclusion pattern to specify the file type of document attachments to exclude from the index. If you specify an exclusion pattern, any attachment that doesn't match the pattern is indexed.

If you specify both an inclusion and an exclusion pattern, attachments that match the exclusion pattern won't be indexed even if they match the inclusion pattern.

After you sync the data source for the first time, the inclusion and exclusion patterns can't be changed.

You can map ServiceNow properties to Amazon Kendra index fields. The following table shows the ServiceNow knowledge article properties that can be mapped and a suggested Amazon Kendra index field. You can also create custom ServiceNow fields that you map to Amazon Kendra index fields.

ServiceNow field name	Suggested Amazon Kendra field name
content	_document_body
displayUrl	sn_display_url
first_name	sn_ka_first_name
kb_category	sn_ka_category
kb_catagory_name	_category

ServiceNow field name	Suggested Amazon Kendra field name
kb_knowledge_base	sn_ka_knowledge_base
last_name	sn_ka_last_name
number	sn_kb_number
published	sn_ka_publish_date
replItemType	sn_replItemType
short_description	_document_title
sys_created_by	sn_createdBy
sys_created_on	_created_at
sys_id	sn_sys_id
sys_updated_by	sn_updatedBy
sys_updated_on	_last_updated_at
url	sn_url
user_name	sn_ka_user_name
valid_to	sn_ka_valid_to
workflow_state	sn_ka_workflow_state

The following table shows the ServiceNow catalog properties that can be mapped and a suggested Amazon Kendra field.

ServiceNow field name	Suggested Amazon Kendra field name
category	sn_sc_category
category_full_name	sn_sc_category_full_name
category_name	_category
description	_document_body
displayUrl	sn_display_url
replItemType	sn_replItemType
sc_catalogs	sn_sc_catalogs
sc_catalogs_name	sn_sc_catalogs_name
short_description	_document_body
sys_created_by	sn_createdBy
sys_created_on	_created_at
sys_id	sn_sys_id

ServiceNow field name	Suggested Amazon Kendra field name
sys_updated_by	sn_updatedBy
sys_updated_on	_last_updated_at
title	_document_title
url	sn_url

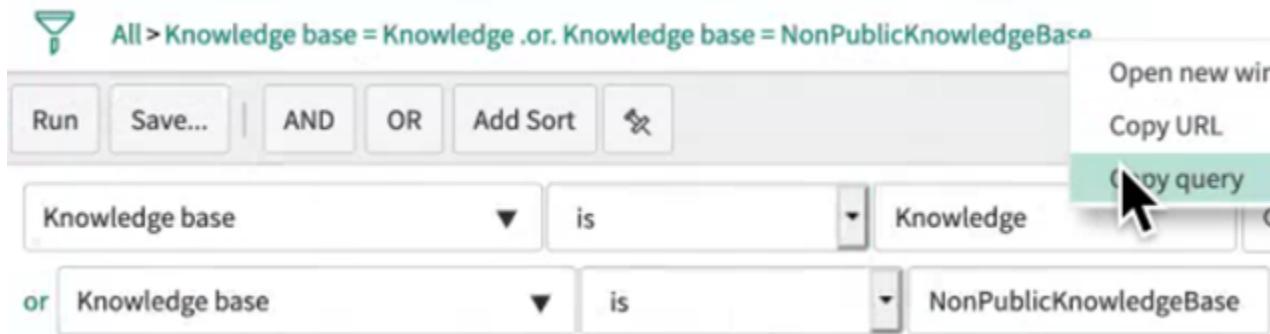
Specifying documents to index with a query

You can use a ServiceNow query to specify the documents you want to include in a Amazon Kendra index. When you use a query, you can specify multiple knowledge bases, including private knowledge bases. Access to the knowledge bases is determined by the user that you use to connect to the ServiceNow instance.

To build a query, you use the ServiceNow query builder. You can use the builder to create the query and to test that the query returns the correct list of documents.

To create a query using the ServiceNow console

1. Log in to the ServiceNow console.
2. From the left menu, choose **Knowledge**, then **Articles**, and then choose **All**.
3. At the top of the page, choose the filter icon.
4. Use the query builder to create the query.
5. When the query is complete, right click the query and choose **Copy query** to copy the query from the query builder. Save this query to use in Amazon Kendra.



Be careful that you don't change any query parameter when you copy the query. If any of the query parameters are not recognized, ServiceNow treats the parameter as empty and doesn't use it to filter the results.

Using a Microsoft SharePoint data source

You can use your Microsoft SharePoint as a data source for Amazon Kendra. For a walk-through of how to use SharePoint in the console, see [Getting started with a Microsoft SharePoint data source \(console\)](#).

When you use Amazon Kendra to index your site, you choose which SharePoint URLs to include in the index, and you specify inclusion and exclusion patterns for the documents stored on those URLs.

Amazon Kendra currently supports SharePoint Online and SharePoint Server (versions 2013 and 2016).

Amazon Kendra requires authentication credentials to access the SharePoint site. If you are using the console to create your data source, you can enter the credentials there or you can choose an existing AWS Secrets Manager secret that stores your credentials. If you are using the API, you must provide the Amazon Resource Name (ARN) of an existing secret. For more information, see [What Is AWS Secrets Manager](#) in the [AWS Secrets Manager User Guide](#).

If you use SharePoint Online, you only need to provide your user name and password in your secret.

If you use SharePoint Server, in addition to your user name and password, you need to provide the server domain name. The server domain name is the NetBIOS name in your active directory provider.

If you use SharePoint Server and need to convert your access control list (ACL) to email format for filtering on user context, you need to provide the LDAP server URL and LDAP search base, or use the directory domain override. The LDAP server URL is the full domain name and the port number. For example, 'ldap://example.com:389'. The LDAP search base are the domain controllers 'example' and 'com'. The directory domain override allows you to use the email domain instead of using LDAP server URL and LDAP search base. For example, the email domain for 'username@example.com' is 'example.com'. You can use this override if you are not concerned about validating your domain and simply want to use your email domain.

Amazon Kendra uses the credential information stored in your secret to access the SharePoint site in a JSON structure.

If you use SharePoint Online, the following is the minimum JSON structure that must be in your secret:

```
{  
    "username": "user name",  
    "password": "password"  
}
```

The secret can contain more information, however, Amazon Kendra ignores other fields.

If you use SharePoint Server, the following is the minimum JSON structure that must be in your secret:

```
{  
    "username": "user name",  
    "password": "password",  
    "domain": "server domain name"  
}
```

You can include the LDAP server URL and LDAP search base in your secret for SharePoint Server using the following JSON structure:

```
{  
    "username": "user name",  
    "password": "password",  
    "domain": "server domain name",  
    "ldapServerUrl": "ldap://example.com:389",  
    "ldapSearchBase": "dc=example,dc=com",  
    "directoryDomainOverride": "example.com"  
}
```

The SharePoint user must have administrative permission to the SharePoint sites that you want to index. For SharePoint lists, the user must have the following permissions:

- Open Items – View the source of documents with server-side file handlers.
- View Application Pages – View forms, views and application pages. Enumerate lists
- View Items – View items in lists and documents in document libraries

- View Versions – View past versions of a list item or document.

For SharePoint websites, the user must have the following permissions:

- Browse Directories – Enumerate files and folders in a website using SharePoint Designer and Web DAV interfaces.
- Browse User Information – View information about users of the website.
- Enumerate Permissions – Enumerate permissions on the website, list, folder, document, or list item.
- Open – Open a website, list, or folder to access items inside the container.
- Use Client Integration Features – Use SOAP, WebDAV, the client object model, or SharePoint Designer interfaces to access the website.
- Use Remote Interfaces – Use features that launch client applications.
- View Pages – View pages on a website.

The data source IAM role must have permission to access the secret. For more information, see [IAM roles for Microsoft SharePoint data sources \(p. 23\)](#).

You must create an index before you create the SharePoint data source. For more information, see [Creating an index \(p. 64\)](#). You provide the ID of the index when you create the data source.

You specify connection and other information in the console or using an instance of the [SharePointConfiguration \(p. 536\)](#) data type. You must provide the following information:

- The credentials required to log in to the SharePoint site.
- The URLs of the SharePoint site, SharePoint site collection, or SharePoint list to index.
- The ARN of an IAM role that has permission to run Amazon Kendra commands. For the required permissions, see [IAM roles for Microsoft SharePoint data sources \(p. 23\)](#).

You can optionally provide the following information:

- Whether Amazon Kendra should index the contents of attachments to SharePoint list items.
- An inclusion pattern to specify the documents that should be included in the index. If you specify an inclusion pattern, any document that does not match the pattern is not indexed.
- An exclusion pattern to specify the documents that should be excluded from the index. If you specify an exclusion pattern, any document that does not match the pattern will be indexed. If you specify both an inclusion and an exclusion pattern, documents that match the exclusion pattern are not indexed even if they match the inclusion pattern.
- Whether Amazon Kendra should use the SharePoint change log mechanism to determine if a document needs to be updated in the index. You should use the change log if you don't want Amazon Kendra to scan all of the documents in the site to update the index. If your change log is large, it may take Amazon Kendra less time to scan the site than to process the change log.
- Field mappings that map attributes from your SharePoint site to Amazon Kendra index fields. For more information, see [Mapping data source fields \(p. 123\)](#).

Using a web crawler data source

You can use Amazon Kendra *Web Crawler* to crawl and index webpages. For a walk-through of how to use the web crawler in the console, see [Getting started with Amazon Kendra Web Crawler \(console\)](#).

You can use the web crawler to crawl webpages and index them as your documents. For the websites you want to crawl and index, you provide either the seed or starting point URLs or the sitemap URLs. You

can only crawl websites that use the secure communication protocol, Hypertext Transfer Protocol Secure (HTTPS). If you receive an error when crawling a website, it could be that the website is blocked from crawling.

You can configure the following crawl settings:

- The range of websites to crawl: website host names only, or websites including subdomains, or websites including subdomains and others domains that the webpages link to.
- The *depth* or number of levels in a website from the seed level to crawl. For example, if a website has 3 levels – index level or the seed level in this example, sections level, and subsections level – and you are only interested in crawling information from the index level to the sections level (levels 0 to 1), you can set your depth to 1.
- The maximum number of URLs on a single webpage that are crawled.
- The maximum size in MB of a webpage to crawl.
- The maximum number of URLs crawled per website host per minute.
- Regular expression patterns to include or exclude certain URLs to crawl.
- The web proxy information to connect to and crawl internal websites.
- The authentication information to access and crawl websites that require user authentication.

You can configure the web crawler using the [CreateDataSource](#) operation. You provide the web crawler configuration information in the [WebCrawlerConfiguration](#) structure.

You use the [SeedUrlConfiguration](#) structure to provide a list of seed URLs and choose whether to crawl only website host names, or include subdomains, or include subdomains and other domains the webpages link to. You also use the [SiteMapsConfiguration](#) structure to provide a list of sitemap URLs.

When selecting websites to index, you must adhere to the [Amazon Acceptable Use Policy](#) and all other Amazon terms. Remember that you must only use Amazon Kendra Web Crawler to index your own webpages, or webpages that you have authorization to index. To learn how to stop Amazon Kendra Web Crawler from indexing your website(s), please see [Stopping Amazon Kendra Web Crawler from indexing your website](#).

Website user authentication

Before connecting to the web crawler, you need to check if the websites you want to crawl require authentication to access the websites. If a website requires basic authentication, you provide the web crawler with the host name of the website, the port number, and a secret in [AWS Secrets Manager](#) that stores your basic authentication credentials of your user name and password.

If you use the Amazon Kendra console, you can choose an existing secret. If you use the Amazon Kendra API, you must provide the Amazon Resource Name (ARN) of an existing secret that contains your user name and password. You can create a secret in [AWS Secrets Manager](#).

The secret must contain the user name and password of the website that you want to crawl. The following is the minimum JSON structure that must be stored in the secret.

```
{  
    "username": "user-name",  
    "password": "password"  
}
```

The secret can contain other information, but Amazon Kendra ignores it.

You use the [AuthenticationConfiguration](#) structure to provide the website host name, website port number, and the secret that stores your authentication credentials.

IAM role for web crawler data sources

When you use the web crawler, you specify an IAM role that grants Amazon Kendra permission to access the web crawler resources such as your index and secret. The secret stores your credentials for websites or web proxy servers that require basic authentication. The IAM role for the web crawler must have permission to access the secret and to use the AWS Key Management Service (AWS KMS) key to decrypt it. The IAM role also needs access to your index so that it can add and update crawled webpages to the index. For more information, see [IAM access roles for Amazon Kendra](#).

Web proxy

You can use a web proxy to connect to internal websites you want to crawl. Amazon Kendra supports connecting to web proxy servers that are backed by basic authentication or you can connect with no authentication. You provide the host name of the website and the port number. You can also provide web proxy credentials using a secret in [AWS Secrets Manager](#).

You use the [ProxyConfiguration](#) structure to provide the website host name and port number. You can also provide the secret that stores your web proxy credentials.

Stopping Amazon Kendra Web Crawler from indexing your website

Amazon Kendra is an intelligent search service that AWS customers may use to index and search documents of their choice. In order to index documents on the web, customers may use Amazon Kendra Web Crawler, indicating which URL(s) should be indexed and other operational parameters. Amazon Kendra customers are required to obtain authorization before indexing any particular website.

You can stop the Amazon Kendra Web Crawler from indexing your website using the `Disallow` directive, as shown below. You can also control which webpages are indexed and which webpages are not crawled.

Amazon Kendra Web Crawler respects standard robots.txt directives like `Allow` and `Disallow`. Each Amazon Kendra customer using the web crawler has a unique user agent or customer ID. You can identify the user agent or customer ID that you would like to control and configure it in the robots.txt directives.

For example, the below directives stop an Amazon Kendra customer from being able to index a directory of your webpages under `/do-not-crawl/`, but allow indexing a sub-directory `/do-not-crawl/except-this/`:

```
User-agent: amazon-kendra-customer-id-[id] # Amazon customer's user agent/ID
Disallow: /do-not-crawl/ # disallow this directory
Allow: /do-not-crawl/except-this/ # allow this subdirectory

User-agent: * # any robot
Disallow: /not-allowed/ # disallow this directory

User-agent: amazon-kendra-web-crawler-* # all customers of Amazon Kendra Web Crawler
Disallow: /confidential/ # disallow this directory
```

Amazon Kendra Web Crawler also supports the robots `noindex` and `nofollow` directives in meta tags in HTML pages. These directives stop the web crawler from indexing a webpage and stops following any links on the webpage. You put the meta tags in the section of the document to specify the rules of robots rules.

For example, the below webpage includes the directives robots `noindex` and `nofollow`:

```
<html>
<head>
    <meta name="robots" content="noindex, nofollow"/>
    ...
</head>
<body>...</body>
</html>
```

If you have any questions or concerns regarding Amazon Kendra Web Crawler, you can reach out to the [AWS support team](#).

Using an Amazon WorkDocs data source

You can use your Amazon WorkDocs as a data source for Amazon Kendra. For a walk-through of how to use Amazon WorkDocs in the console, see [Getting started with Amazon WorkDocs \(console\)](#).

When you use Amazon Kendra to index documents in your Amazon WorkDocs site, choose the directory ID that corresponds with your Amazon WorkDocs site repository. You can specify regular expression patterns to include or exclude specific documents in your Amazon WorkDocs site.

Amazon WorkDocs connector is available in Oregon, North Virginia, Sydney, Singapore and Ireland regions.

You need to create an index before you create the Amazon WorkDocs data source. For more information, see [Creating an index](#). You provide the ID of the index when you create the data source.

To connect to an Amazon WorkDocs site, you specify connection and other information in the console or by using the [WorkDocsConfiguration](#) data type.

Provide the directory ID of the Amazon WorkDocs site you want to index.

You also need to provide the Amazon Resource Name (ARN) of an IAM role that gives permission to access your Amazon WorkDocs site. You provide the ARN of an IAM role using the [CreateDataSource](#) operation. For more permissions information, see [IAM roles for Amazon WorkDocs data sources](#)

You also can add the following optional information:

- Whether Amazon Kendra should index the contents of comments of your documents. Each comment is indexed as a separate document.
- Whether Amazon Kendra should use the Amazon WorkDocs change log mechanism to determine if a document needs to be updated in the index. Use the change log if you don't want Amazon Kendra to scan all of the site documents. If your change log is large, it may take Amazon Kendra less time to scan the documents in the Amazon WorkDocs site than to process the change log. If you are syncing your Amazon WorkDocs data source with your index for the first time, all documents are scanned.
- Inclusion or exclusion pattern: If you specify an inclusion pattern, any document with a file name that does not match the pattern is not indexed. If you specify an inclusion and exclusion pattern, documents that match the exclusion pattern are not indexed even if they match the inclusion pattern.
- Field mappings that map your Amazon WorkDocs fields to Amazon Kendra index fields. For more information, see [Mapping data source fields](#).

Deleting data sources

You delete a data source when you want to remove the information contained in the data source from your Amazon Kendra index. For example, delete a data source when:

- A data source is incorrectly configured. Delete the data source, wait for the data source to finish deleting, and then recreate it.
- You migrated documents from one data source to another. Delete the original data source and recreate it in the new location.
- You have reached the limit of data sources for an index. Delete one of the existing data sources and add a new one. For more information about the number of data sources that you can create, see [Quotas \(p. 260\)](#).

To delete a data source, use the console, the AWS Command Line Interface (AWS CLI), the `DeleteDataSource` operation, or a AWS CloudFormation script. Deleting a data source removes all of the information about the data source from the index. If you only want to stop synching the data source, change the synchronization schedule for the data source to "run on demand".

To delete a data source (console)

1. Sign in to the AWS Management Console and open the Amazon Kendra console at <https://console.aws.amazon.com/kendra/>.
2. In the navigation pane, choose **Indexes**, and then choose the index that contains the data source to delete.
3. In the navigation pane, choose **Data sources**.
4. Choose the data source to remove.
5. Choose **Delete** to delete the data source.

To delete a data source (CLI)

- In the AWS Command Line Interface, use the following command. The command is formatted for Linux and macOS. If you are using Windows, replace the Unix line continuation character (\) with a caret (^).

```
aws kendra delete-data-source \
  --id data-source-id \
  --index-id index-id
```

When you delete a data source, Amazon Kendra removes all of the stored information about the data source. Amazon Kendra removes all of the document data stored in the index, and all run histories and metrics associated with the data source. Deleting a data source does not remove the original documents from your storage.

Deleting a data source is an asynchronous operation. When you start deleting a data source, the data source status changes to `DELETING`. It remains in the `DELETING` state until the information related to the data source is removed. After the data source is deleted, it no longer appears in the results of a call to the [ListDataSources \(p. 357\)](#) operation. If you call the [DescribeDataSource \(p. 322\)](#) operation with the deleted data source's identifier, you receive a `ResourceNotFoundException`.

Documents in the data source may be included in the document count returned by the `DescribeIndex` operation while Amazon Kendra deletes a data source. Documents from the data source may appear in search results while Amazon Kendra deletes the data source.

Amazon Kendra releases the resources for a data source as soon as you call the `DeleteDataSource` operation or choose to delete the data source in the console. If you are deleting the data source to reduce the number of data sources below your limit, you can create a new data source right away.

If you are deleting a data source and then creating another data source to the document data, wait for the first data source to be deleted before you sync the new data source.

You can delete a data source that is in the process of syncing with Amazon Kendra. The sync is stopped and the data source is removed. If you attempt to start a sync when the data source is being deleted, you get a `ConflictException` exception.

You can't delete a data source if the associated index is in the `DELETING` state. Deleting an index deletes all of the data sources for the index. You can start deleting an index while a data source for that index is in the `DELETING` state.

If you have two data sources pointing to the same documents, such as two data sources pointing to the same S3 bucket, documents in the index might be inconsistent when one of the data sources is deleted. When two data sources reference the same documents, only one copy of the document data is stored in the index. Removing one data source removes the index data for the documents. The other data source is not aware that the documents have been removed, so Amazon Kendra won't correctly re-index the documents the next time it syncs. When you have two data sources pointing to the same document location, you should delete both data sources and then recreate one.

Creating custom document attributes

When the source of your data is an S3 bucket or an Amazon S3 data source, you can apply custom attributes to your documents using metadata files. For example, you could create a custom attribute called "Department" with values "HR", "Sales", and "Manufacturing". You can apply these attributes to your documents so that you can limit the response to documents in the "HR" department, for example.

You can create up to 500 custom attributes.

For other data sources, you map fields in the external data source to the corresponding custom attributes in Amazon Kendra. For more information, see [Mapping data source fields \(p. 123\)](#).

Before you can use a custom attribute, you must first create a field in the index. Use the console or the [UpdateIndex \(p. 411\)](#) operation to create the index fields. The supported field types are date, long, string, and string list.

With the `UpdateIndex` operation, you add custom fields using the `DocumentMetadataConfigurationUpdates` parameter

The following JSON example is a `DocumentMetadataConfigurationUpdates` structure that adds a field called "Department" to the index.

```
"DocumentMetadataConfigurationUpdates": [  
    {  
        "Name": "Department",  
        "Type": "STRING_VALUE"  
    }  
]
```

Amazon Kendra has 15 reserved attributes that you can use. The attributes are:

- `_authors` (String list) – A list of one or more authors responsible for the content of the document.
- `_category` (String) – A category that places a document in a specific group.
- `_created_at` (ISO 8601 encoded string) – The date and time that the document was created.

It is important for the time zone to be included in the ISO 8601 date-time format. For example, 20120325T123010+01:00 is the ISO 8601 date-time format for March 25th 2012 at 12:30PM (plus 10 seconds) in Central European Time.

- `_data_source_id` (String) – The identifier of the data source that contains the document.
- `_document_body` (String) – The content of the document.

- `_document_id` (String) – A unique identifier for the document.
- `_document_title` (String) – The title of the document.
- `_excerpt_page_number` (Long) – The page number in a PDF file where the document excerpt appears. If your index was created before September 8, 2020, you must re-index your documents before you can use this attribute.
- `_faq_id` (String) – If this is an FAQ question and answer, a unique identifier for them.
- `_file_type` (String) – The file type of the document, such as pdf or doc.
- `_last_updated_at` (ISO 8601 encoded string) – The date and time that the document was last updated.

It is important for the time zone to be included in the ISO 8601 date-time format. For example, 20120325T123010+01:00 is the ISO 8601 date-time format for March 25th 2012 at 12:30PM (plus 10 seconds) in Central European Time.

- `_source_uri` (String) – The URI where the document is available. For example, the URI of the document on a company website.
- `_version` (String) – An identifier for the specific version of a document.
- `_view_count` (Long) – The number of times that the document has been viewed.
- `_language_code` (String) – The code for a language that applies to the document. This defaults to English if you do not specify a language. For more information on supported languages, including their codes, see [Adding documents in languages other than English](#).

After you have created a custom attribute, you can use the attribute when you call the `Query` operation. You can use it for faceted search, use it to filter the response, and choose whether or not the attribute should be returned in the response. For more information, see [Filtering queries \(p. 134\)](#).

Adding custom attributes with the BatchPutDocument operation

When you use the [BatchPutDocument \(p. 277\)](#) operation to add a document to your index, you specify custom attributes as part of the `Attributes` structure. You can add multiple attributes when you call the operation. You can create up to 500 custom attributes. The following example is a `Attributes` structure that adds "Department" and "`_category`" attributes to a document.

```
"Attributes":  
{  
    "Department": "HR",  
    "_category": "Vacation policy"  
}
```

Adding custom attributes to an Amazon S3 data source

When you use an Amazon S3 bucket as a data source for your index, you add metadata to the documents with companion metadata files. You place the metadata JSON files in a directory structure that is parallel to your documents. For more information, see [S3 document metadata \(p. 88\)](#).

You specify custom attributes in the `Attributes` JSON structure. You can create up to 500 custom attributes. For example, the following example is an `Attributes` structure that defines three custom attributes and one reserved attribute.

```
"Attributes": {
```

```
        "brand": "Amazon Basics",
        "price": 1595,
        "_category": "sports",
        "subcategories": ["outdoors", "electronics"]
    }
```

Mapping data source fields

When the source of your data is a data source other than Amazon Simple Storage Service (Amazon S3), you can map data source fields to fields in your index. For example, if you have a field that contains department information for a document, you can map it to an index field called "Department" so that you can use the field in queries.

You can create field mappings for the following data sources:

- Confluence
- Database
- Google Workspace Drives
- Microsoft OneDrive
- Microsoft SharePoint
- Salesforce
- ServiceNow
- Amazon Kendra Web Crawler
- Amazon WorkDocs

If you store your documents in an S3 bucket, or if you use an Amazon S3 data source, you can provide custom attributes directly using metadata files. For more information, see [Creating custom document attributes \(p. 121\)](#).

Mapping your data source fields to an index field is a three-step process:

1. Create an index. For more information, see [Creating an index \(p. 64\)](#).
2. Update the index to add custom fields.
3. Create a data source that maps data source fields to the index fields.

To update the index to add custom fields, use the console or the [UpdateIndex \(p. 411\)](#) operation. You can add a total of 500 custom fields to your index.

When you are using the console, you can choose to map a data source field to one of the seven reserved field names, or you can choose to create a new index field that maps to the field. For database data sources, if the name of the database column matches the name of a reserved field, the field and column are automatically mapped.

With the API, you add custom and reserved fields using the `DocumentMetadataConfigurationUpdates` parameter.

The following JSON example is a `DocumentMetadataConfigurationUpdates` structure that adds a field called "Department" to the index.

```
"DocumentMetadataConfigurationUpdates": [
    {
```

```
        "Name": "Department",
        "Type": "STRING_VALUE"
    }
]
```

When you create the field, you have the option of setting how the field should be used in searches. You can choose from the following:

- **Displayable** – Determines whether the field is returned in the query response. The default is `true`.
- **Facetable** – Indicates that the field can be used to create facets. The default is `false`.
- **Searchable** – Determines whether the field is used in the search. The default is `true` for string fields and `false` for number and date fields.
- **Sortable** – Indicates that the field can be used to sort the response from a query. Can only be set for date, number, and string fields. Can't be set for string list fields.

The following JSON example is a `DocumentMetadataConfigurationUpdates` structure that adds a field called "Department" to the index and marks it as facetable.

```
"DocumentMetadataConfigurationUpdates": [
    {
        "Name": "Department",
        "Type": "STRING_VALUE",
        "Search": {
            "Facetable": true
        }
    }
]
```

Amazon Kendra has 15 reserved fields that you can map to data source fields. You must provide values for these fields. The fields are:

- `_authors` (String list) – A list of one or more authors responsible for the content of the document.
- `_category` (String) – A category that places a document in a specific group.
- `_created_at` (ISO 8601 encoded string) – The date and time in ISO 8601 format that the document was created. For example, `20120325T123010+01:00` is the ISO 8601 date-time format for March 25th 2012 at 12:30PM (plus 10 seconds) in Central European Time.
- `_data_source_id` (String) – The identifier of the data source that contains the document.
- `_document_body` (String) – The content of the document.
- `_document_id` (String) – A unique identifier for the document.
- `_document_title` (String) – The title of the document.
- `_excerpt_page_number` (Long) – The page number in a PDF file where the document excerpt appears. If your index was created before September 8, 2020, you must re-index your documents before you can use this attribute.
- `_faq_id` (String) – If this is an FAQ question and answer, a unique identifier for them.
- `_file_type` (String) – The file type of the document, such as `pdf` or `doc`.
- `_last_updated_at` (ISO 8601 encoded string) – The date and time in ISO 8601 format that the document was last updated. For example, `20120325T123010+01:00` is the ISO 8601 date-time format for March 25th 2012 at 12:30PM (plus 10 seconds) in Central European Time.
- `_source_uri` (String) – The URI where the document is available. For example, the URI of the document on a company website.
- `_version` (String) – An identifier for the specific version of a document.
- `_view_count` (Long) – The number of times that the document has been viewed.

- `_language_code (String)` – The code for a language that applies to the document. This defaults to English if you do not specify a language. For more information on supported languages, including their codes, see [Adding documents in languages other than English](#).

After you have created the index fields, you can map the data source fields to the index fields. If you are using the console, you can create index fields and map data source fields using the **Custom field mappings** editor. If you are using the API, you can add field mappings using the [CreateDataSource](#) (p. 283) or [UpdateDataSource](#) (p. 403) operations.

You can prevent custom index fields from being searchable. In the console, you simply uncheck **Searchable** for a field in the index field settings. If you use the API, you simply set `Searchable` to `FALSE` for a field using the `Search` structure.

Configuring Amazon Kendra to use a VPC

Amazon Kendra connects to your Amazon virtual private cloud (VPC) to index information stored in databases running in your private cloud. When you create the database data source, you provide security group and subnet identifiers for the subnet that contains your database. Amazon Kendra uses this information to create an elastic network interface that it uses to securely communicate with your database.

If your database isn't running on an Amazon VPC, you can connect your database to your Amazon VPC using a virtual private network (VPN). You get a default VPC when you create your Amazon account. For information about setting up a VPN, see the [AWS Virtual Private Network Documentation](#).

To use a VPC, you must tell Amazon Kendra the identifier of the subnet that the database belongs to and the identifiers of any security groups that Amazon Kendra must use to access the subnet. For example, if you are using the default port for a MySQL database, the security groups must enable Amazon Kendra to access port 3306 on the host that runs the database.

Note

Only use private subnets in the VPC configuration of your data source. If your RDS instance is in a public subnet in your VPC, then you can't use that subnet directly to sync your data source. Instead, create a private subnet that has outbound access to a NAT gateway in the public subnet. When you configure the VPC configuration for your database data source, specify that private subnet.

The identifiers for subnets and security groups are configured in the Amazon VPC control panel. To see the identifiers, open the Amazon VPC console as follows:

To view subnet identifiers

1. Sign in to the AWS Management Console and open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. From the navigation pane, choose **Subnets**.
3. From the subnet list, choose the subnet that contains your database server.
4. From the description tab, make a note of the identifier in the **Subnet ID** field.

To view security group identifiers

1. Sign in to the AWS Management Console and open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. From the navigation pane, choose **Security Groups**.

3. From the security group list, choose the group that you want the identifier for.
4. From the description tab, make a note of the identifier in the **Group ID** field.

If Amazon Kendra must route the connection between two or more subnets, you can provide more than one subnet. For example, if the subnet that contains your database server is out of IP addresses, Amazon Kendra can connect to a subnet with free IP addresses and route the connection to the first subnet. If you list multiple subnets, the subnets must be able to communicate with each other. Each subnet should be associated with a route table that provides outbound internet access using a network address translator (NAT) device.

You can also provide multiple security groups. The combined effect of the security groups should allow Amazon Kendra to access the database server that you have specified in the connection configuration for the data source.

Connecting to a database in a VPC

The following example shows how to connect a database data source to a MySQL database running in a VPC. The example assumes that you are starting with your default VPC and that you need to create a MySQL database. If you already have a VPC, make sure that it is configured as shown. If you have a MySQL database, you can use that instead of creating a new one.

Topics

- [Step 1: Configure a VPC \(p. 126\)](#)
- [Step 2: Configure security \(p. 127\)](#)
- [Step 3: Create a database \(p. 127\)](#)
- [Step 4: Create a database data source \(p. 127\)](#)

Step 1: Configure a VPC

Configure your VPC so that you have a private subnet and a security group that enables Amazon Kendra to access a MySQL database running in the subnet.

To configure a VPC

1. Sign in to the AWS Management Console and open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. From the navigation pane, choose **Route tables**, then choose **Create route table**.
3. For the **Name tag** field, enter **Private subnet route table**. In the **VPC** field, choose your VPC, and then choose **Create**. Choose **Close** to return to the list of route tables.
4. From the navigation pane, choose **NAT Gateways** then choose **Create NAT Gateway**.
5. In the **Subnet** field, choose the subnet that should be the public subnet and note the subnet ID.
6. If you don't have an Elastic IP address, choose **Create New EIP**, choose **Create a NAT Gateway**, and then choose **Close**.
7. From the navigation pane, choose **Route Tables**.
8. From the route table list, choose the **Private subnet route table** that you created in step 3. From **Actions**, choose **Edit Routes**.
9. Choose **Add route**. Add the destination 0.0.0.0/0 to allow all outgoing traffic to the internet. For **Target**, choose **NAT Gateway** and then choose the gateway that you created in step 4. Choose **Save routes**, and then choose **Close**.
10. From **Actions**, choose **Edit subnet associations**.
11. Choose the subnets that you want to be private. Do not choose the subnet with the NAT gateway that you noted above.

Step 2: Configure security

Next, configure security groups for your database.

To create security groups

1. Sign in to the AWS Management Console and open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. From the description of your VPC, note the IPv4 CIDR.
3. From the navigation pane, choose **Security Groups** and then choose **Create security group**.
4. In **Security group name** enter **DataSourceInboundSecurityGroup**. Provide a description, then choose your VPC from the list. Choose **Create** and then choose **Close**.
5. Choose the **Inbound** tab.
6. Choose **Edit rules**, and then choose **Add Rule**
7. For a database, enter the port number for the **Port Range**. For example, for MySQL it is **3306**, and for HTTPS it is **443**. For the **Source**, type the Classless Inter-Domain Routing (CIDR) of your VPC. Choose **Save rules** and then choose **Close**.

The security group allows anyone within the VPC to connect to the database, and it allows outbound connections to the internet.

Step 3: Create a database

Create a database to hold your documents. If you already have a database, you can use that instead.

For an example of creating a MySQL database, see [Getting Starting with a MySQL database data source \(Console\)](#).

Step 4: Create a database data source

After you have configured your VPC and created your database, you can create a data source for the database.

You need to configure your VPC, the private subnets that you created in your VPC, and the security group that you created in your VPC for your database.

For an example of creating a data source for a MySQL database, see [Getting Starting with a MySQL database data source \(Console\)](#).

Deleting an index

Delete an index from Amazon Kendra when you are no longer using the index. For example, delete an index when:

- You are no longer using the index and want to reduce charges to your AWS account. An Amazon Kendra index accrues charges while it is running whether or not you make queries on the index.
- You want to reconfigure the index for a different edition of Amazon Kendra. Delete the existing index and then create a new one with the different edition.
- You have reached the maximum number of indexes in your account and don't want to exceed your quota. Delete an existing index and add a new one. For information about the maximum number of indexes that you can create, see [Quotas \(p. 260\)](#).

To delete an index, use the console, the AWS Command Line Interface, an AWS CloudFormation script, or the `DeleteIndex` operation. Deleting an index removes the index and all associated data sources and document data. Deleting an index doesn't remove the original documents from your storage.

To delete an index (console)

1. Sign in to the AWS Management Console and open the Amazon Kendra console at <https://console.aws.amazon.com/kendra/>.
2. In the navigation pane, choose **Indexes**, and then choose the index to delete.
3. Choose **Delete** to delete the selected index.

To delete an index (CLI)

- In the AWS CLI, use the following command. The command is formatted for Linux and macOS. If you are using Windows, replace the Unix line continuation character (\) with a caret (^).

```
aws kendra delete-index \
  --id index-id
```

Deleting an index is an asynchronous operation. When you start deleting an index, the index status changes to `DELETING`. It remains in the `DELETING` state until all of the information related to the index is removed. Once the index is deleted, it no longer appears in the results of a call to the [ListIndices \(p. 370\)](#) operation. If you call the [DescribeIndex \(p. 335\)](#) operation with the deleted index's identifier, you receive a `ResourceNotFoundException`.

Searching indexes

To search an Amazon Kendra index, you use the [Query \(p. 384\)](#) operation. The `Query` operation returns information about the indexed documents that you use in your application. This section shows you how to make a query, perform filters, and interpret the response that you get from the `Query` operation. This section also shows how to submit feedback about the quality of a search result.

Topics

- [Querying an index \(p. 129\)](#)
- [Filtering queries \(p. 134\)](#)
- [Filtering on user context \(p. 136\)](#)
- [Query responses \(p. 143\)](#)
- [Query suggestions \(p. 145\)](#)
- [Tuning responses \(p. 145\)](#)
- [Sorting responses \(p. 146\)](#)
- [Response types \(p. 147\)](#)

Querying an index

When you search your index, Amazon Kendra uses all of the information that you provided about your documents to determine the documents most relevant to the search terms entered. Some of the items that Amazon Kendra considers are:

- The text of the document.
- The title of the document.
- Custom text fields that you have marked searchable.
- The date field that you have indicated should be used to determine the "freshness" of a document.

When a set of relevant documents has been selected from the index, Amazon Kendra filters the response based on any attribute filters that you have requested for the search. For example, if you have a custom attribute called "department," you can filter the response to only return documents from a department called "legal." For more information, see [Creating custom document attributes \(p. 121\)](#).

After finding the relevant documents and then filtering based on the attributes that you set, Amazon Kendra returns the results. The results are sorted by the relevance that Amazon Kendra determined for each doc. The results are paginated so that you can show a page at a time to your user.

The following Python example shows how to search an index by using the [the section called "Query" \(p. 384\)](#) operation. The example determines the type of the search result (answer, document, question/answer) and displays the answer text.

For information about the query responses, see [Query responses \(p. 143\)](#).

Note

You can use this code to filter document attributes. The topic [Filtering queries \(p. 134\)](#) contains examples that you can use to replace the following code.

```
response=kendra.query(  
    QueryText = query,  
    IndexId = index)
```

Prerequisites

To run this example, you must:

- Set up permissions. For more information, see [IAM access roles for Amazon Kendra \(p. 12\)](#).
- Set up the AWS CLI. For more information, see [Setting up the AWS CLI \(p. 10\)](#).
- Create a data source and index. For more information, see [Getting started with an S3 bucket \(console\) \(p. 40\)](#).

Searching an index (console)

You can use the Amazon Kendra search console to test your index. You can make queries and see the results.

To search an index with the console

1. Sign in to the AWS Management Console and open the Amazon Kendra console at <http://console.aws.amazon.com/kendra/>.
2. On the navigation pane, choose **Indexes**.
3. Choose your index.
4. On the navigation page, choose **Search console**.
5. Enter a query in the text box and then press enter or choose the magnifying glass icon.
6. Amazon Kendra returns the results of the search.

Searching an index (SDK)

To search an index with Python or Java

- The following example searches an index. Change the value of `query` to your search query and `index_id` or `indexId` to the index identifier of the index that you want to search.

Python

```
import boto3
import pprint

kendra = boto3.client('kendra')

query='${searchString}'
index_id='${indexID}'

response=kendra.query(
    QueryText = query,
    IndexId = index_id)

print ('\nSearch results for query: ' + query + '\n')

for query_result in response['ResultItems']:

    print('-----')
    print('Type: ' + str(query_result['Type']))

    if query_result['Type']=='ANSWER' or query_result['Type'] == 'QUESTION_ANSWER':
        answer_text = query_result['DocumentExcerpt']['Text']
        print(answer_text)
```

```
if query_result['Type']=='DOCUMENT':
    if 'DocumentTitle' in query_result:
        document_title = query_result['DocumentTitle']['Text']
        print('Title: ' + document_title)
        document_excerpt = query_result['DocumentExcerpt']['Text']
        print(document_excerpt)

print ('-----\n\n')
```

Java

```
package com.amazonaws.kendra;

import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.QueryRequest;
import software.amazon.awssdk.services.kendra.model.QueryResponse;
import software.amazon.awssdk.services.kendra.model.QueryResultItem;

public class SearchIndexExample {
    public static void main(String[] args) {
        KendraClient kendra = KendraClient.builder().build();

        String query = "some queries";
        String indexId = "anIndexId";

        QueryRequest queryRequest = QueryRequest
            .builder()
            .queryText(query)
            .indexId(indexId)
            .build();

        QueryResponse queryResponse = kendra.query(queryRequest);

        System.out.println(String.format("\nSearch results for query: %s", query));
        for(QueryResultItem item: queryResponse.resultItems()) {
            System.out.println("-----");
            System.out.println(String.format("Type: %s", item.type()));

            switch(item.type()) {
                case QUESTION_ANSWER:
                case ANSWER:
                    String answerText = item.documentExcerpt().text();
                    System.out.println(answerText);
                    break;
                case DOCUMENT:
                    String documentTitle = item.documentElement().text();
                    System.out.println(String.format("Title: %s", documentTitle));
                    String documentExcerpt = item.documentElement().text();
                    System.out.println(String.format("Excerpt: %s",
documentExcerpt));
                    break;
                default:
                    System.out.println(String.format("Unknown query result type:
%s", item.type()));
            }
            System.out.println("-----\n");
        }
    }
}
```

Searching in languages

You can search for documents in a supported language. This is useful for helping your users overcome language barriers by allowing them to search and find documents in their native language. You pass the language code in the [AttributeFilter](#) to return filtered documents in your chosen language. You can type the query in a supported language.

If you do not specify a language, Amazon Kendra queries documents in English by default. For more information on supported languages, including their codes, see [Adding documents in languages other than English](#).

To search for documents in a supported language in the console, go to **Search console** and enter a query in the text box in a supported language. You choose the language you want to return documents by selecting the dropdown **Language**. You press enter or select the magnifying glass icon to return documents in your chosen languages.

The following examples show how to search for documents in Spanish and Japanese.

To search an index in Spanish in the console

1. Sign in to the AWS Management Console and open the Amazon Kendra console at <http://console.aws.amazon.com/kendra/>.
2. On the navigation pane, choose **Indexes**.
3. Choose your index.
4. On the navigation page, choose **Search console**.
5. Select the **Languages** dropdown and choose Spanish.
6. Enter a query into the text box and then press enter or select the magnifying glass icon.
7. Amazon Kendra returns the results of the search in Spanish.

To search an index in Spanish with Python or Java

- The following example searches an index in Spanish. Change the value `searchString` to your search query and the value `indexID` to the identifier of the index that you want to search. The language code for Spanish is `es`. You can replace these with your own language code.

CLI

```
{  
    "EqualsTo":{  
        "Key": "_language_code",  
        "Value": {  
            "StringValue": "es"  
        }  
    }  
}
```

Python

```
import boto3  
import pprint  
  
kendra = boto3.client('kendra')  
  
query = '${searchString}'  
index_id = '${indexID}'  
  
response=kendra.query(
```

```

        QueryText = query,
        IndexId = index_id,
        AttributeFilter = {
            "EqualsTo": {
                "Key": "_language_code",
                "Value": {
                    "StringValue": "es"
                }
            }
        })
    }

print ('\nSearch results|Resultados de la búsqueda: ' + query + '\n')

for query_result in response['ResultItems']:

    print('-----')
    print('Type: ' + str(query_result['Type']))

    if query_result['Type'] == 'ANSWER' or query_result['Type'] ==
    'QUESTION_ANSWER':
        answer_text = query_result['DocumentExcerpt']['Text']
        print(answer_text)

    if query_result['Type'] == 'DOCUMENT':
        if 'DocumentTitle' in query_result:
            document_title = query_result['DocumentTitle']['Text']
            print('Title: ' + document_title)
            document_text = query_result['DocumentExcerpt']['Text']
            print(document_text)

    print ('-----\n\n')

```

Java

```

package com.amazonaws.kendra;

import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.QueryRequest;
import software.amazon.awssdk.services.kendra.model.QueryResponse;
import software.amazon.awssdk.services.kendra.model.QueryResultItem;

public class SearchIndexExample {
    public static void main(String[] args) {
        KendraClient kendra = KendraClient.builder().build();

        String query = "searchString";
        String indexId = "indexID";

        QueryRequest queryRequest = QueryRequest.builder()
            .queryText(query)
            .indexId(indexId)
            .attributeFilter(
                AttributeFilter.builder()
                    .withEqualsTo(
                        DocumentAttribute.builder()
                            .withKey("_language_code")
                            .WithValue("es")
                            .build())
                    .build())
            .build();
        build();

        QueryResponse queryResponse = kendra.query(queryRequest);

```

```

System.out.println(String.format("\nSearch results|"
                               "Resultados de la búsqueda: %s", query));
for(QueryResultItem item: queryResponse.resultItems()) {
    System.out.println("-----");
    System.out.println(String.format("Type: %s", item.type()));

    switch(item.type()) {
        case QUESTION_ANSWER:
        case ANSWER:
            String answerText = item.documentExcerpt().text();
            System.out.println(answerText);
            break;
        case DOCUMENT:
            String documentTitle = item.documentElement().text();
            System.out.println(String.format("Title: %s", documentTitle));
            String documentExcerpt = item.documentElement().text();
            System.out.println(String.format("Excerpt: %s",
                documentExcerpt));
            break;
        default:
            System.out.println(String.format("Unknown query result type:
%s", item.type()));
    }
    System.out.println("-----\n");
}
}

```

Filtering queries

You can improve the response from the [Query \(p. 384\)](#) operation by using filters. Filters restrict the documents in the response to ones that directly apply to the query. You can create faceted search suggestions, use Boolean logic to filter out documents that don't match specific criteria, or filter out specific document attributes from the response.

Facets

Facets are scoped views of a set of search results. For example, if your index provides search results for cities across the world, facets can offer search results filtered by a specific city. Or, create facets to display results by a specific author. The following example shows how to get facet information for the `_category` custom attribute.

```

response=kendra.query(
    QueryText = query,
    IndexId = index,
    Facets = [
        {
            "DocumentAttributeKey" : "_category"
        }
    ]
)

```

Facet information is returned in the `FacetResults` response array. You use the contents to display faceted search suggestions in your application. For example, if the document attribute `Category` contains the city that a search could apply to, use that information to display a list of city searches.

Users can choose a city to get faceted search results scoped to the city. To make the faceted search, call [Query \(p. 384\)](#) and use the chosen document attribute to filter the results. For an example, see [Using document attributes to filter search results \(p. 135\)](#).

The following sample JSON response shows facets scoped to the `_category` document attribute. The response includes the count of documents that include the value of the document attribute.

```
{  
    "FacetResults": [  
        {  
            "DocumentAttributeKey": "_category",  
            "DocumentAttributeValueCountPairs": [  
                {  
                    "Count": 3,  
                    "DocumentAttributeValue": {  
                        "StringValue": "Dubai"  
                    }  
                },  
                {  
                    "Count": 3,  
                    "DocumentAttributeValue": {  
                        "StringValue": "Seattle"  
                    }  
                },  
                {  
                    "Count": 1,  
                    "DocumentAttributeValue": {  
                        "StringValue": "Paris"  
                    }  
                }  
            ]  
        }  
    ]  
}
```

When you use a string list field to create facets, the facet results returned are based on the contents of the string list. For example, if you have a string list field that contains two items, one with the value "corgi", "dachshund," and one with the value "husky," you get a `FacetsResults` structure with three facets.

For more information, see [Query responses \(p. 143\)](#).

Using document attributes to filter search results

By default, `Query` returns all search results. To filter responses, you can perform logical operations on the document attributes. For example, if you only want documents for a specific city, you can filter on the `City` and `State` custom document attributes. You use the [AttributeFilter \(p. 430\)](#) input parameter to create a Boolean operation on filters that you supply.

Most attributes can be used to filter responses for all [response types](#). However, the `_excerpt_page_number` attribute is only applicable to `ANSWER` response types when filtering responses.

The following example shows how to perform a logical AND operation by filtering on a specific city, *Seattle*, and state, *Washington*.

```
response=kendra.query(  
    QueryText = query,  
    IndexId = index,  
    AttributeFilter = {'AndAllFilters':
```

```
[  
    {"EqualsTo": {"Key": "City", "Value": {"StringValue": "Seattle"}},  
    {"EqualsTo": {"Key": "State", "Value": {"StringValue": "Washington"}}}  
]  
)  
)
```

The following example shows how to perform a logical OR operation for when any of the `Fileformat`, `Author`, or `SourceURI` keys match the specified values.

```
response=kendra.query(  
    QueryText = query,  
    IndexId = index,  
    AttributeFilter = {'OrAllFilters':  
        [  
            {"EqualsTo": {"Key": "Fileformat", "Value": {"StringValue":  
                "AUTO_DETECT"}},  
            {"EqualsTo": {"Key": "Author", "Value": {"StringValue": "Ana Carolina"}},  
            {"EqualsTo": {"Key": "SourceURI", "Value": {"StringValue": "https://  
aws.amazonaws.com/234234242342"}}}  
        ]  
    }  
)
```

For `StringList` fields, use the `ContainsAny` or `ContainsAll` attribute filters to return documents with the specified string. The following example shows how to return all documents that have the values "Seattle" or "Portland" in their `Locations` custom attribute.

```
response=kendra.query(  
    QueryText = query,  
    IndexId = index,  
    AttributeFilter = {  
        "ContainsAny": { "Key": "Locations", "Value": { "StringListValue":  
            [ "Seattle", "Portland" ] } }  
    }  
)
```

Filtering a document's attributes

By default, all document attributes for a document are returned in the `DocumentAttributes` response field. You can choose to include only specific document attributes by using the `IncludeDocumentAttributes` input parameter. In the following example, only the `SourceURI` and `Author` document attributes are included in the response for a document.

```
response=kendra.query(  
    QueryText = query,  
    IndexId = index,  
    IncludeDocumentAttributes = ["SourceURI", "Author"]  
)
```

Filtering on user context

Amazon Kendra enables you to filter a query response to remove documents from a user's search results based on their user name and group membership. You can filter search results by user token, user ID, or user attribute. Amazon Kendra can map users to their groups.

Filtering by user token

When a document is indexed into Amazon Kendra, a corresponding access control list (ACL) is ingested. The ACL specifies which user names and group names are allowed or denied access to the document. Documents without an ACL are public documents.

When you query the documents, you want to ensure the correct ACL is applied. For example, you can use an Open ID token. It uses a JSON format. When you issue a query, Amazon Kendra extracts and validates the token. Amazon Kendra pulls the user and group information and runs the query. All of the documents the user has access to including public documents are returned.

You provide the user token in the [UserContext](#) object and pass this in the [Query](#).

The following shows how to include a user token.

```
response = kendra.query(  
    QueryText = query,  
    IndexId = index,  
    UserToken = {  
        Token = "token"  
    })
```

You can map users to groups. When you use user-context filtering, it is not required to include all of the groups that a user belongs to when you issue the query. The [PutPrincipalMapping](#) operation allows you to map users to their groups. If you do not want to use the [PutPrincipalMapping](#) operation, you must provide the user name and all the groups the user belongs to when you issue a query. You can also fetch access levels of groups and users in your AWS Single Sign-On identity source by using the [UserGroupResolutionConfiguration](#) structure.

Filtering by user ID

When you query the index, you use the user and group name IDs to specify the user and group information.

You can also filter data sources by the user's group access.

Specifying a data source is useful if a group is tied to multiple data sources, but you only want the group to access documents of a certain data source. For example, the groups "Research", "Engineering", and "Sales and Marketing" are all tied to the company's documents stored in the data sources Confluence and Salesforce. However, "Sales and Marketing" team only needs access to customer-related documents stored in Salesforce. This means when a user who works in sales and marketing searches for customer-related documents, the user can see documents from Salesforce in their search results. Users who do not work in sales and marketing do not see Salesforce documents in their search results.

Warning

You are required to provide one of the following:

- User and groups information, and (optional) data sources information.
- Only the user information if you map your users to groups and data sources using the [PutPrincipalMapping](#) operation. You can also fetch access levels of groups and users in your AWS Single Sign-On identity source by using the [UserGroupResolutionConfiguration](#) structure.

If this information is not included in the query, Amazon Kendra returns all documents. If you provide this information, only documents with matching user IDs, groups, and data sources are returned.

You provide the user, groups and data sources information in the [UserContext](#) object and pass this in the [Query](#). The user ID, and the list of groups and data sources should match the name you specify in the [Principal](#) object to identify the user, groups and data sources. The [Principal](#) object allows you to add a user, group, or data source to either an allow list or a deny list for accessing a document.

The following shows how to include user ID, groups, and data sources.

```
response = kendra.query(  
    QueryText = query,  
    IndexId = index,  
    UserId = {  
        UserId = "user1"  
    },  
    Groups = {  
        Groups = ["Sales and Marketing"]  
    },  
    DataSourceGroups = {  
        DataSourceGroups = [{"DataSourceId" : "SalesforceCustomerDocsGroup", "GroupId": "Sales and Marketing"}]  
    })
```

Filtering by user attribute

When you query the index, you use the built-in attributes `_user_id` and `_group_ids` to specify the user and group information. You can set up to 100 group identifiers. You can add a user or a group to an allow list or a deny list for accessing a document using the [Principal](#) object. If a user or group is added to the deny list, the document is filtered out of any query that contains the user or group.

Warning

You are required to provide one of the following:

- User and group information.
- Only the user information if you map your users to groups and data sources using the [PutPrincipalMapping](#) operation. You can also fetch access levels of groups and users in your AWS Single Sign-On identity source by using the [UserGroupResolutionConfiguration](#) structure.

If this information is not included in the query, Amazon Kendra returns all documents. If you provide this information, only documents with matching user IDs and groups are returned.

You provide the user and groups attributes in the [AttributeFilter](#) object and pass this in the [Query](#).

The following example shows a request that filters the query response based on the user ID and the groups "HR" and "IT", which the user belongs to. The query will return any document that has the user or the "HR" or "IT" groups in the allow list. If the user or either group is in the deny list for a document, the document is not returned.

```
response = kendra.query(  
    QueryText = query,  
    IndexId = index,  
    AttributeFilter = {  
        "OrAllFilters": [  
            {  
                "EqualsTo": {  
                    "Key": "_user_id",  
                    "Value": {  
                        "StringValue": "user1"  
                    }  
                }  
            }  
        ]  
    }  
}
```

```
        },
        {
            "EqualsTo": {
                "Key": "_group_ids",
                "Value": {
                    "StringListValue": ["HR", "IT"]
                }
            }
        ]
    }
}
```

You can also specify which data source a group can access in the `Principal` object.

Note

User context filtering isn't an authentication or authorization control for your content. It doesn't do user authentication on the user and groups sent to the `Query` operation. It is up to your application to ensure that the user and group information sent to `Query` operation is authenticated and authorized.

There is an implementation of user context filtering for each data source. The following section describes each implementation.

Topics

- [User context filtering for documents added directly to an index \(p. 139\)](#)
- [User context filtering for frequently asked questions \(p. 139\)](#)
- [User context filtering for database data sources \(p. 140\)](#)
- [User context filtering for Confluence data sources \(p. 140\)](#)
- [User context filtering for Google Drive data sources \(p. 140\)](#)
- [User context filtering for Microsoft OneDrive data sources \(p. 141\)](#)
- [User context filtering for Amazon S3 data sources \(p. 142\)](#)
- [User context filtering for Salesforce data sources \(p. 142\)](#)
- [User context filtering for ServiceNow data sources \(p. 142\)](#)
- [User context filtering for Microsoft SharePoint data sources \(p. 143\)](#)
- [User context filtering for Amazon WorkDocs data sources \(p. 143\)](#)

User context filtering for documents added directly to an index

To specify user context filters for a document that you add directly to an index, you provide the filter in the `AccessControlList` field of the document. You provide three pieces of information:

- The access that the entity should have. You can say `ALLOW` or `DENY`.
- The type of entity. You can say `USER` or `GROUP`.
- The name of the entity.

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for frequently asked questions

You can add user context information to frequently asked questions when you use custom .csv or JSON file formats. For more information, see [Adding questions and answers directly to an index \(p. 80\)](#).

User context filtering for database data sources

For a database data source, information for user context filtering comes from a column in the source table. You specify the column name in the console or when you create the data source using the `AclConfiguration` field.

A database data source has the following limitations:

- You can only specify an allow list for a database data source. You can't specify a deny list.
- You can only specify groups. You can't specify individual users for the allow list.
- The database column should be string containing a semicolon delimited list of groups.

User context filtering for Confluence data sources

When you use a Confluence data source, Amazon Kendra gets user and group information from the Confluence instance.

You configure user and group access to spaces using the space permissions page. For pages and blogs, you use the restrictions page. For more information about space permissions, see [Space Permissions Overview](#) on the Confluence Support website. For more information about page and blog restrictions, see [Page Restrictions](#) on the Confluence Support website.

The Confluence group and user names are mapped as follows:

- `_group_ids` – Group names are present on spaces, pages, and blogs where there are restrictions. They are mapped from the name of the group in Confluence. Group names are always lower case.
- `_user_id` – User names are present on the space, page, or blog where there are restrictions. They are mapped depending on the type of Confluence instance you are using.
 - Server – The `_user_id` is the username. The username is always lower case.
 - Cloud – The `_user_id` is the account ID of the user.

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for Google Drive data sources

A Google Workspace Drive data source returns user and group information for Google Drive users and groups. Group and domain membership are mapped to the `_group_ids` index field. The Google Drive user name is mapped to the `_user_id` field.

When you provide one or more user email addresses in the `Query` operation, only documents that have been shared with those email addresses are returned. The following `AttributeFilter` parameter only returns documents shared with "martha@example.com".

```
"AttributeFilter": {  
    "EqualsTo": {  
        "Key": "_user_id",  
        "Value": {  
            "StringValue": "martha@example.com"  
        }  
    }  
}
```

If you provide one or more group email addresses in the query, only documents shared with the groups are returned. The following `AttributeFilter` parameter only returns documents shared with the "hr@example.com" group.

```

"AttributeFilter": {
    "EqualsTo": {
        "Key": "_group_ids",
        "Value": {
            "StringListValue": ["hr@example.com"]
        }
    }
}

```

If you provide the domain in the query, all documents shared with the domain are returned. The following `AttributeFilter` parameter returns documents shared with the "example.com" domain.

```

"AttributeFilter": {
    "EqualsTo": {
        "Key": "_group_ids",
        "Value": {
            "StringListValue": ["example.com"]
        }
    }
}

```

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for Microsoft OneDrive data sources

Amazon Kendra retrieves user and group information from Microsoft OneDrive when it indexes the documents on the site. The user and group information is taken from the underlying Microsoft SharePoint site that hosts OneDrive.

When you use a OneDrive user or group for user context filtering, calculate the ID as follows:

1. Get the site name. For example, <https://host.onmicrosoft.com/sites/siteName>.
2. Take the MD5 hash of the site name. For example, 430a6b90503eef95c89295c8999c7981.
3. Create the user email or group ID by concatenating the MD5 hash with a vertical bar (|) and the ID. For example, if a group name is "site owners", the group ID would be:

```
"430a6b90503eef95c89295c8999c7981|site owners"
```

For the user name "someone@host.onmicrosoft.com," the user ID would be the following:

```
"430a6b90503eef95c89295c8999c7981|someone@host.onmicrosoft.com"
```

Send the user or group ID to Amazon Kendra as the `_user_id` or `_group_ids` attribute when you call the [Query \(p. 384\)](#) operation. For example, the AWS CLI command that uses a group to filter the query response looks like this:

```

aws kendra query \
    --index-id index ID
    --query-text "query text"
    --attribute-filter '{
        "EqualsTo": {
            "Key": "_group_ids",
            "Value": {"StringValue": "430a6b90503eef95c89295c8999c7981|site
owners"}
        }
    }'

```

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for Amazon S3 data sources

You add user context filtering to a document in an Amazon S3 data source using a metadata file associated with the document. You add the information to the `AccessControlList` field in the JSON document. For more information about adding metadata to the documents indexed from an Amazon S3 data source, see [S3 document metadata \(p. 88\)](#).

You provide three pieces of information:

- The access that the entity should have. You can say `ALLOW` or `DENY`.
- The type of entity. You can say `USER` or `GROUP`.
- The name of the entity.

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for Salesforce data sources

A Salesforce data source returns user and group information from Salesforce access control list (ACL) entities. You can apply user context filtering to Salesforce standard objects and chatter feeds. User context filtering is not available for Salesforce knowledge articles.

For standard objects, the `_user_id` and `_group_ids` are used as follows:

- `_user_id` – The user name of the Salesforce user.
- `_group_ids` –
 - Name of the Salesforce Profile
 - Name of the Salesforce Group
 - Name of the Salesforce UserRole
 - Name of the Salesforce PermissionSet

For chatter feeds, the `_user_id` and `_group_ids` are used as follows:

- `_user_id` – The user name of the Salesforce user. Only available if the item is posted in the user's feed.
- `_group_ids` – Group IDs are used as follows. Only available if the feed item is posted in a chatter or collaboration group.
 - The name of the chatter or collaboration group.
 - If the group is public, `PUBLIC:ALL`.

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for ServiceNow data sources

User context filtering isn't currently supported for ServiceNow.

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for Microsoft SharePoint data sources

Amazon Kendra retrieves user and group information from Microsoft SharePoint when it indexes the documents on the site. To filter your documents, provide user and group information when you call the [Query](#) operation.

To filter using a user name, use the user's email address. For example, `johnstiles@example.com`.

When you use a SharePoint group for user context filtering, calculate the group ID as follows:

1. Get the site name. For example, `https://host.onmicrosoft.com/sites/siteName`.
2. Take the SHA256 hash of the site name. For example, `430a6b90503eef95c89295c8999c7981`.
3. Create the group ID by concatenating the SHA256 hash with a vertical bar (|) and the group name. For example, if the group name is "site owners", the group ID would be:

```
"430a6b90503eef95c89295c8999c7981|site owners"
```

Send the group ID to Amazon Kendra as the `_group_ids` attribute when you call the [Query](#) (p. 384) operation. For example, the AWS CLI command looks like this:

```
aws kendra query \
    --index-id index ID
    --query-text "query text"
    --attribute-filter '{
        "EqualsTo": {
            "Key": "_group_ids",
            "Value": {"StringValue": "430a6b90503eef95c89295c8999c7981|site
owners"}}
    }'
```

You can add up to 200 entries in the `AccessControlList` field.

User context filtering for Amazon WorkDocs data sources

When you use an Amazon WorkDocs data source, Amazon Kendra gets user and group information from the Amazon WorkDocs instance.

The Amazon WorkDocs group and user names are mapped as follows:

- `_group_ids` – Group names are present on files where there are set access permissions. They are mapped from the name of the group in Amazon WorkDocs.
- `_user_id` – User names are present on files where there are set access permissions. They are mapped from the user name in Amazon WorkDocs.

You can add up to 200 entries in the `AccessControlList` field.

Query responses

A call to [Query](#) (p. 384) returns information about the results of a search. The results are in an array of [QueryResultItem](#) (p. 504) objects (`ResultItems`). Each `QueryResultItem` includes a summary of the result. Document attributes associated with the query result are included.

Summary information

The summary information varies depending on the type of result. In each case, it includes document text that matches the search term. It also includes highlight information that you can use to highlight the search text in your application's output. For example, if the search term is *what is the height of the Space Needle?*, the summary information includes text location for the words *height* and *space needle*. For information about response types, see [Response types \(p. 147\)](#).

Document attributes

Each result contains document attributes for the document that matches a query. Some of the attributes are predefined, such as `DocumentId`, `DocumentTitle`, and `DocumentUri`. Others are custom attributes that you define. You can use document attributes to filter the response from the `Query` operation. For example, you might want only the documents written by a specific author or a specific version of a document. For more information, see [Filtering queries \(p. 134\)](#). You specify document attributes when you add documents to an index. For more information, see [Creating custom document attributes \(p. 121\)](#).

The following is sample JSON code for a query result. Note the document attributes in `DocumentAttributes` and `AdditionalAttributes`.

```
{
    "QueryId": "query-id",
    "ResultItems": [
        {
            "Id": "result-id",
            "Type": "ANSWER",
            "AdditionalAttributes": [
                {
                    "Key": "AnswerText",
                    "ValueType": "TEXT_WITH_HIGHLIGHTS_VALUE",
                    "Value": {
                        "TextWithHighlightsValue": {
                            "Text": "text",
                            "Highlights": [
                                {
                                    "BeginOffset": 55,
                                    "EndOffset": 90,
                                    "TopAnswer": false
                                }
                            ]
                        }
                    }
                }
            ],
            "DocumentId": "document-id",
            "DocumentTitle": {
                "Text": "title"
            },
            "DocumentExcerpt": {
                "Text": "text",
                "Highlights": [
                    {
                        "BeginOffset": 0,
                        "EndOffset": 300,
                        "TopAnswer": false
                    }
                ]
            },
            "DocumentURI": "uri",
            "DocumentAttributes": [],
            "ScoreAttributes": "score",
            "FeedbackToken": "token"
        }
    ]
}
```

```
        },
        {
            "Id": "result-id",
            "Type": "DOCUMENT",
            "AdditionalAttributes": [],
            "DocumentId": "document-id",
            "DocumentTitle": {
                "Text": "title",
                "Highlights": []
            },
            "DocumentExcerpt": {
                "Text": "text",
                "Highlights": [
                    {
                        "BeginOffset": 74,
                        "EndOffset": 77,
                        "TopAnswer": false
                    }
                ]
            },
            "DocumentURI": "uri",
            "DocumentAttributes": [
                {
                    "Key": "_source_uri",
                    "Value": {
                        "StringValue": "uri"
                    }
                }
            ],
            "ScoreAttributes": "score",
            "FeedbackToken": "token",
        }
    ],
    "FacetResults": [],
    "TotalNumberOfResults": number
}
```

Query suggestions

Amazon Kendra *Query suggestions* can help customers type their search queries faster and guide their search. Amazon Kendra suggests queries that are relevant to your users based on popular queries in the *query log*, or query history. Amazon Kendra uses all the queries your users search for and learns from these queries to make suggestions to your users.

For example, a user starts typing the query 'upcoming events'. Amazon Kendra has learned from the query log that many users have searched for 'upcoming events 2050' many times. The user sees 'upcoming events 2050' appear directly underneath their search bar, auto-completing their search query. The user selects this query suggestion by choosing the first search result, which is the document 'Upcoming events: What's happening in 2050'.

You can specify how Amazon Kendra selects eligible queries to suggest to your users. You can also block certain queries from being suggested to your users.

For more information, see [Suggesting popular search queries](#).

Tuning responses

By default, query responses are sorted by the relevance score that Amazon Kendra determines for each result in the response.

You can modify the effect of a field or attribute on the search relevance through *relevance tuning*. To quickly test relevance tuning, use the [Query \(p. 384\)](#) operation to pass in tuning configurations in the query. Then you can see the different search results that you get from different configurations. Relevance tuning at the query level is not supported in the console. You cannot tune fields or attributes that are of the type `StringList`.

You can tune results for any built-in or custom attribute of the following types:

- Date value
- Long value
- String value

You can't sort attributes of the following type:

- String list values

Rank and tune document results (AWS SDK)

Set the `Searchable` parameter to true to boost the document metadata configuration.

To tune an attribute in a query, set the `DocumentRelevanceOverrideConfigurations` parameter of the `Query` operation and specify the name of the attribute to tune.

The following JSON example shows a `DocumentRelevanceOverrideConfigurations` structure that overrides the tuning for the attribute called "department" in the index.

```
"DocumentRelevanceOverrideConfigurations" : [
    {
        "Name": "department",
        "Relevance": {
            "Importance": 1,
            "valueImportanceMap": {
                "IT": 3,
                "HR": 7
            }
        }
    }
]
```

Sorting responses

By default, query responses are sorted by the relevance score that Amazon Kendra determines for each result in the response. To change the sort order, make a document attribute sortable and then configure Amazon Kendra to use that attribute to sort responses.

Amazon Kendra uses the sorting attribute as part of the criteria for the documents returned by the query. For example, the results returned by a query sorted by `_created_at` might not contain the same results as a query sorted by `_version`.

You can sort results on any built-in or custom attribute of the following types:

- Date value
- Long value
- String value

You can't sort attributes of the following type:

- String list values

You can sort on only one document attribute in each query. Queries return 100 results. If there are fewer than 100 documents with the sorting attribute set, documents without a value for the sorting attribute are returned at the end of the results, sorted by relevance to the query.

To sort document results (AWS SDK)

1. To use the [UpdateIndex \(p. 411\)](#) operation to make an attribute sortable, set the `Sortable` parameter to `true`. The following JSON example is a `DocumentMetadataConfigurationUpdates` structure that adds an attribute called "Department" to the index and makes it sortable.

```
"DocumentMetadataConfigurationUpdates": [  
    {  
        "Name": "Department",  
        "Type": "STRING_VALUE",  
        "Search": {  
            "Sortable": "true"  
        }  
    }  
]
```

2. To use a sortable attribute in a query, set the `SortingConfiguration` parameter of the [Query \(p. 384\)](#) operation. Specify the name of the attribute to sort and whether to sort the response in ascending or descending order.

The following JSON example shows the `SortingConfiguration` parameter that you use to sort the results of a query by the "Department" attribute in ascending order.

```
"SortingConfiguration": {  
    "DocumentAttributeKey": "Department",  
    "SortOrder": "ASC"  
}
```

To sort document results (console)

1. To make an attribute sortable in the console, choose **Sortable** in the attribute definition. You can make an attribute sortable when you create the attribute, or you can modify it later.
2. To sort a query response in the console, choose the attribute to sort the response from the **Sort** menu. Only attributes that were marked sortable during datasource configuration appear in the list.

Response types

Amazon Kendra returns three types of query response.

- Answer
- Document
- Question and answer

The type of the response is returned in the `Type` response field of the [QueryResultItem \(p. 504\)](#) operation.

Answer

Amazon Kendra detected one or more question answers in the response. A factoid is the response to a who, what, when, or where question such as *Where is the nearest service center to me?* Amazon Kendra returns text in the index that best matches the query. The text is in the `AnswerText` field and contains highlight information for the search term within the response text. `AnswerText` includes the full document excerpt with highlighted text, while `DocumentExcerpt` includes the truncated (290 characters) document excerpt with highlighted text.

Amazon Kendra only returns one answer per document, and that is the answer with the highest confidence. To return multiple answers from a document, you need to split the document into multiple documents.

```
{
    'AnswerText': {
        'TextWithHighlights': [
            {
                'BeginOffset': 271,
                'EndOffset': 279,
                'TopAnswer': False
            },
            {
                'BeginOffset': 481,
                'EndOffset': 489,
                'TopAnswer': False
            },
            {
                'BeginOffset': 547,
                'EndOffset': 555,
                'TopAnswer': False
            },
            {
                'BeginOffset': 764,
                'EndOffset': 772,
                'TopAnswer': False
            }
        ],
        'Text': 'Asynchronous operations can\n' 'also process
\n' 'documents that are in PDF\n' 'format. Using PDF format files allows you to process\n' 'multi-
page\n' 'documents.\n' 'For information about how Amazon Textract represents
\n' 'documents as Block objects,
        ' 'see Documents and Block Objects.\n' '\n' '\n' 'For information about document\n' 'limits,
        see Limits in Amazon Textract.
\n' '\n' '\n' '\n' 'The Amazon Textract synchronous\n' 'operations can process documents stored in an Amazon
\n' 'S3 Bucket or you can pass\n' 'base64 encoded image bytes.\n' 'For more information,
        see\n' 'Calling Amazon Textract Synchronous Operations.\n' 'Asynchronous operations require input documents
\n' 'to be supplied in an Amazon\n' 'S3 Bucket.'
    },
    'DocumentExcerpt': {
        'Highlights': [
            {
                'BeginOffset': 0,
                'EndOffset': 300,
                'TopAnswer': False
            }
        ],
        'Text': 'Asynchronous operations can\n' 'also process
\n' 'documents that are in PDF\n' 'format. Using PDF format files allows you to process\n' 'multi-page
\n' 'documents.\n' 'For information about how Amazon Textract represents\n' ''
    },
    'Type': 'ANSWER'
}
```

```
}
```

Document

Amazon Kendra returns ranked documents for those that match the search term. The ranking is based on the confidence that Amazon Kendra has in the accuracy of the search result. Information about the matching document is returned in the [QueryResultItem \(p. 504\)](#). It includes the title of the document. The excerpt includes highlight information for search text and the section of matching text in the document. The URI for matching documents is in the `SourceURI` document attribute. The following sample JSON shows the document summary for a matching document.

```
{
  'DocumentTitle': {
    'Highlights': [
      {
        'BeginOffset': 7,
        'EndOffset': 15,
        'TopAnswer': False
      },
      {
        'BeginOffset': 97,
        'EndOffset': 105,
        'TopAnswer': False
      }
    ],
    'Text': 'AmazonTextractAPIPermissions: Actions,
\n''Permissions,
andResourcesReference--'AmazonTextract'
},
  'DocumentExcerpt': {
    'Highlights': [
      {
        'BeginOffset': 68,
        'EndOffset': 76,
        'TopAnswer': False
      },
      {
        'BeginOffset': 121,
        'EndOffset': 129,
        'TopAnswer': False
      }
    ],
    'Text': '...LoggingandMonitoring\tMonitoring
\n''\tCloudWatchMetricsforAmazonTextract
\n''\tLoggingAmazonTextractAPICallswithAWSCloudTrail\n''\tAPIReference\tActions
\tAnalyzeDocument\n''\tDetectDocumentText\n''\tGetDocumentAnalysis...
},
  'Type': 'DOCUMENT'
}
```

Question and answer

A question and answer response is returned when Amazon Kendra matches a question with one of the frequently asked questions in your index. The response includes the matching question and answer in the [QueryResultItem \(p. 504\)](#) field. It also includes highlight information for query terms detected in query string. The following JSON shows a question and answer response. Note that the response includes the question text.

```
{
  'AnswerText': {
```

```
'TextWithHighlights': [
    ],
    'Text': '605feet'
},
'DocumentExcerpt': {
    'Highlights': [
        {
            'BeginOffset': 0,
            'EndOffset': 8,
            'TopAnswer': False
        }
    ],
    'Text': '605feet'
},
'Type': 'QUESTION_ANSWER',
'QuestionText': {
    'Highlights': [
        {
            'BeginOffset': 12,
            'EndOffset': 18,
            'TopAnswer': False
        },
        {
            'BeginOffset': 26,
            'EndOffset': 31,
            'TopAnswer': False
        },
        {
            'BeginOffset': 32,
            'EndOffset': 38,
            'TopAnswer': False
        }
    ],
    'Text': 'whatistheheightoftheSpaceNeedle?'
}
}
```

For information about adding question and answer text to an index, see [Adding questions and answers directly to an index \(p. 80\)](#)

Tuning search relevance

Amazon Kendra queries produce search results ranked by their relevance. The searchable fields or attributes in the index all contribute to this ranking.

You can modify the effect of a field or attribute on the search relevance through *relevance tuning*. Tuning search relevance can either be done manually at the index level, where you set tuning configurations for your index, or at the query level by overriding configurations set at the index level.

When you use relevance tuning, a result is given a boost in the response when the query includes terms that match the field or attribute. You also specify how much of a boost the document receives when there is a match. Relevance tuning doesn't cause Amazon Kendra to include a document in the query response, it is only one of the factors that Amazon Kendra uses to determine the relevance of a document.

You can boost specific fields or attributes in your index to assign more importance to specific responses. For example when someone searches for "When is re:Invent?" you could boost the relevance of document freshness in the "_last_update_at" field. Or, in an index of research reports, you could boost a specific data source in the "source" field.

You can also boost documents based on votes or view counts which is common in forums and other support knowledge bases. You can combine boosts, for example to boost documents that are viewed more as well as more recent.

You set the amount of boost that a document receives by using the `Importance` parameter. The higher the `Importance`, the more the field or attribute boosts the relevance of a document. When you tune your index or tune at the query level, increase the value of the `Importance` parameter in small increments until you get the effect that you want. To determine if you are improving search results, perform the search and compare the results to previous queries.

You can specify date, number, or string attributes to tune an index or tune at the query level. You can also tune fields or attributes that are of the type `StringList`. Each field or attribute has specific criteria for when it boosts a result.

- **Date fields or attributes** – There are three specific criteria for date fields, `Duration`, `Freshness` and `RankOrder`.
 - `Duration` sets the time period that the boost applies to. For example, if you set the time period to 86400 seconds (i.e. one day), the boost begins to lessen after one day. The higher the importance, the faster the boost effect lessens.
 - `Freshness` determines how recent a document is when applied to a field or attribute. If you apply `Freshness` to either the field for date created or date last updated, then a more recently created or last updated document is considered "fresher" than an older document. For example, if document 1 was created on November 14, and document 2 was created on November 5, document 1 is "fresher" than document 2. And if document 1 was last updated on November 14, and document 2 was last updated on November 20, document 2 is "fresher" than document 1. The fresher the document, the more this boost is applied. You can only have one `Freshness` field in your index.
 - `RankOrder` applies the boost in either ascending or descending order. If you specify `ASCENDING`, later dates have precedence . If you specify `DESCENDING`, earlier dates have precedence.
- **Number fields or attributes** – For number fields or attributes, you can specify the rank order that Amazon Kendra should use when determining the relevance of the field or attribute. If you specify `ASCENDING`, then higher numbers are given precedence. If you specify `DESCENDING`, then lower numbers have precedence.
- **String fields or attributes** – For string fields or attributes, you can create categories of a field to give each category a different boost. For example, if you boost a field or attribute called "Department", you

can give a different boost to documents from "HR" than to documents from "Legal". You can boost a field or attribute of the type `String` and `StringList`.

Relevance tuning at the index level

You tune the relevance of a field or attribute at the index level by using either the [console](#) to set tuning in the index details or the [UpdateIndex \(p. 411\)](#) operation. The following example sets the `_last_updated_at` field as the `Freshness` field for a document.

```
"DocumentMetadataConfigurationUpdates" : [
    {
        "Name": "_last_updated_at",
        "Type": "DATE_VALUE",
        "Relevance": {
            "Freshness": TRUE,
            "Importance": 2
        }
    }
]
```

The following example applies different importance to the different categories in the "department" field.

```
"DocumentMetadataConfigurationUpdates" : [
    {
        "Name": "department",
        "Type": "STRING_VALUE",
        "Relevance": {
            "Importance": 2,
            "ValueImportanceMap": {
                "HR": 3,
                "Legal": 1
            }
        }
    }
]
```

Relevance tuning at the query level

You tune the relevance of a field or attribute at the query level by using the [Query \(p. 384\)](#) operation. Tuning at the query level can speed up the process of testing relevance tuning because you don't need to manually update the tuning configurations in the index for each test. You can tune the relevance of a document by passing tuning configurations in the query. Then you can see the different results that you get from different configurations. A configuration that is passed in the query overrides the configuration that is set at the index level.

The following example overrides the importance applied to the "department" field and each department category set at the index level, shown in the above example. When a user inputs their search query, the "department" field has a fair level of importance and the Legal department has more importance than the HR department.

```
"DocumentRelevanceOverrideConfigurations" : [
    {
        "Name": "department",
        "Type": "STRING_VALUE",
        "Relevance": {
            "Importance": 5,
            "ValueImportanceMap": {
                "HR": 2,
                "Legal": 8
            }
        }
    }
]
```

```
    }
```

Relevance tuning at the query level is not supported in the console.

Suggesting popular search queries

Amazon Kendra *Query suggestions* can help customers type their search queries faster and guide their search.

Amazon Kendra suggests queries relevant to your users based on popular queries in the *query log*, or query history. Amazon Kendra uses all of the queries your users search for and learns from these queries to make suggestions to your users. Amazon Kendra currently suggests popular queries to users when they start typing their query. Amazon Kendra suggests a query if the *prefix* or first few characters of the query matches what the user starts typing as their query.

For example, a user starts typing the query 'upcoming events'. Amazon Kendra has learned from the query log that many users have searched for 'upcoming events 2050' many times. The user sees 'upcoming events 2050' appear directly underneath their search bar, auto-completing their search query. The user selects this query suggestion by choosing the first search result, which is the document 'Upcoming events: What's happening in 2050'.

You can specify how Amazon Kendra selects eligible queries to suggest to your users. For example, you can specify that a query suggestion must have been searched by at least 10 unique users (default is 3), have been searched within the last 30 days, and does not contain any words or phrases from your [block list](#). Amazon Kendra requires a query to have at least one search result and contain at least one word of more than four characters.

Query suggestions are case insensitive. Amazon Kendra converts the query prefix and the suggested query to lower case, ignores all single and double quotation marks, and replaces multiple white space characters with a single space. Amazon Kendra matches all other special characters as they are. Amazon Kendra does not show any suggestions if a user types fewer than two characters or more than 60 characters.

You can retrieve query suggestions relevant to your users by using the [GetQuerySuggestions](#) operation. Query suggestions are enabled by default at no additional cost. You can disable query suggestions at any time by using the [UpdateQuerySuggestionsConfig](#) operation. You can test your settings before you apply suggestions to your search application in two ways:

- By using the [UpdateQuerySuggestionsConfig](#) operation.
- In the console in **Query suggestions settings**.

You use the [GetQuerySuggestions](#) operation to integrate query suggestions with your console application for your users to start seeing the suggestions.

Query suggestions settings

You can configure the following settings using the [UpdateQuerySuggestionsConfig](#) operation:

- **Mode** – Query suggestions are either `ENABLED` or `LEARN_ONLY`. Amazon Kendra enables query suggestions by default. `LEARN_ONLY` turns off query suggestions. If turned off, Amazon Kendra continues to learn suggestions but doesn't make query suggestions to users.
- **Query log time window** – How recent your queries are in your query log time window. The time window is an integer value for the number of days from current day to past days.

- **Queries without user information** – Set to `true` to include all queries or set to `false` to only include queries with user information.
- **Unique users** – The minimum number of unique users who must search a query for the query to be eligible to suggest to your users. This number is an integer value.
- **Query count** – The minimum number of times a query must be searched for the query to be eligible to suggest to your users. This number is an integer value.

These settings affect how queries are selected as popular queries to suggest to your users. How you tune your settings will depend on your specific needs, for example:

- If your users usually search once a month on average, then you can set the number of days in the query log time window to 30, so that you capture most of your users' recent queries before they become outdated in the time window.
- If only a small number of your queries include user information, and you don't want to suggest queries based on a small sample size, then you can set queries to include all users.
- If you define *popular queries* as being searched by at least 10 unique users and searched at least 100 times, then you set the unique users to 10 and the query count to 100.

Your changes to settings might not take effect right away. You can track the settings changes by using the [DescribeQuerySuggestionsConfig](#) operation. The time for your updated settings to take effect depends on the updates that you make and the number of search queries in your index.

Console

To check that query suggestions are enabled and ready

1. In the left navigation pane, under **Indexes**, go to your index, and then for **Enrichments**, select **Query suggestions**.
2. On the **Query suggestions** page, go to **Query suggestions settings** and do the following:
 - a. Check that **Status** is **Enabled**.
 - b. Check that the number of **Queries ready for suggestions** is more than 0.

To edit query suggestions settings

1. In the left navigation pane, under **Indexes**, go to your index, and then for **Enrichments**, select **Query suggestions**.
2. On the **Query suggestions** page, go to **Query suggestions settings** and choose **Edit**.

To clear suggestions

1. In the left navigation pane, under **Indexes**, go to your index, and then for **Enrichments**, select **Query suggestions**.
2. On the **Query suggestions** page, go to **Query suggestions settings** and then for **Actions**, choose **Clear suggestions**.

CLI

To retrieve query suggestions

```
aws kendra get-query-suggestions \  
--index-id index-id \  
...
```

```
--query-text "query-text" \
--max-suggestions-count 1      // If you want to limit the number of suggestions.
```

To update query suggestions

For example, to change the query log time window and the minimum number of times a query must be searched:

```
aws kendra update-query-suggestions-config \
--index-id index-id \
--query-log-look-back-window-in-days 30 \
--minimum-query-count 100
```

Note

The time for your updated settings to take effect depends on the updates you make and the number of search queries in your index.

To clear suggestions

```
aws kendra clear-query-suggestions \
--index-id index-id \
```

Note

Amazon Kendra learns new suggestions based on new queries added to the query log from the time you cleared suggestions.

Python

To retrieve query suggestions

```
import boto3
from botocore.exceptions import ClientError

kendra = boto3.client("kendra")

print("Getting suggestions for a query.")

index_id = "index-id"
query_text = "query"

try:
    query_suggestions_response = kendra.get_querySuggestions(
        IndexId = index_id,
        QueryText = query_text,
        MaxSuggestionsCount = 5
    )

    # Printing out the suggestions you received.
    if ("Suggestions" in query_suggestions_response.keys()):
        for (suggestion: query_suggestions_response["Suggestions"]):
            print(suggestion["Value"]["Text"]["Text"]);
    }
}

except ClientError as e:
    print("%s" % e)

print("Program ends.")
```

To update query suggestions,

For example, to change the query log time window and the minimum number of times a query must be searched:

```
import boto3
from botocore.exceptions import ClientError
import pprint
import time

kendra = boto3.client("kendra")

print("Updating query suggestions settings/configuration for an index.")

query_log_look_back_window_in_days = 30
minimum_query_count = 100
thesaurus_role_arn = "role-arn"

index_id = "index-id"

s3_bucket_name = "bucket-name"
s3_key = "thesaurus-file"
source_s3_path= {
    'Bucket': s3_bucket_name,
    'Key': s3_key
}

try:
    kendra.update_querySuggestionsConfig(
        MinimumQueryCount = minimum_query_count,
        IndexId = index_id,
        QueryLogLookBackWindowInDays = query_log_look_back_window_in_days
    )

    print("Wait for Amazon Kendra to update the query suggestions.")

    while True:
        # Get query suggestions description of settings/configuration.
        query_sugg_config_response = kendra.describe_querySuggestionsConfig(
            IndexId = index_id
        )

        # If status is not UPDATING, then quit.
        status = query_sugg_config_response["Status"]
        print("Updating query suggestions config. Status: " + status)
        if status != "UPDATING":
            break
        time.sleep(60)

except ClientError as e:
    print("%s" % e)

print("Program ends.")
```

To clear suggestions

```
import boto3
from botocore.exceptions import ClientError

kendra = boto3.client("kendra")

print("Clearing out query suggestions for an index.")

index_id = "index-id"

try:
```

```

kendra.clear_querySuggestions(
    IndexId = index_id
)

# Confirm last cleared date-time and there are no suggestions:
query_sugg_config_response = kendra.describe_querySuggestionsConfig(
    IndexId = index_id
)
print("Query Suggestions last cleared at: " +
str(query_sugg_config_response["LastClearTime"]));
print("Number of suggestions available to use after clearing: " +
str(query_sugg_config_response["TotalSuggestionsCount"]));

except ClientError as e:
    print("%s" % e)

print("Program ends.")

```

You can check your current settings by using the [DescribeQuerySuggestionsConfig](#) operation. Additionally, this operation shows the following information about your query suggestions for an index:

- **Mode** – Query suggestions are either `ENABLED` or `LEARN_ONLY`. Amazon Kendra enables query suggestions by default. `LEARN_ONLY` turns off query suggestions. You can change the mode by using the [UpdateQuerySuggestionsConfig](#) operation. If turned off, Amazon Kendra continues to learn suggestions but doesn't make query suggestions to users.
- **Status** – Query suggestions are either `ACTIVE` or `UPDATING`.
- **Suggestions count** – The total number of queries ready to be suggested to your users. If the count is much lower than you expected, it could be because Amazon Kendra needs more queries to learn from or your current query suggestions settings are too strict.
- **Suggestions last build time** – The last time suggestions were updated. Amazon Kendra automatically updates suggestions every 24 hours, or when you change a setting or when you apply a [block list](#).
- **Suggestions last cleared time** – The last time suggestions were cleared.

Block certain queries from suggestions

A *block list* stops Amazon Kendra from suggesting certain popular queries to your users. It is a list of words or phrases you want to exclude from query suggestions. Amazon Kendra excludes queries containing an exact match of the words or phrases in the block list.

You can use a block list to safeguard against offensive words or phrases that commonly appear in your query log and that Amazon Kendra could select as suggestions. A block list can also prevent Amazon Kendra from suggesting popular queries that contain information that is not ready to be publicly released or announced. For example, if your users commonly query about a release of a new product that you don't want to suggest because you are not ready to release soon, then you can block queries that contain the product name and product information from suggestions.

You can create a block list for queries by using the [CreateQuerySuggestionsBlockList](#) operation. You put each block word or phrase on a separate line in a text file. Then you upload the text file to your S3 bucket and provide the path or location to the file on Amazon S3. Amazon Kendra currently supports creating only one block list.

You can replace the text file of your blocked words and phrases in your S3 bucket and use the [UpdateQuerySuggestionsBlockList](#) to update the block list in Amazon Kendra.

You can use the [DescribeQuerySuggestionsBlockList](#) operation to get the status of your block list and other useful information, such as when your block list was last updated, how many words or phrases

are in your current block list, and helpful error messages when creating a block list. You can also use the [ListQuerySuggestionsBlockLists](#) operation to get a list of block list summaries for an index.

To delete your block list, use the [DeleteQuerySuggestionsBlockList](#) operation.

Your updates to the block list might not take effect right away. You can track updates by using the [DescribeQuerySuggestionsBlockList](#) operation.

Console

To import a block list

1. In the left navigation pane, under **Indexes**, go to your index, and then for **Enrichments**, select **Query suggestions**.
2. On the **Query suggestions** page, go to **Block list** and choose **Import block list**.
3. In the **Block list file location on S3** field enter the location of your block list text file in your S3 bucket.

To update and reload a block list

1. In the left navigation pane, under **Indexes**, go to your index, and then for **Enrichments**, select **Query suggestions**.
2. Replace the block list text file in the S3 bucket with your updated file.
3. On the **Query suggestions** page, go to **Block list** and choose **Reload**.

To delete a block list

1. In the left navigation pane, under **Indexes**, go to your index, and then for **Enrichments**, select **Query suggestions**.
2. On the **Query suggestions** page, go to **Block list** and then choose **Delete**.

CLI

To create a block list

```
aws kendra create-query-suggestions-block-list \
--index-id index-id \
--name "block-list-name" \
--description "block-list-description" \
--source-s3-path "Bucket=bucket-name,Key=query-suggestions/block_list.txt" \
--role-arn role-arn
```

To update a block list

```
aws kendra update-query-suggestions-block-list \
--index-id index-id \
--name "block-list-name" \
--description "block-list-description" \
--source-s3-path "Bucket=bucket-name,Key=query-suggestions/block_list.txt" \
--role-arn role-arn
```

To delete a block list

```
aws kendra delete-query-suggestions-block-list \
--index-id index-id \
```

```
--id block-list-id
```

Python

To create a block list

```
import boto3
from botocore.exceptions import ClientError
import pprint
import time

kendra = boto3.client("kendra")

print("Create a query suggestions block list")

block_list_name = "block-list-name"
block_list_description = "block-list-description"
block_list_role_arn = "role-arn"

index_id = "index-id"

s3_bucket_name = "bucket-name"
s3_key = "query-suggestions/block_list.txt"
source_s3_path = {
    'Bucket': s3_bucket_name,
    'Key': s3_key
}

try:
    block_list_response = kendra.create_querySuggestionsBlockList(
        Description = block_list_description,
        Name = block_list_name,
        RoleArn = block_list_role_arn,
        IndexId = index_id,
        SourceS3Path = source_s3_path
    )

    print(block_list_response)

    block_list_id = block_list_response["Id"]

    print("Wait for Kendra to create the block list.")

    while True:
        # Get block list description
        block_list_description = kendra.describeQuerySuggestionsBlockList(
            Id = block_list_id,
            IndexId = index_id
        )
        # If status is not CREATING, then quit.
        status = block_list_description["Status"]
        print("Creating block list. Status: " + status)
        if status != "CREATING":
            break
        time.sleep(60)
except ClientError as e:
    print("%s" % e)

print("Program ends.")
```

To update a block list

```
import boto3
```

```
from botocore.exceptions import ClientError
import pprint
import time

kendra = boto3.client("kendra")

print("Update a block list for query suggestions.")

block_list_name = "block-list-name"
block_list_description = "block-list-description"
block_list_role_arn = "role-arn"

block_list_id = "block-list-id"
index_id = "index-id"

s3_bucket_name = "bucket-name"
s3_key = "query-suggestions/block_list_updated.txt"
source_s3_path = {
    'Bucket': s3_bucket_name,
    'Key': s3_key
}

try:
    kendra.update_querySuggestionsBlockList(
        Id = block_list_id,
        IndexId = index_id,
        Description = block_list_description,
        Name = block_list_name,
        RoleArn = block_list_role_arn,
        SourceS3Path = source_s3_path
    )

    print("Wait for Amazon Kendra to update the block list.")

    while True:
        # Get block list description
        block_list_description = kendra.describeQuerySuggestionsBlockList(
            Id = block_list_id,
            IndexId = index_id
        )
        # If status is not UPDATING, then the update has finished.
        status = block_list_description["Status"]
        print("Updating block list. Status: " + status)
        if status != "UPDATING":
            break
        time.sleep(60)

except ClientError as e:
    print("%s" % e)

print("Program ends.")
```

To delete a block list

```
import boto3
from botocore.exceptions import ClientError

kendra = boto3.client("kendra")

print("Delete a block list for query suggestions.")

querySuggestionsBlockListId = "query-suggestions-block-list-id"
index_id = "index-id"

try:
```

```
kendra.delete_querySuggestionsBlockList(  
    Id = querySuggestionsBlockListId,  
    IndexId = indexId  
)  
  
except ClientError as e:  
    print("%s" % e)  
  
print("Program ends.")
```

Clear suggestions

You can clear query suggestions by using the [ClearQuerySuggestions](#) operation. Clearing suggestions deletes existing query suggestions only, not the queries in the query log. After you clear suggestions, Amazon Kendra learns new suggestions based on new queries added to the query log from the time you cleared suggestions.

No suggestions available

If you don't see suggestions for a query, it could be for one of the following reasons:

- There are not enough queries in your index for Amazon Kendra to learn from.
- Your query suggestions settings are too strict, resulting in most queries being filtered out from suggestions.
- You recently cleared suggestions, and Amazon Kendra still needs time for new queries to accumulate in order to learn new suggestions.

You can check your current settings using the [DescribeQuerySuggestionsConfig](#) operation.

Submitting feedback for incremental learning

Amazon Kendra uses incremental learning to improve search results. Using feedback from queries, incremental learning improves the ranking algorithms and optimizes search results for greater accuracy.

For example, suppose that your users search for the phrase "health care benefits." If users consistently choose the second result from the list, over time Amazon Kendra boosts that result to the first place result. The boost decreases over time, so if users stop selecting a result, Amazon Kendra eventually removes it and shows another more popular result instead. This helps Amazon Kendra prioritize results based on relevance, age, and content.

Incremental learning is enabled for all indexes and for all document types. For more information, see [Response types \(p. 147\)](#).

Amazon Kendra starts learning as soon as you provide feedback, though it can take over 24 hours to see the results of the feedback. Amazon Kendra provides three methods for you to submit feedback: the AWS console, a JavaScript library that you can include on your search results page, and an API that you can use.

Amazon Kendra accepts two types of user feedback:

- **Clicks** - Information about which query results the user chose. The feedback includes the result ID and the Unix timestamp of the date and time that the search result was chosen.

To submit click feedback, your application must collect click information from the activities of your users, and then submit that information to Amazon Kendra. You can collect click information with the console, the JavaScript library, and the Amazon Kendra API.

- **Relevance** - Information about the relevance of a search result, which the user typically provides. The feedback contains the result ID and a relevance indicator (`RELEVANT` or `NOT_RELEVANT`). The user determines the relevance information.

To submit relevance feedback, your application must provide a feedback mechanism that enables the user to choose the appropriate relevance for a query result, and then submit that information to Amazon Kendra. You can only collect relevance information with the console and the Amazon Kendra API.

Feedback is used while the index is active. Feedback only affects the index that it is submitted to, it can't be used across indexes or for different accounts.

You should provide additional user context when you query your Amazon Kendra index. When you provide user context, Amazon Kendra is able to tell if the feedback is provided by a single user or by multiple users and adjust search results accordingly.

When you provide user context, the feedback for the query is associated with the specific user provided in the context. If you don't specify user context, you can provide a visitor ID that is used to group and aggregate queries.

If you don't provide user context or a visitor ID, the feedback is anonymous and aggregated with other anonymous feedback.

The following code shows how to include user context as a token or the visitor ID.

```
response = kendra.query(  
    QueryText = query,
```

```
IndexId = index,
UserToken = {
    Token = "token"
})
OR

response = kendra.query(
QueryText = query,
IndexId = index,
VisitorId = "visitor-id")
```

For web applications, you can use cookies, locations, or browser users to generate a visitor ID for each user.

For head queries, the largest volume of queries, providing click-through feedback provides enough information to improve overall accuracy. For tail queries, those that are rare, subject matter experts should submit relevant and non-relevant feedback to improve accuracy for those queries.

In addition to the console, you can use one of two methods: a JavaScript library or the [SubmitFeedback](#) ([p. 396](#)) operation. You should only use one method of gathering feedback. For best results, you should submit feedback within 24 hours of making the query.

Topics

- [Using the Amazon Kendra JavaScript library to submit feedback \(p. 164\)](#)
- [Using the Amazon Kendra API to submit feedback \(p. 166\)](#)

Using the Amazon Kendra JavaScript library to submit feedback

Amazon Kendra provides a JavaScript library that you can use to add click feedback to your search results page. To use the library, you insert a script tag in your client code that displays the search result, then add information to each of the document links in your result list. When a user chooses a link to view a document, click information is sent to Amazon Kendra.

The library works with browsers that support JavaScript version ES6/ES2015.

Step 1: Insert a script tag into your Amazon Kendra search application

In your client code that renders the Amazon Kendra search results, insert a `<script>` tag and add a reference to the JavaScript library:

```
<script>
(function(w, d, s, c, g, n) {
    if(!w[n]) {
        w[n] = w[n] || function () {
            (w[n].q = w[n].q || []).push(arguments);
        }
        w[n].st = new Date().getTime();
        w[n].ep = g;
        var e = document.createElement(s),
            j = document.getElementsByTagName(s)[0];
        e.async = 1;
        e.src = c;
        e.type = 'module';
    }
})()
```

```

        j.parentNode.insertBefore(e, j);
    }
})(window, document, 'script',
'library download URL',
'feedback endpoint',
'kendraFeedback');
</script>

```

The script asynchronously downloads the JavaScript library from an Amazon Kendra hosted CDN and initializes a global variable called `kendraFeedback` that enables you to set optional parameters.

Replace `library download URL` and `feedback endpoint` with an identifier from the following table based on the region that hosts your Amazon Kendra index.

Region	Download URL	Feedback endpoint
us-east-1	https://d2zm0lpns956f8.cloudfront.net/ksf-v1.js	https://ujxwp5s92h.execute-api.us-east-1.amazonaws.com/prod/submit
us-west-2	https://d2iezfppnccoujy.cloudfront.net/ksf-v1.js	https://wg6nim909c.execute-api.us-west-2.amazonaws.com/prod/submit
eu-west-1	https://d3gptlxulu4us.cloudfront.net/ksf-v1.js	https://po2b11740b.execute-api.eu-west-1.amazonaws.com/prod/submit
ap-southeast-2	https://dopqntoe6z0ce.cloudfront.net/ksf-v1.js	https://oovf4nvjj7.execute-api.ap-southeast-2.amazonaws.com/prod/submit

For example, if your index is in US East (N. Virginia), `library download URL` is <https://d2zm0lpns956f8.cloudfront.net/ksf-v1.js> and `feedback endpoint` is <https://ujxwp5s92h.execute-api.us-east-1.amazonaws.com/prod/submit>.

There are two optional settings that you can make for the Amazon Kendra JavaScript library:

- `disableCookies` – By default, Amazon Kendra sets a cookie that uniquely identifies the user. Set this to `true` to disable the cookie.

```
kendraFeedback('disableCookie', 'true | false');
```

`searchDivClassName` – By default, Amazon Kendra monitors all links on your search results page for clicks. Set this to a `<div>` class name to monitor only links in the specified class.

```
kendraFeedback('searchDivClassName', 'class name');
```

Step 2: Add the feedback token to search results

On your result page, add an HTML attribute called `data-kendra-token` to the anchor tag or immediate parent div tag that contains a link to the document from the query response. For example:

```
<a href="document location" data-kendra-token="feedback token value"></a>
```

OR

```
<div data-url="document location" data-kendra-token="feedback token value"></div>
```

A query response contains a token in the `feedbackToken` field. The token uniquely identifies the response if the user chooses it. Assign the value of the token to the `data-kendra-token` attribute. The Amazon Kendra JavaScript library looks for this token when the user chooses the result and submits it to an Amazon Kendra endpoint as feedback.

The Amazon Kendra JavaScript library only submits the feedback token and other metadata such as the time the result was chosen and a unique visitor ID.

Step 3: Test the feedback script

To make sure that the JavaScript library is configured correctly and sending feedback to the right endpoint, do the following. This example uses the Chrome browser.

1. Open the Web developer tools in the browser. On Chrome, open the **Chrome menu** in the upper right corner of the browser, choose **More tools** and then choose **Developer tools**.
2. Make sure that there are no errors related to the Amazon Kendra JavaScript library in the console tab.
3. Make a search and choose any result. In the **Network** tab of the developer tools. You should see a request sent to the feedback endpoint, the token for the result, and a 200 OK status.

Using the Amazon Kendra API to submit feedback

To use the Amazon Kendra API to submit query feedback, use the [SubmitFeedback \(p. 396\)](#) operation. To identify the query, you supply the `IndexID` of the index that the query applies to, and the `QueryID` returned in the response from the [Query \(p. 384\)](#) operation.

The following example shows how to submit click and relevance feedback using the Amazon Kendra API. You can submit multiple sets of feedback through the `ClickFeedbackItems` and `RelevanceFeedbackItems` arrays. This example submits a single click and a single relevance feedback item. The feedback submittal uses the current time.

To submit feedback for a search (AWS SDK)

1. Use the following code and change the following values:
 - a. `index_id` - Change to the ID of the index that the query applies to.
 - b. `query_id` - Change to the query that you want to provide feedback on.
 - c. `result_id` - Change to the ID of the query result that you want to provide feedback on. The query response contains the result ID.
 - d. `relevance_value` - Change to either `RELEVANT` (the query result is relevant) or `NOT_RELEVANT` (the query result is not relevant).

Python

```
import boto3
import time

kendra = boto3.client('kendra')

index_id = '${indexID}'
query_id = '${queryID}'
result_id = '${resultID}'
feedback_item = {'ClickTime': int(time.time()),
                 'ResultId':result_id}
```

```
relevance_value = 'RELEVANT'
relevance_item = {'RelevanceValue': relevance_value,
                  'ResultId':result_id
                 }

response=kendra.submit_feedback(
    QueryId = query_id,
    IndexId = index_id,
    ClickFeedbackItems = [feedback_item],
    RelevanceFeedbackItems = [relevance_item]
)

print ('Submitted feedback for query: ' + query_id)
```

Java

```
package com.amazonaws.kendra;

import java.time.Instant;
import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.ClickFeedback;
import software.amazon.awssdk.services.kendra.model.RelevanceFeedback;
import software.amazon.awssdk.services.kendra.model.RelevanceType;
import software.amazon.awssdk.services.kendra.model.SubmitFeedbackRequest;
import software.amazon.awssdk.services.kendra.model.SubmitFeedbackResponse;

public class SubmitFeedbackExample {
    public static void main(String[] args) {
        KendraClient kendra = KendraClient.builder().build();

        SubmitFeedbackRequest submitFeedbackRequest = SubmitFeedbackRequest
            .builder()
            .indexId("anIndexID")
            .queryId("aQueryID")
            .clickFeedbackItems(
                ClickFeedback
                    .builder()
                    .clickTime(Instant.now())
                    .resultId("aResultId")
                    .build())
            .relevanceFeedbackItems(
                RelevanceFeedback
                    .builder()
                    .relevanceValue(RelevanceType.RELEVANT)
                    .resultId("aResultId")
                    .build())
            .build();

        SubmitFeedbackResponse response =
            kendra.submitFeedback(submitFeedbackRequest);

        System.out.println("Feedback is submitted");
    }
}
```

2. Run the code. After the feedback has been submitted, the code displays a message.

Adding synonyms to an index

To add synonyms to an index, you specify them in a thesaurus file. You can include business-specific terms in Amazon Kendra using synonyms. Generic English synonyms, such as `leader`, `head`, are built into Amazon Kendra and should not be included in a thesaurus file. Amazon Kendra supports synonyms for all response types, which include `DOCUMENT` response types and `QUESTION_ANSWER` or `ANSWER` response types.

Amazon Kendra provides different results based on your synonyms. For example, using the synonym pair `Dynamo`, `Amazon DynamoDB`, Amazon Kendra correlates `Dynamo` with `Amazon DynamoDB`. The query "What is dynamo?" then returns a document such as "What is Amazon DynamoDB?". With synonyms, Amazon Kendra can more easily pick up the correlation.

The thesaurus file uses the [Solr synonym format](#). Amazon Kendra has a limit on the number of thesauri per index. See [Quotas](#).

Synonyms can be useful in the following scenarios:

- Specialized terms that are not traditional English language synonyms such as `NLP`, `Natural Language Processing`.
- Proper nouns with complex semantic associations. These are nouns that the general public are unlikely to understand, for example, in machine learning, `cost`, `loss`, `model performance`.
- Different forms of product names, for example, `Elastic Compute Cloud`, `EC2`.
- Domain-specific or business-specific terms, such as product names. For example, `Route53`, `DNS`.

Do not use synonyms in the following scenarios:

- Generic English language synonyms such as `leader`, `head`. These synonyms are not domain-specific, and using synonyms in these scenarios might have unintended effects.
- Typographical errors such as `teh => the`.
- Morphological variants like the plurals and possessives of nouns, the comparative and superlative form of adjectives, and the past tense, past participle and progressive form of verbs. One example of comparative and superlative adjectives is `good`, `better`, `best`.
- Unigram (single word) stop words such as `WHO`. Unigram stop words are not allowed in the thesaurus and are excluded from search. For example, `WHO => World Health Organization` is rejected. You can use `W.H.O.` however as a synonym term, and you can use stop words as part of a multi-word synonym. For example, `of` is not allowed but `United States of America` is accepted.

Custom synonyms make it easy to improve Amazon Kendra's understanding of your business-specific terminology by expanding your queries to cover your business-specific synonyms. Although synonyms can improve search accuracy, it is important to understand how synonyms affect latency so you can optimize for this.

A general rule for synonyms is: the more terms in your query that are matched and expanded with synonyms, the greater potential impact on latency. Other factors that affect latency include the average size of documents indexed, the size of your index, any filtering on search results, and the overall load on your Amazon Kendra index. Queries that don't match any synonyms are not affected.

A general guideline for how synonyms affect latency:

Use case	Increase in latency*
Typical natural language or keyword queries of 3 to 5 words each	Less than 15 percent
1 query term expands to 3 synonyms	
Index of about 500,000 documents (averaging 10.48 KB of extracted text per document) or 30,000 FAQ / question pairs	

*Performance varies based on your specific use of synonyms and configurations on your index. It's best to test search performance to obtain more accurate benchmarks for your specific use case.

If your thesaurus is large, has a high term expansion ratio, and your latency increase is not within acceptable boundaries, you can try one or both of the following:

- Trim your thesaurus to reduce the expansion ratio (number of synonyms per term).
- Trim the overall coverage of terms (number of lines in your thesaurus).

Alternatively, you can increase the provisioning capacity (virtual storage units) to offset the latency increase.

Topics

- [Creating a thesaurus file \(p. 169\)](#)
- [Adding a thesaurus to an index \(p. 170\)](#)
- [Updating a thesaurus \(p. 173\)](#)
- [Deleting a thesaurus \(p. 176\)](#)
- [Toggling highlights in search results \(p. 177\)](#)

Creating a thesaurus file

An Amazon Kendra thesaurus file is a UTF-8-encoded file containing a list of synonyms in the Solr synonym list format. Synonyms are case sensitive. The thesaurus file must be less than 5 MB.

There are two ways to specify synonym mappings:

- *Bidirectional synonyms* are specified as a comma-separated list of terms. If the token matches any of the terms, then all the terms in the list are substituted, which includes the original token.
- *Unidirectional synonyms* are specified as two comma-separated lists of terms with the symbol "`=>`" between them. If the token matches any word on the left, then the list on the right is substituted. Mapping is only from the left to the right.

The following example shows a thesaurus file with synonyms for the sample AWS documentation for Amazon Kendra. Each line contains a single synonym rule. A synonym does not do an exact match on special characters. For example, if you search for `dead-letter-queue`, Kendra matches documents with the phrase `dead letter queue`. Blank lines and comments are ignored.

```
# Lines starting with pound are comments and blank lines are ignored.
# Synonym relationships can be defined as unidirectional or bidirectional relationships.
```

```
# Unidirection relationships are represented by any term sequence
# on the left hand side (LHS) of ">" followed by synonyms on the right hand side (RHS)
CodeStar => AWS CodeStar
# This will map CodeStar to AWS CodeStar, but not vice-versa

# Multiple synonym relationships may be defined in one line as well by comma separation.
autoscaling group, ASG => Auto Scaling group, autoscaling
# The above is equivalent to:
# autoscaling group => Auto Scaling group, autoscaling
# ASG => Auto Scaling group, autoscaling

# Bi-directional synonyms are comma separated terms with no ">"
DNS, Route53, Route 53
# DNS, Route53, and Route 53 map to one another and are interchangeable at match time
# The above is equivalent to:
# DNS => Route53, Route 53
# Route53 => DNS, Route 53
# Route 53 => DNS, Route53

# Overlapping LHS terms will be merged
Beta => Alpha
Beta => Gamma
Beta, Delta
# is equivalent to:
# Beta => Alpha, Gamma, Delta
# Delta => Beta

# Synonym rule count is the total number of lines defining synonym relationships
# Term count is the total number of unique terms for all rules.
# This thesaurus has a synonym rule count of 6 and a term count of 18.
# Comments and blanks lines do not count.
```

This example has 6 rules and 18 terms. Each line contains a single synonym rule. A synonym does not do an exact match on special characters. For example, if you search for dead-letter-queue, Kendra will match documents matching dead letter queue. Blank lines and comments are ignored. Some rules are ignored. For example, `a => b` is a rule, but `a => a` is ignored and does not count as a rule. A synonym does not do an exact match on special characters. For example, if you search for dead-letter-queue, Amazon Kendra will match document containing dead letter queue (no hyphen). You can have a maximum of 10,000 synonym rules per thesaurus.

The term count is the number of unique terms in the theaurus file. This example has the following terms: AWS CodeStar, autoscaling group, asg, Auto Scaling group, autoscaling, DNS, Route53, Route 53, dns, route53, route 53, beta, Alpha, Gamma, Delta, and delta. You can have up to 10 synonyms per term.

For more information about Amazon Kendra quotas, see [Quotas for Amazon Kendra \(p. 260\)](#).

Adding a thesaurus to an index

The following procedures show how to add a thesaurus file containing synonyms to an index. It can take up to 30 minutes to see the effects of your updated thesaurus file. For more information about the thesaurus file, see [Creating a thesaurus file \(p. 169\)](#).

Console

To add a thesaurus

1. In the left navigation pane, under the index where you want to add a list of synonyms, your thesaurus, choose **Synonyms**.
2. On the **Synonym** page, choose **Add Thesaurus**.

3. In **Define thesaurus**, give your thesaurus a name and an optional description.
4. In **Thesaurus settings**, provide the Amazon S3 path to your thesaurus file. The file must be smaller than 5 MB.
5. For **IAM Role**, select a role or select **Create a new role** and specify a role name to create a new role. Amazon Kendra uses this role to access the Amazon S3 resource on your behalf. The IAM role has the prefix "AmazonKendra-".
6. Choose **Save** to save the configuration and add the thesaurus. Once the thesaurus is ingested, it is active and synonyms are highlighted in results. It can take up to 30 minutes to see the effects of your thesaurus file.

CLI

To add a thesaurus to an index with the AWS CLI, call `create-thesaurus`:

```
aws kendra create-thesaurus \
--index-id index-id \
--name "thesaurus-name" \
--description "thesaurus-description" \
--source-s3-path "Bucket=bucket-name,Key=thesaurus/synonyms.txt" \
--role-arn role-arn
```

Call `list-thesauri` to see a list of thesauruses:

```
aws kendra list-thesauri \
--index-id index-id
```

To view details for a thesaurus, call `describe-thesaurus`:

```
aws kendra describe-thesaurus \
--index-id index-id \
--index-id thesaurus-id
```

It can take up to 30 minutes to see the effects of your thesaurus file.

Python

```
import boto3
from botocore.exceptions import ClientError
import pprint
import time

kendra = boto3.client("kendra")

print("Create a thesaurus")

thesaurus_name = "thesaurus-name"
thesaurus_description = "thesaurus-description"
thesaurus_role_arn = "role-arn"

index_id = "index-id"

s3_bucket_name = "bucket-name"
s3_key = "thesaurus-file"
source_s3_path= {
    'Bucket': s3_bucket_name,
    'Key': s3_key
}

try:
```

```

thesaurus_response = kendra.create_thesaurus(
    Description = thesaurus_description,
    Name = thesaurus_name,
    RoleArn = thesaurus_role_arn,
    IndexId = index_id,
    SourceS3Path = source_s3_path
)

pprint.pprint(thesaurus_response)

thesaurus_id = thesaurus_response["Id"]

print("Wait for Kendra to create the thesaurus.")

while True:
    # Get thesaurus description
    thesaurus_description = kendra.describe_thesaurus(
        Id = thesaurus_id,
        IndexId = index_id
    )
    # If status is not CREATING quit
    status = thesaurus_description["Status"]
    print("Creating thesaurus. Status: " + status)
    if status != "CREATING":
        break
    time.sleep(60)

except ClientError as e:
    print("%s" % e)

print("Program ends.")

```

Java

```

package com.amazonaws.kendra;

import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.CreateThesaurusRequest;
import software.amazon.awssdk.services.kendra.model.CreateThesaurusResponse;
import software.amazon.awssdk.services.kendra.model.DescribeThesaurusRequest;
import software.amazon.awssdk.services.kendra.model.DescribeThesaurusResponse;
import software.amazon.awssdk.services.kendra.model.S3Path;
import software.amazon.awssdk.services.kendra.model.ThesaurusStatus;

public class CreateThesaurusExample {

    public static void main(String[] args) throws InterruptedException {

        KendraClient kendra = KendraClient.builder().build();

        String thesaurusName = "thesaurus-name";
        String thesaurusDescription = "thesaurus-description";
        String thesaurusRoleArn = "role-arn";

        String s3BucketName = "bucket-name";
        String s3Key = "thesaurus-file";
        String indexId = "index-id";

        System.out.println(String.format("Creating a thesaurus named %s", thesaurusName));
        CreateThesaurusRequest createThesaurusRequest = CreateThesaurusRequest
            .builder()
            .name(thesaurusName)
            .indexId(indexId)
            .description(thesaurusDescription)
            .roleArn(thesaurusRoleArn)
    }
}

```

```
.sourceS3Path(S3Path.builder()
    .bucket(s3BucketName)
    .key(s3Key)
    .build())
    .build();
CreateThesaurusResponse createThesaurusResponse =
kendra.createThesaurus(createThesaurusRequest);
System.out.println(String.format("Thesaurus response %s",
createThesaurusResponse));

String thesaurusId = createThesaurusResponse.id();

System.out.println(String.format("Waiting until the thesaurus with ID %s is
created.", thesaurusId));

while (true) {
    DescribeThesaurusRequest describeThesaurusRequest =
DescribeThesaurusRequest.builder()
    .id(thesaurusId)
    .indexId(indexId)
    .build();
    DescribeThesaurusResponse describeThesaurusResponse =
kendra.describeThesaurus(describeThesaurusRequest);
    ThesaurusStatus status = describeThesaurusResponse.status();
    if (status != ThesaurusStatus.CREATING) {
        break;
    }
    TimeUnit.SECONDS.sleep(60);
}
System.out.println("Thesaurus creation is complete.");
}
```

Updating a thesaurus

You can change the configuration of a thesaurus after it is created. You can change details like thesaurus name and IAM information. You can also change the location of the thesaurus file Amazon S3 path. If you change the path to the thesaurus file, Amazon Kendra replaces the existing thesaurus with the thesaurus specified in the updated path.

It can take up to 30 minutes to see the effects of your updated thesaurus file.

Note

If there are validation or syntax errors in the thesaurus file, the previously uploaded thesaurus file is retained.

The following procedures show how to modify thesaurus details.

Console

To modify thesaurus details

1. In the left navigation pane, under the index you want to modify, choose **Synonyms**.
2. On the **Synonym** page, select the thesaurus you want to modify and then choose **Edit**.
3. On the **Update thesaurus** page, update the thesaurus details.
4. (Optional) Choose **Change the thesaurus file path** and then specify an Amazon S3 path to the new thesaurus file. Your existing thesaurus file is replaced by the file you specify. If you do not change the path, Amazon Kendra reloads the thesaurus from the existing path.

If you select **Keep the current thesaurus file**, Amazon Kendra does not reload the thesaurus file.

5. Choose **Save** to save the configuration.

You can also reload the thesaurus from the existing thesaurus path.

To reload a thesaurus from an existing path

1. In the left navigation pane, under the index you want to modify, choose **Synonyms**.
2. On the **Synonym** page, select the thesaurus you want to reload and then choose **Reload**.
3. On the **Reload thesaurus file** page, confirm you want to reload the thesaurus file.

CLI

To update a thesaurus, call `update-thesaurus`:

```
aws kendra update-thesaurus \
--index-id index-id \
--name "thesaurus-name" \
--description "thesaurus-description" \
--source-s3-path "Bucket=bucket-name,Key=thesaurus/synonyms.txt" \
--role-arn role-arn
```

Python

```
import boto3
from botocore.exceptions import ClientError
import pprint
import time

kendra = boto3.client("kendra")

print("Update a thesaurus")

thesaurus_name = "thesaurus-name"
thesaurus_description = "thesaurus-description"
thesaurus_role_arn = "role-arn"

thesaurus_id = "thesaurus-id"
index_id = "index-id"

s3_bucket_name = "bucket-name"
s3_key = "thesaurus-file"
source_s3_path= {
    'Bucket': s3_bucket_name,
    'Key': s3_key
}

try:
    kendra.update_thesaurus(
        Id = thesaurus_id,
        IndexId = index_id,
        Description = thesaurus_description,
        Name = thesaurus_name,
        RoleArn = thesaurus_role_arn,
        SourceS3Path = source_s3_path
    )
    print("Wait for Kendra to update the thesaurus.")

```

```

while True:
    # Get thesaurus description
    thesaurus_description = kendra.describe_thesaurus(
        Id = thesaurus_id,
        IndexId = index_id
    )
    # If status is not UPDATING quit
    status = thesaurus_description["Status"]
    print("Updating thesaurus. Status: " + status)
    if status != "UPDATING":
        break
    time.sleep(60)

except ClientError as e:
    print("%s" % e)

print("Program ends.")

```

Java

```

package com.amazonaws.kendra;

import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.UpdateThesaurusRequest;
import software.amazon.awssdk.services.kendra.model.DescribeThesaurusRequest;
import software.amazon.awssdk.services.kendra.model.DescribeThesaurusResponse;
import software.amazon.awssdk.services.kendra.model.S3Path;
import software.amazon.awssdk.services.kendra.model.ThesaurusStatus;

public class UpdateThesaurusExample {

    public static void main(String[] args) throws InterruptedException {

        KendraClient kendra = KendraClient.builder().build();

        String thesaurusName = "thesaurus-name";
        String thesaurusDescription = "thesaurus-description";
        String thesaurusRoleArn = "role-arn";

        String s3BucketName = "bucket-name";
        String s3Key = "thesaurus-file";

        String thesaurusId = "thesaurus-id";
        String indexId = "index-id";

        UpdateThesaurusRequest updateThesaurusRequest = UpdateThesaurusRequest
            .builder()
            .id(thesaurusId)
            .indexId(indexId)
            .name(thesaurusName)
            .description(thesaurusDescription)
            .roleArn(thesaurusRoleArn)
            .sourcesS3Path(S3Path.builder()
                .bucket(s3BucketName)
                .key(s3Key)
                .build())
            .build();
        kendra.updateThesaurus(updateThesaurusRequest);

        System.out.println(String.format("Waiting until the thesaurus with ID %s is
updated.", thesaurusId));

        // a new source s3 path requires re-consumption by Kendra
        // and so can take as long as a Create Thesaurus operation
        while (true) {

```

```
DescribeThesaurusRequest describeThesaurusRequest =
DescribeThesaurusRequest.builder()
    .id(thesaurusId)
    .indexId(indexId)
    .build();
DescribeThesaurusResponse describeThesaurusResponse =
kendra.describeThesaurus(describeThesaurusRequest);
ThesaurusStatus status = describeThesaurusResponse.status();
if (status != ThesaurusStatus.UPDATING) {
    break;
}

TimeUnit.SECONDS.sleep(60);
}

System.out.println("Thesaurus update is complete.");
}
}
```

Deleting a thesaurus

The following procedures show how to delete a thesaurus.

Console

1. In the left navigation pane, under the index you want to modify, choose **Synonyms**.
2. On the **Synonym** page, select the thesaurus you want to delete.
3. On the **Thesaurus detail** page, choose **Delete** and then confirm to delete.

CLI

To delete a thesarus to an index with the AWS CLI, call `delete-thesaurus`:

```
aws kendra create-thesaurus \
--index-id index-id \
--id thesaurus-id
```

Python

```
import boto3
from botocore.exceptions import ClientError

kendra = boto3.client("kendra")

print("Delete a thesaurus")

thesaurus_id = "thesaurus-id"
index_id = "index-id"

try:
    kendra.delete_thesaurus(
        Id = thesaurus_id,
        IndexId = index_id
    )

except ClientError as e:
    print("%s" % e)
```

```
    print("Program ends.")
```

Java

```
package com.amazonaws.kendra;

import software.amazon.awssdk.services.kendra.KendraClient;
import software.amazon.awssdk.services.kendra.model.DeleteThesaurusRequest;

public class DeleteThesaurusExample {

    public static void main(String[] args) throws InterruptedException {

        KendraClient kendra = KendraClient.builder().build();

        String thesaurusId = "thesaurus-id";
        String indexId = "index-id";

        DeleteThesaurusRequest updateThesaurusRequest = DeleteThesaurusRequest
            .builder()
            .id(thesaurusId)
            .indexId(indexId)
            .build();
        kendra.deleteThesaurus(updateThesaurusRequest);
    }
}
```

Toggling highlights in search results

Synonym highlighting is on by default. Highlight information is included in Amazon Kendra SDK and CLI query results. If you interact with Amazon Kendra using the SDK or CLI, you determine how to display results.

You can use the Amazon Kendra search console to test queries and see results. You can toggle synonym highlighting in the search console. The console tracks your preferences. The settings only apply to the search console session.

Synonym highlights will have the highlight type `THESAURUS_SYNONYM`. For more information about highlights, see the [Highlight \(p. 490\)](#) data type.

Tutorial: Building a metadata-enriched, intelligent search solution with Amazon Kendra

This tutorial shows you how to build a metadata-enriched, natural language based, intelligent search solution for your enterprise data using [Amazon Kendra](#), [Amazon Comprehend](#), [Amazon Simple Storage Service \(S3\)](#), and [AWS CloudShell](#).

Amazon Kendra is an intelligent search service that can build a search index for your unstructured, natural language data repositories. To make it easier for your customers to find and filter relevant answers, you can use Amazon Comprehend to extract metadata from your data and ingest it into your Amazon Kendra search index.

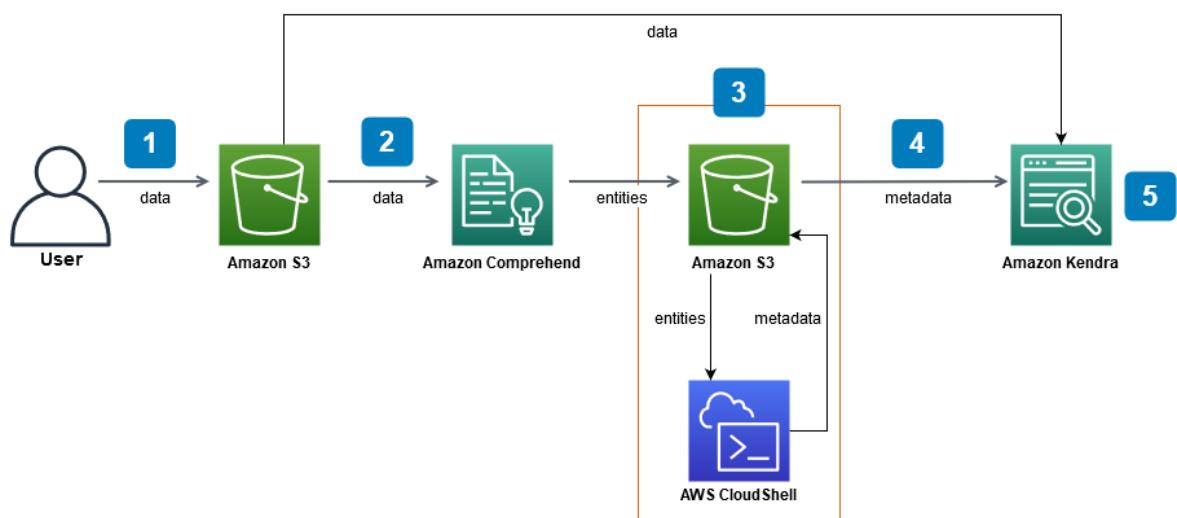
Amazon Comprehend is a natural language processing (NLP) service that can identify entities. Entities are references to people, places, locations, organizations, and objects in your data.

This tutorial uses a sample dataset of news articles to extract entities, convert them to metadata, and ingest them into your Amazon Kendra index to run searches on. The added metadata lets you filter your search results using any subset of these entities, and improves search accuracy. By following this tutorial, you will learn how to create a search solution for your enterprise data without any specialized machine learning knowledge.

This tutorial shows you how to build your search solution using the following steps:

1. Storing a sample dataset of news articles in Amazon S3.
2. Using Amazon Comprehend to extract entities from your data.
3. Running a Python 3 script to convert the entities into Amazon Kendra index metadata format and storing this metadata in S3.
4. Creating an Amazon Kendra search index and ingesting the data and the metadata.
5. Querying the search index.

The following diagram shows the workflow:



Estimated time to complete this tutorial: 1 hour

Estimated cost: Some of the actions in this tutorial incur charges on your AWS account. For more information on the cost of each service, see the price pages for [Amazon S3](#), [Amazon Comprehend](#), [AWS CloudShell](#), and [Amazon Kendra](#).

Topics

- [Prerequisites \(p. 179\)](#)
- [Step 1: Adding documents to Amazon S3 \(p. 179\)](#)
- [Step 2: Running an entities analysis job on Amazon Comprehend \(p. 186\)](#)
- [Step 3: Formatting the entities analysis output as Amazon Kendra metadata \(p. 192\)](#)
- [Step 4: Creating an Amazon Kendra index and ingesting the metadata \(p. 201\)](#)
- [Step 5: Querying the Amazon Kendra index \(p. 217\)](#)
- [Step 6: Cleaning up \(p. 224\)](#)

Prerequisites

To complete this tutorial, you need the following resources:

- An AWS account. If you do not have an AWS account, follow the steps in [Setting up Amazon Kendra](#) to set up your AWS account.
- A development computer running Windows, macOS, or Linux, to access the AWS Management Console. For more information, see [Configuring the AWS Management Console](#).
- An [AWS Identity and Access Management \(IAM\)](#) user. To learn how to set up an IAM user and group for your account, see the [Getting Started](#) section in the *IAM User Guide*.

If you are using the AWS Command Line Interface, you also need to attach the following policy to your IAM user to grant it the basic permissions required to complete this tutorial.

For more information, see [Creating IAM policies](#) and [Adding and removing IAM identity permissions](#).

- The [AWS Regional Services List](#). To reduce latency, you should choose the AWS region closest to your geographic location that is supported by both Amazon Comprehend and Amazon Kendra.
- (Optional) An [AWS Key Management Service](#). While this tutorial does not use encryption, you might want to use encryption best practices for your specific use case.
- (Optional) An [Amazon Virtual Private Cloud](#). While this tutorial does not use a VPC, you might want to use VPC best practices to ensure data security for your specific use case.

Step 1: Adding documents to Amazon S3

Before you run an Amazon Comprehend entities analysis job on your dataset, you create an Amazon S3 bucket to host the data, metadata, and the Amazon Comprehend entities analysis output.

Topics

- [Downloading the sample dataset \(p. 180\)](#)
- [Creating an Amazon S3 bucket \(p. 181\)](#)
- [Creating data and metadata folders in your S3 bucket \(p. 183\)](#)
- [Uploading the input data \(p. 185\)](#)

Downloading the sample dataset

Before Amazon Comprehend can run an entities analysis job on your data, you must download and extract the dataset and upload it to an S3 bucket.

To download and extract the dataset (Console)

1. Download the [tutorial-dataset.zip](#) folder on your device.
2. Extract the tutorial-dataset folder to access the data folder.

To download and extract the dataset (Terminal)

1. To download the tutorial-dataset, run the following command on a terminal window:

Linux

```
curl -o path/tutorial-dataset.zip https://docs.aws.amazon.com/kendra/latest/dg/samples/tutorial-dataset.zip
```

Where:

- *path/* is the local filepath to the location you want to save the zip folder in.

macOS

```
curl -o path/tutorial-dataset.zip https://docs.aws.amazon.com/kendra/latest/dg/samples/tutorial-dataset.zip
```

Where:

- *path/* is the local filepath to the location you want to save the zip folder in.

Windows

```
curl -o path/tutorial-dataset.zip https://docs.aws.amazon.com/kendra/latest/dg/samples/tutorial-dataset.zip
```

Where:

- *path/* is the local filepath to the location you want to save the zip folder in.

2. To extract the data from the zip folder, run the following command on the terminal window:

Linux

```
unzip path/tutorial-dataset.zip -d path/
```

Where:

- *path/* is the local filepath to your saved zip folder.

macOS

```
unzip path/tutorial-dataset.zip -d path/
```

Where:

- *path/* is the local filepath to your saved zip folder.

Windows

```
tar -xf path/tutorial-dataset.zip -C path/
```

Where:

- *path/* is the local filepath to your saved zip folder.

At the end of this step, you should have the extracted files in a decompressed folder called `tutorial-dataset`. This folder contains a `README` file with an Apache 2.0 open source attribution and a folder called `data` containing the dataset for this tutorial. The dataset consists of 100 files with `.story` extensions.

Creating an Amazon S3 bucket

After downloading and extracting the sample data folder, you store it in an Amazon S3 bucket.

Important

The name of an Amazon S3 bucket must be unique across all of AWS.

To create an S3 bucket (Console)

1. Sign in to the AWS Management Console and open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. In **Buckets**, choose **Create bucket**.
3. For **Bucket name**, enter a unique name.
4. For **Region**, choose the AWS region where you want to create the bucket.
Note
You must choose a region that supports both Amazon Comprehend and Amazon Kendra.
You cannot change the region of a bucket after you have created it.
5. Keep the default settings for **Block Public Access settings for this bucket**, **Bucket Versioning**, and **Tags**.
6. For **Default encryption**, choose **Disable**.
7. Keep the default settings for the **Advanced settings**.
8. Review your bucket configuration and then choose **Create bucket**.

To create an S3 bucket (AWS CLI)

1. To create an S3 bucket, use the `create-bucket` command in the AWS CLI:

Linux

```
aws s3api create-bucket \
```

```
--bucket DOC-EXAMPLE-BUCKET \
--region aws-region \
--create-bucket-configuration LocationConstraint=aws-region
```

Where:

- *DOC-EXAMPLE-BUCKET* is your bucket name,
- *aws-region* is the region you want to create your bucket in.

macOS

```
aws s3api create-bucket \
--bucket DOC-EXAMPLE-BUCKET \
--region aws-region \
--create-bucket-configuration LocationConstraint=aws-region
```

Where:

- *DOC-EXAMPLE-BUCKET* is your bucket name,
- *aws-region* is the region you want to create your bucket in.

Windows

```
aws s3api create-bucket ^
--bucket DOC-EXAMPLE-BUCKET ^
--region aws-region ^
--create-bucket-configuration LocationConstraint=aws-region
```

Where:

- *DOC-EXAMPLE-BUCKET* is your bucket name,
- *aws-region* is the region you want to create your bucket in.

Note

You must choose a region that supports both Amazon Comprehend and Amazon Kendra. You cannot change the region of a bucket after you have created it.

2. To ensure that your bucket was created successfully, use the [list](#) command:

Linux

```
aws s3 ls
```

macOS

```
aws s3 ls
```

Windows

```
aws s3 ls
```

Creating data and metadata folders in your S3 bucket

After creating your S3 bucket, you create data and metadata folders inside it.

To create folders in your S3 bucket (Console)

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. In **Buckets**, click on the name of your bucket from the list of buckets.
3. From the **Objects** tab, choose **Create folder**.
4. For the new folder name, enter **data**.
5. For the encryption settings, choose **Disable**.
6. Choose **Create folder**.
7. Repeat steps 3 to 6 to create another folder for storing the Amazon Kendra metadata and name the folder created in step 4 **metadata**.

To create folders in your S3 bucket (AWS CLI)

1. To create the data folder in your S3 bucket, use the [put-object](#) command in the AWS CLI:

Linux

```
aws s3api put-object \
    --bucket DOC-EXAMPLE-BUCKET \
    --key data/
```

Where:

- *DOC-EXAMPLE-BUCKET* is your bucket name.

macOS

```
aws s3api put-object \
    --bucket DOC-EXAMPLE-BUCKET \
    --key data/
```

Where:

- *DOC-EXAMPLE-BUCKET* is your bucket name.

Windows

```
aws s3api put-object ^
    --bucket DOC-EXAMPLE-BUCKET ^
    --key data/
```

Where:

- *DOC-EXAMPLE-BUCKET* is your bucket name.

2. To create the **metadata** folder in your S3 bucket, use the [put-object](#) command in the AWS CLI:

Linux

```
aws s3api put-object \  
    --bucket DOC-EXAMPLE-BUCKET \  
    --key metadata/
```

Where:

- **DOC-EXAMPLE-BUCKET** is your bucket name.

macOS

```
aws s3api put-object \  
    --bucket DOC-EXAMPLE-BUCKET \  
    --key metadata/
```

Where:

- **DOC-EXAMPLE-BUCKET** is your bucket name.

Windows

```
aws s3api put-object ^  
    --bucket DOC-EXAMPLE-BUCKET ^  
    --key metadata/
```

Where:

- **DOC-EXAMPLE-BUCKET** is your bucket name.

3. To ensure that your folders were created successfully, check the contents of your bucket using the [list](#) command:

Linux

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET/
```

Where:

- **DOC-EXAMPLE-BUCKET** is your bucket name.

macOS

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET/
```

Where:

- **DOC-EXAMPLE-BUCKET** is your bucket name.

Windows

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET/
```

Where:

- *DOC-EXAMPLE-BUCKET* is your bucket name.

Uploading the input data

After creating your data and metadata folders, you upload the sample dataset into the data folder.

To upload the sample dataset into the data folder (Console)

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. In **Buckets**, click on the name of your bucket from the list of buckets and then click on **Data**.
3. Choose **Upload** and then choose **Add files**.
4. In the dialog box, navigate to the data folder inside the `tutorial-dataset` folder in your local device, select all the files, and then choose **Open**.
5. Keep the default settings for **Destination**, **Permissions**, and **Properties**.
6. Choose **Upload**.

To upload the sample dataset into the data folder (AWS CLI)

1. To upload the sample data into the data folder, use the `copy` command in the AWS CLI:

Linux

```
aws s3 cp path/tutorial-dataset/data s3://DOC-EXAMPLE-BUCKET/data/ --recursive
```

Where:

- *path/* is the filepath to the `tutorial-dataset` folder on your device,
- *DOC-EXAMPLE-BUCKET* is your bucket name.

macOS

```
aws s3 cp path/tutorial-dataset/data s3://DOC-EXAMPLE-BUCKET/data/ --recursive
```

Where:

- *path/* is the filepath to the `tutorial-dataset` folder on your device,
- *DOC-EXAMPLE-BUCKET* is your bucket name.

Windows

```
aws s3 cp path/tutorial-dataset/data s3://DOC-EXAMPLE-BUCKET/data/ --recursive
```

Where:

- *path/* is the filepath to the `tutorial-dataset` folder on your device,
- *DOC-EXAMPLE-BUCKET* is your bucket name.

2. To ensure that your dataset files were uploaded successfully to your data folder, use the `list` command in the AWS CLI:

Linux

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET/data/
```

Where:

- *DOC-EXAMPLE-BUCKET* is the name of your S3 bucket.

macOS

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET/data/
```

Where:

- *DOC-EXAMPLE-BUCKET* is the name of your S3 bucket.

Windows

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET/data/
```

Where:

- *DOC-EXAMPLE-BUCKET* is the name of your S3 bucket.

At the end of this step, you have an S3 bucket with your dataset stored inside the data folder, and an empty `metadata` folder, which will store your Amazon Kendra metadata.

Step 2: Running an entities analysis job on Amazon Comprehend

After storing the sample dataset in your S3 bucket, you run an Amazon Comprehend entities analysis job to extract entities from your documents. These entities will form Amazon Kendra custom attributes and help you filter search results on your index. For more information, see [Detect Entities](#).

Topics

- [Running an Amazon Comprehend entities analysis job \(p. 186\)](#)

Running an Amazon Comprehend entities analysis job

To extract entities from your dataset, you run an Amazon Comprehend entities analysis job.

If you are using the AWS CLI in this step, you first create and attach an AWS IAM role and policy for Amazon Comprehend and then run an entities analysis job. To run an entities analysis job on your sample data, Amazon Comprehend needs:

- an AWS Identity and Access Management (IAM) role that recognizes it as a trusted entity
- an AWS IAM policy attached to the IAM role that gives it permissions to access your S3 bucket

For more information, see [Overview of managing access permissions to Amazon Comprehend resources](#) and [Using Identity-Based Policies \(IAM Policies\) for Amazon Comprehend](#).

To run an Amazon Comprehend entities analysis job (Console)

1. Open the Amazon Comprehend console at <https://console.aws.amazon.com/comprehend/>.

Important

Ensure that you are in the same region in which you created your Amazon S3 bucket. If you are in another region, choose the AWS region where you created your S3 bucket from the **Region selector** in the top navigation bar.

2. Choose **Launch Amazon Comprehend**.
3. In the left navigation pane, choose **Analysis jobs**.
4. Choose **Create job**.
5. In the **Job settings** section, do the following:
 - a. For **Name**, enter **data-entities-analysis**.
 - b. For **Analysis type**, choose **Entities**.
 - c. For **Language**, choose **English**.
 - d. Keep **Job encryption** turned off.
6. In the **Input data** section, do the following:
 - a. For **Data source**, choose **My documents**.
 - b. For **S3 location**, choose **Browse S3**.
 - c. For **Choose resources**, click on the name of your bucket from the list of buckets.
 - d. For **Objects**, select the option button for data and choose **Choose**.
 - e. For **Input format**, choose **One document per file**.
7. In the **Output data** section, do the following:
 - a. For **S3 location**, choose **Browse S3** and then select the option box for your bucket from the list of buckets and choose **Choose**.
 - b. Keep **Encryption** turned off.
8. In the **Access permissions** section, do the following:
 - a. For **IAM role**, choose **Create an IAM role**.
 - b. For **Permissions to access**, choose **Input and Output S3 buckets**.
 - c. For **Name suffix**, enter **comprehend-role**. This role provides access to your Amazon S3 bucket.
9. Keep the default **VPC settings**.
10. Choose **Create job**.

To run an Amazon Comprehend entities analysis job (AWS CLI)

1. To create and attach an IAM role for Amazon Comprehend that recognizes it as a trusted entity, do the following:
 - a. Save the following trust policy as a JSON file called **comprehend-trust-policy.json** in a text editor on your local device.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": "comprehend.amazonaws.com",  
            "Action": "sts:AssumeRole"  
        }  
    ]  
}
```

```
    "Principal": {
        "Service": "comprehend.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
}
]
```

- b. To create an IAM role called `comprehend-role` and attach your saved `comprehend-trust-policy.json` file to it, use the `create-role` command:

Linux

```
aws iam create-role \
    --role-name comprehend-role \
    --assume-role-policy-document file://path/comprehend-trust-
policy.json
```

Where:

- *path/* is the filepath to `comprehend-trust-policy.json` on your local device.

macOS

```
aws iam create-role \
    --role-name comprehend-role \
    --assume-role-policy-document file://path/comprehend-trust-
policy.json
```

Where:

- *path/* is the filepath to `comprehend-trust-policy.json` on your local device.

Windows

```
aws iam create-role ^
    --role-name comprehend-role ^
    --assume-role-policy-document file://path/comprehend-trust-
policy.json
```

Where:

- *path/* is the filepath to `comprehend-trust-policy.json` on your local device.

- c. Copy the Amazon Resource Name (ARN) to your text editor and save it locally as `comprehend-role-arn`.

Note

The ARN has a format similar to `arn:aws:iam::123456789012:role/comprehend-role`. You need the ARN you saved as `comprehend-role-arn` to run the Amazon Comprehend analysis job.

2. To create and attach an IAM policy to your IAM role that grants it permissions to access your S3 bucket, do the following:
- Save the following trust policy as a JSON file called `comprehend-S3-access-policy.json` in a text editor on your local device.

```
{
```

```
"Version": "2012-10-17",
"Statement": [
    {
        "Action": [
            "s3:GetObject"
        ],
        "Resource": [
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
        ],
        "Effect": "Allow"
    },
    {
        "Action": [
            "s3>ListBucket"
        ],
        "Resource": [
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
        ],
        "Effect": "Allow"
    },
    {
        "Action": [
            "s3:PutObject"
        ],
        "Resource": [
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
        ],
        "Effect": "Allow"
    }
]
```

- b. To create an IAM policy called `comprehend-S3-access-policy` to access your S3 bucket, use the [create-policy](#) command:

Linux

```
aws iam create-policy \
    --policy-name comprehend-S3-access-policy \
    --policy-document file://path/comprehend-S3-access-policy.json
```

Where:

- *path/* is the filepath to `comprehend-S3-access-policy.json` on your local device.

macOS

```
aws iam create-policy \
    --policy-name comprehend-S3-access-policy \
    --policy-document file://path/comprehend-S3-access-policy.json
```

Where:

- *path/* is the filepath to `comprehend-S3-access-policy.json` on your local device.

Windows

```
aws iam create-policy ^
    --policy-name comprehend-S3-access-policy ^
```

```
--policy-document file://path/comprehend-S3-access-policy.json
```

Where:

- *path*/ is the filepath to comprehend-S3-access-policy.json on your local device.

- Copy the Amazon Resource Name (ARN) to your text editor and save it locally as comprehend-S3-access-arn.

Note

The ARN has a format similar to `arn:aws:iam::123456789012:role/comprehend-S3-access-policy`. You need the ARN you saved as comprehend-S3-access-arn to attach the comprehend-S3-access-policy to your IAM role.

- To attach the comprehend-S3-access-policy to your IAM role, use the [attach-role-policy](#) command:

Linux

```
aws iam attach-role-policy \
    --policy-arn policy-arn \
    --role-name comprehend-role
```

Where:

- *policy-arn* is the ARN you saved as comprehend-S3-access-arn.

macOS

```
aws iam attach-role-policy \
    --policy-arn policy-arn \
    --role-name comprehend-role
```

Where:

- *policy-arn* is the ARN you saved as comprehend-S3-access-arn.

Windows

```
aws iam attach-role-policy ^
    --policy-arn policy-arn ^
    --role-name comprehend-role
```

Where:

- *policy-arn* is the ARN you saved as comprehend-S3-access-arn.

- To run an Amazon Comprehend entities analysis job, use the [start-entities-detection-job](#) command:

Linux

```
aws comprehend start-entities-detection-job \
    --input-data-config S3Uri=s3://DOC-EXAMPLE-BUCKET/
data/, InputFormat=ONE_DOC_PER_FILE \
    --output-data-config S3Uri=s3://DOC-EXAMPLE-BUCKET/ \
    --data-access-role-arn role-arn \
    --job-name data-entities-analysis \
    --language-code en \
```

```
--region aws-region
```

Where:

- *DOC-EXAMPLE-BUCKET* is the name of your S3 bucket,
- *role-arn* is the ARN you saved as `comprehend-role-arn`,
- *aws-region* is your AWS region.

macOS

```
aws comprehend start-entities-detection-job \
    --input-data-config S3Uri=s3://DOC-EXAMPLE-BUCKET/
data/, InputFormat=ONE_DOC_PER_FILE \
    --output-data-config S3Uri=s3://DOC-EXAMPLE-BUCKET/ \
    --data-access-role-arn role-arn \
    --job-name data-entities-analysis \
    --language-code en \
    --region aws-region
```

Where:

- *DOC-EXAMPLE-BUCKET* is the name of your S3 bucket,
- *role-arn* is the ARN you saved as `comprehend-role-arn`,
- *aws-region* is your AWS region.

Windows

```
aws comprehend start-entities-detection-job ^
    --input-data-config S3Uri=s3://DOC-EXAMPLE-BUCKET/
data/, InputFormat=ONE_DOC_PER_FILE ^
    --output-data-config S3Uri=s3://DOC-EXAMPLE-BUCKET/ ^
    --data-access-role-arn role-arn ^
    --job-name data-entities-analysis ^
    --language-code en ^
    --region aws-region
```

Where:

- *DOC-EXAMPLE-BUCKET* is the name of your S3 bucket,
- *role-arn* is the ARN you saved as `comprehend-role-arn`,
- *aws-region* is your AWS region.

4. Copy the entities analysis JobId and save it in a text editor as `comprehend-job-id`. The JobId helps you track the status of your entities analysis job.
5. To track the progress of your entities analysis job, use the `describe-entities-detection-job` command:

Linux

```
aws comprehend describe-entities-detection-job \
    --job-id entities-job-id \
    --region aws-region
```

Where:

- *entities-job-id* is your saved `comprehend-job-id`,

- `aws-region` is your AWS region.

macOS

```
aws comprehend describe-entities-detection-job \
    --job-id entities-job-id \
    --region aws-region
```

Where:

- `entities-job-id` is your saved `comprehend-job-id`,
- `aws-region` is your AWS region.

Windows

```
aws comprehend describe-entities-detection-job ^
    --job-id entities-job-id ^
    --region aws-region
```

Where:

- `entities-job-id` is your saved `comprehend-job-id`,
- `aws-region` is your AWS region.

It can take several minutes for the `JobStatus` to change to `COMPLETED`.

At the end of this step, Amazon Comprehend stores the entity analysis results as a zipped `output.tar.gz` file inside an `output` folder within an auto-generated folder in your S3 bucket. Make sure that your analysis job status is complete before you move on to the next step.

Step 3: Formatting the entities analysis output as Amazon Kendra metadata

To convert the entities extracted by Amazon Comprehend to the metadata format required by an Amazon Kendra index, you run a Python 3 script. The results of the conversion are stored in the `metadata` folder in your Amazon S3 bucket.

For more information on Amazon Kendra metadata format and structure, see [S3 document metadata](#).

Topics

- [Downloading and extracting the Amazon Comprehend output \(p. 193\)](#)
- [Uploading the output into the S3 bucket \(p. 195\)](#)
- [Converting the output to Amazon Kendra metadata format \(p. 196\)](#)
- [Cleaning up your Amazon S3 bucket \(p. 199\)](#)

Downloading and extracting the Amazon Comprehend output

To format the Amazon Comprehend entities analysis output, you must first download the Amazon Comprehend entities analysis output `.tar.gz` archive and extract the entities analysis file.

To download and extract the output file (Console)

1. In the Amazon Comprehend console navigation pane, navigate to **Analysis jobs**.
2. Choose your entities analysis job `data-entities-analysis`.
3. Under **Output**, choose the link displayed next to **Output data location**. This redirects you to the `.tar.gz` archive in your S3 bucket.
4. In the **Overview** tab, choose **Download**.

Tip

The output of all Amazon Comprehend analysis jobs have the same name. Renaming your archive will help you track it more easily.

5. Decompress and extract the downloaded Amazon Comprehend file to your device.

To download and extract the output file (AWS CLI)

1. To access the name of the Amazon Comprehend auto-generated folder in your S3 bucket which contains the results of the entities analysis job, use the `describe-entities-detection-job` command:

Linux

```
aws comprehend describe-entities-detection-job \
    --job-id entities-job-id \
    --region aws-region
```

Where:

- `entities-job-id` is your saved `comprehend-job-id` from the section called “Step 2: Detecting entities” (p. 186),
- `aws-region` is your AWS region.

macOS

```
aws comprehend describe-entities-detection-job \
    --job-id entities-job-id \
    --region aws-region
```

Where:

- `entities-job-id` is your saved `comprehend-job-id` from the section called “Step 2: Detecting entities” (p. 186),
- `aws-region` is your AWS region.

Windows

```
aws comprehend describe-entities-detection-job ^
    --job-id entities-job-id ^
```

```
--region aws-region
```

Where:

- *entities-job-id* is your saved comprehend-job-id from the section called “Step 2: Detecting entities” (p. 186),
- *aws-region* is your AWS region.

2. From the OutputDataConfig object in your entities job description, copy and save the S3Uri value as comprehend-S3uri on a text editor.

Note

The S3Uri value has a format similar to *s3://DOC-EXAMPLE-BUCKET/.../output/output.tar.gz*.

3. To download the entities output archive, use the `copy` command:

Linux

```
aws s3 cp s3://DOC-EXAMPLE-BUCKET/.../output/output.tar.gz path/output.tar.gz
```

Where:

- *s3://DOC-EXAMPLE-BUCKET/.../output/output.tar.gz* is the S3Uri value you saved as comprehend-S3uri,
- *path/* is the local directory where you wish to save the output.

macOS

```
aws s3 cp s3://DOC-EXAMPLE-BUCKET/.../output/output.tar.gz path/output.tar.gz
```

Where:

- *s3://DOC-EXAMPLE-BUCKET/.../output/output.tar.gz* is the S3Uri value you saved as comprehend-S3uri,
- *path/* is the local directory where you wish to save the output.

Windows

```
aws s3 cp s3://DOC-EXAMPLE-BUCKET/.../output/output.tar.gz path/output.tar.gz
```

Where:

- *s3://DOC-EXAMPLE-BUCKET/.../output/output.tar.gz* is the S3Uri value you saved as comprehend-S3uri,
- *path/* is the local directory where you wish to save the output.

4. To extract the entities output, run the following command on a terminal window:

Linux

```
tar -xf path/output.tar.gz -C path/
```

Where:

- *path/* is the filepath to the downloaded `output.tar.gz` archive on your local device.

macOS

```
tar -xf path/output.tar.gz -C path/
```

Where:

- *path*/ is the filepath to the downloaded `output.tar.gz` archive on your local device.

Windows

```
tar -xf path/output.tar.gz -C path/
```

Where:

- *path*/ is the filepath to the downloaded `output.tar.gz` archive on your local device.

At the end of this step, you should have a file on your device called `output` with a list of Amazon Comprehend identified entities.

Uploading the output into the S3 bucket

After downloading and extracting the Amazon Comprehend entities analysis file, you upload the extracted `output` file to your Amazon S3 bucket.

To upload the extracted Amazon Comprehend output file (Console)

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. In **Buckets**, click on the name of your bucket and then choose **Upload**.
3. In **Files and folders**, choose **Add files**.
4. In the dialog box, navigate to your extracted `output` file in your device, select it, and choose **Open**.
5. Keep the default settings for **Destination**, **Permissions**, and **Properties**.
6. Choose **Upload**.

To upload the extracted Amazon Comprehend output file (AWS CLI)

1. To upload the extracted `output` file to your bucket, use the `copy` command:

Linux

```
aws s3 cp path/output s3://DOC-EXAMPLE-BUCKET/output
```

Where:

- *path*/ is the local filepath to your extracted `output` file,
- *DOC-EXAMPLE-BUCKET* is the name of your S3 bucket.

macOS

```
aws s3 cp path/output s3://DOC-EXAMPLE-BUCKET/output
```

Where:

- *path/* is the local filepath to your extracted output file,
- *DOC-EXAMPLE-BUCKET* is the name of your S3 bucket.

Windows

```
aws s3 cp path/output s3://DOC-EXAMPLE-BUCKET/output
```

Where:

- *path/* is the local filepath to your extracted output file,
- *DOC-EXAMPLE-BUCKET* is the name of your S3 bucket.

2. To ensure that the output file was uploaded successfully to your S3 bucket, check its contents by using the [list](#) command:

Linux

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET/
```

Where:

- *DOC-EXAMPLE-BUCKET* is the name of your S3 bucket.

macOS

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET/
```

Where:

- *DOC-EXAMPLE-BUCKET* is the name of your S3 bucket.

Windows

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET/
```

Where:

- *DOC-EXAMPLE-BUCKET* is the name of your S3 bucket.

Converting the output to Amazon Kendra metadata format

To convert the Amazon Comprehend output to Amazon Kendra metadata, you run a Python 3 script. If you are using the Console, you use AWS CloudShell for this step.

To run the Python 3 script (Console)

1. Download the [converter.py.zip](#) zipped file on your device.
2. Extract the Python 3 file `converter.py`.

3. Sign into the [AWS Management Console](#) and make sure your AWS region is set to the same region as your S3 bucket and your Amazon Comprehend analysis job.
4. Choose the **AWS CloudShell icon** or type **AWS CloudShell** in the **Search** box on the top navigation bar to launch an environment.

Note

When AWS CloudShell launches in a new browser window for the first time, a welcome panel displays and lists key features. The shell is ready for interaction after you close this panel and the command prompt displays.

5. After the terminal is prepared, choose **Actions** from the navigation pane and then choose **Upload file** from the menu.
6. In the dialog box that opens, choose **Select file** and then choose the downloaded Python 3 file `converter.py` from your device. Choose **Upload**.
7. In the AWS CloudShell environment, enter the following command:

```
python3 converter.py
```

8. When the shell interface prompts you to **Enter the name of your S3 bucket**, enter the name of your S3 bucket and press enter.
9. When the shell interface prompts you to **Enter the full filepath to your Comprehend output file**, enter `output` and press enter.
10. When the shell interface prompts you to **Enter the full filepath to your metadata folder**, enter `metadata/` and press enter.

Important

For the metadata to be formatted correctly, the input values in steps 8-10 must be exact.

To run the Python 3 script (AWS CLI)

1. To download the Python 3 file `converter.py`, run the following command on a terminal window:

Linux

```
curl -o path/converter.py.zip https://docs.aws.amazon.com/kendra/latest/dg/samples/converter.py.zip
```

Where:

- *path/* is the filepath to the location you want to save the zipped file in.

macOS

```
curl -o path/converter.py.zip https://docs.aws.amazon.com/kendra/latest/dg/samples/converter.py.zip
```

Where:

- *path/* is the filepath to the location you want to save the zipped file in.

Windows

```
curl -o path/converter.py.zip https://docs.aws.amazon.com/kendra/latest/dg/samples/converter.py.zip
```

Where:

- *path/* is the filepath to the location you want to save the zipped file in.

2. To extract the Python 3 file, run the following command on the terminal window:

Linux

```
unzip path/converter.py.zip -d path/
```

Where:

- *path/* is the filepath to your saved converter.py.zip.

macOS

```
unzip path/converter.py.zip -d path/
```

Where:

- *path/* is the filepath to your saved converter.py.zip.

Windows

```
tar -xf path/converter.py.zip -C path/
```

Where:

- *path/* is the filepath to your saved converter.py.zip.

3. Make sure that Boto3 is installed on your device by running the following command.

Linux

```
pip3 show boto3
```

macOS

```
pip3 show boto3
```

Windows

```
pip3 show boto3
```

Note

If you do not have Boto3 installed, run `pip3 install boto3` to install it.

4. To run the Python 3 script to convert the output file, run the following command.

Linux

```
python path/converter.py
```

Where:

- *path/* is the filepath to your saved converter.py.zip.

macOS

```
python path/converter.py
```

Where:

- *path/* is the filepath to your saved converter.py.zip.

Windows

```
python path/converter.py
```

Where:

- *path/* is the filepath to your saved converter.py.zip.

5. When the AWS CLI prompts you to Enter the name of your S3 bucket, enter the name of your S3 bucket and press enter.
6. When the AWS CLI prompts you to Enter the full filepath to your Comprehend output file, enter **output** and press enter.
7. When the AWS CLI prompts you to Enter the full filepath to your metadata folder, enter **metadata/** and press enter.

Important

For the metadata to be formatted correctly, the input values in steps 5-7 must be exact.

At the end of this step, the formatted metadata is deposited inside the **metadata** folder in your S3 bucket.

Cleaning up your Amazon S3 bucket

Since the Amazon Kendra index syncs all files stored in a bucket, we recommend you clean up your Amazon S3 bucket to prevent redundant search results.

To clean up your Amazon S3 bucket (Console)

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. In **Buckets**, choose your bucket and then select the Amazon Comprehend entity analysis output folder, the Amazon Comprehend entity analysis .temp file, and the extracted Amazon Comprehend output file.
3. From the **Overview** tab choose **Delete**.
4. In **Delete objects**, choose **Permanently delete objects?** and enter **permanently delete** in the text input field.
5. Choose **Delete objects**.

To clean up your Amazon S3 bucket (AWS CLI)

1. To delete all files and folders in your S3 bucket except the data and metadata folders, use the [remove](#) command in the AWS CLI:

Linux

```
aws s3 rm s3://DOC-EXAMPLE-BUCKET/ --recursive --exclude "data/*" --exclude "metadata/*"
```

Where:

- *DOC-EXAMPLE-BUCKET* is the name of your S3 bucket.

macOS

```
aws s3 rm s3://DOC-EXAMPLE-BUCKET/ --recursive --exclude "data/*" --exclude "metadata/*"
```

Where:

- *DOC-EXAMPLE-BUCKET* is the name of your S3 bucket.

Windows

```
aws s3 rm s3://DOC-EXAMPLE-BUCKET/ --recursive --exclude "data/*" --exclude "metadata/*"
```

Where:

- *DOC-EXAMPLE-BUCKET* is the name of your S3 bucket.

2. To ensure that the objects were successfully deleted from your S3 bucket, check its contents by using the [list](#) command:

Linux

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET/
```

Where:

- *DOC-EXAMPLE-BUCKET* is the name of your S3 bucket.

macOS

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET/
```

Where:

- *DOC-EXAMPLE-BUCKET* is the name of your S3 bucket.

Windows

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET/
```

Where:

- *DOC-EXAMPLE-BUCKET* is the name of your S3 bucket.

At the end of this step, you have converted the Amazon Comprehend entities analysis output to Amazon Kendra metadata. You are now ready to create an Amazon Kendra index.

Step 4: Creating an Amazon Kendra index and ingesting the metadata

To implement your intelligent search solution, you create an Amazon Kendra index and ingest your S3 data and metadata into it.

Before you add metadata to your Amazon Kendra index, you create custom index fields corresponding to custom document attributes, which in turn correspond to the Amazon Comprehend entity types. Amazon Kendra uses the index fields and custom document attributes you create to search and filter your documents.

For more information, see [Index](#) and [Creating custom document attributes](#).

Topics

- [Creating an Amazon Kendra index \(p. 201\)](#)
- [Updating the IAM role for Amazon S3 access \(p. 207\)](#)
- [Creating Amazon Kendra custom search index fields \(p. 209\)](#)
- [Adding the Amazon S3 bucket as a data source for the index \(p. 213\)](#)
- [Syncing the Amazon Kendra index \(p. 215\)](#)

Creating an Amazon Kendra index

To query your source documents, you create an Amazon Kendra index.

If you are using the AWS CLI in this step, you create and attach an AWS IAM role and policy that allows Amazon Kendra to access your CloudWatch logs before creating an index. For more information, see [Prerequisites](#).

To create an Amazon Kendra index (Console)

1. Open the Amazon Kendra console at <https://console.aws.amazon.com/kendra/>.

Important

Ensure that you are in the same region in which you created your Amazon Comprehend entities analysis job and your Amazon S3 bucket. If you are in another region, choose the AWS region where you created your Amazon S3 bucket from the **Region selector** in the top navigation bar.

2. Choose **Create an index**.
3. For **Index details** on the **Specify index details** page, do the following:
 - a. For **Index name**, enter **kendra-index**.
 - b. Keep the **Description** field blank.
 - c. For **IAM role**, choose **Create a new role**. This role provides access to your Amazon S3 bucket.
 - d. For **Role name**, enter **kendra-role**. The IAM role will have the prefix **AmazonKendra-**.
 - e. Keep default settings for **Encryption** and **Tags** and choose **Next**.
4. For **Access control settings** on the **Configure user access control** page, choose **No** and then choose **Next**.
5. For **Provisioning editions** on the **Provisioning details** page, choose **Developer edition** and choose **Create**.

To create an Amazon Kendra index (AWS CLI)

1. To create and attach an IAM role for Amazon Kendra that recognizes it as a trusted entity, do the following:
 - a. Save the following trust policy as a JSON file called **kendra-trust-policy.json** in a text editor on your local device.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Principal": {  
            "Service": "kendra.amazonaws.com"  
        },  
        "Action": "sts:AssumeRole"  
    }  
}
```

- b. To create an IAM role called **kendra-role** and attach your saved **kendra-trust-policy.json** file to it, use the [create-role](#) command:

Linux

```
aws iam create-role \  
    --role-name kendra-role \  
    --assume-role-policy-document file://path/kendra-trust-policy.json
```

Where:

- *path* is the filepath to **kendra-trust-policy.json** on your local device.

macOS

```
aws iam create-role \  
    --role-name kendra-role \  
    --assume-role-policy-document file://path/kendra-trust-policy.json
```

Where:

- *path* is the filepath to **kendra-trust-policy.json** on your local device.

Windows

```
aws iam create-role ^
--role-name kendra-role ^
--assume-role-policy-document file://path/kendra-trust-policy.json
```

Where:

- *path*/ is the filepath to `kendra-trust-policy.json` on your local device.
- Copy the Amazon Resource Name (ARN) to your text editor and save it locally as `kendra-role-arn`.

Note

The ARN has a format similar to `arn:aws:iam::123456789012:role/kendra-role`. You need the ARN you saved as `kendra-role-arn` to run Amazon Kendra jobs.

- Before you create an index, you must provide your `kendra-role` the permission to write to CloudWatch Logs. To do this, complete the following steps:
 - Save the following trust policy as a JSON file called `kendra-cloudwatch-policy.json` in a text editor on your local device.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "cloudwatch:PutMetricData",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "Kendra"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "logs:DescribeLogGroups",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "logs>CreateLogGroup",
            "Resource": "arn:aws:logs:aws-region:aws-account-id:log-group:/aws/kendra/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogStreams",
                "logs>CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:aws-region:aws-account-id:log-group:/aws/kendra/*:log-stream:*"
        }
    ]
}
```

Replace `aws-region` with your AWS region, and `aws-account-id` with your 12-digit AWS account ID.

- b. To create an IAM policy to access CloudWatch Logs, use the [create-policy](#) command:

Linux

```
aws iam create-policy \  
    --policy-name kendra-cloudwatch-policy \  
    --policy-document file://path/kendra-cloudwatch-policy.json
```

Where:

- *path/* is the filepath to `kendra-cloudwatch-policy.json` on your local device.

macOS

```
aws iam create-policy \  
    --policy-name kendra-cloudwatch-policy \  
    --policy-document file://path/kendra-cloudwatch-policy.json
```

Where:

- *path/* is the filepath to `kendra-cloudwatch-policy.json` on your local device.

Windows

```
aws iam create-policy ^  
    --policy-name kendra-cloudwatch-policy ^  
    --policy-document file://path/kendra-cloudwatch-policy.json
```

Where:

- *path/* is the filepath to `kendra-cloudwatch-policy.json` on your local device.

- c. Copy the Amazon Resource Name (ARN) to your text editor and save it locally as `kendra-cloudwatch-arn`.

Note

The ARN has a format similar to `arn:aws:iam::123456789012:role/kendra-cloudwatch-policy`. You need the ARN you saved as `kendra-cloudwatch-arn` to attach the `kendra-cloudwatch-policy` to your IAM role.

- d. To attach the `kendra-cloudwatch-policy` to your IAM role, use the [attach-role-policy](#) command:

Linux

```
aws iam attach-role-policy \  
    --policy-arn policy-arn \  
    --role-name kendra-role
```

Where:

- *policy-arn* is your saved `kendra-cloudwatch-arn`.

macOS

```
aws iam attach-role-policy \
    --policy-arn policy-arn \
    --role-name kendra-role
```

Where:

- *policy-arn* is your saved kendra-cloudwatch-arn.

Windows

```
aws iam attach-role-policy ^
    --policy-arn policy-arn ^
    --role-name kendra-role
```

Where:

- *policy-arn* is your saved kendra-cloudwatch-arn.

3. To create an index, use the [create-index](#) command:

Linux

```
aws kendra create-index \
    --name kendra-index \
    --edition DEVELOPER_EDITION \
    --role-arn role-arn \
    --region aws-region
```

Where:

- *role-arn* is your saved kendra-role-arn,
- *aws-region* is your AWS region.

macOS

```
aws kendra create-index \
    --name kendra-index \
    --edition DEVELOPER_EDITION \
    --role-arn role-arn \
    --region aws-region
```

Where:

- *role-arn* is your saved kendra-role-arn,
- *aws-region* is your AWS region.

Windows

```
aws kendra create-index ^
    --name kendra-index ^
    --edition DEVELOPER_EDITION ^
    --role-arn role-arn ^
```

```
--region aws-region
```

Where:

- *role-arn* is your saved `kendra-role-arn`,
- *aws-region* is your AWS region.

4. Copy the index ID and save it in a text editor as `kendra-index-id`. The ID helps you track the status of your index creation.
5. To track the progress of your index creation job, use the [describe-index](#) command:

Linux

```
aws kendra describe-index \  
    --id kendra-index-id \  
    --region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *aws-region* is your AWS region.

macOS

```
aws kendra describe-index \  
    --id kendra-index-id \  
    --region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *aws-region* is your AWS region.

Windows

```
aws kendra describe-index ^  
    --id kendra-index-id ^  
    --region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *aws-region* is your AWS region.

The index creation process on average takes 15 minutes, but can take longer. When the status of the index is active, your index is ready to use. While your index is being created, you can start the next step.

If you are using the AWS CLI in this step, you create and attach an IAM policy to your Amazon Kendra IAM role that gives your index permissions to access your S3 bucket.

Updating the IAM role for Amazon S3 access

While the index is being created, you update your Amazon Kendra IAM role to allow the index you created to read data from your Amazon S3 bucket. For more information, see [IAM access roles for Amazon Kendra](#).

To update your IAM role (Console)

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the left navigation pane, choose **Roles** and enter **kendra-role** in the **Search** box above **Role name**.
3. From the suggested options, click on **kendra-role**.
4. In **Summary**, choose **Attach policies**.
5. In **Attach permissions**, in the **Search** box, enter **s3** and select the checkbox next to the **AmazonS3ReadOnlyAccess** policy from the suggested options.
6. Choose **Attach policy**. On the **Summary** page, you will now see two policies attached to the IAM role.
7. Return to the Amazon Kendra console at <https://console.aws.amazon.com/kendra/> and wait for the status of your index to change from **Creating** to **Active** before continuing to the next step.

To update your IAM role (AWS CLI)

1. Save the following text in a JSON file called **kendra-S3-access-policy.json** in a text editor on your local device.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": [  
                "s3:GetObject"  
            ],  
            "Resource": [  
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"  
            ],  
            "Effect": "Allow"  
        },  
        {  
            "Action": [  
                "s3>ListBucket"  
            ],  
            "Resource": [  
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET"  
            ],  
            "Effect": "Allow"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "kendra:BatchPutDocument",  
                "kendra:BatchDeleteDocument",  
                "kendra>ListDataSourceSyncJobs"  
            ],  
            "Resource": [  
                "arn:aws:kendra:aws-region:aws-account-id:index/kendra-index-id"  
            ]  
        }  
    ]  
}
```

}

Replace **DOC-EXAMPLE-BUCKET** with your S3 bucket name, **aws-region** with your AWS region, **aws-account-id** with your 12-digit AWS account ID, and **kendra-index-id** with your saved **kendra-index-id**.

2. To create an IAM policy to access your S3 bucket, use the [create-policy](#) command:

Linux

```
aws iam create-policy \
    --policy-name kendra-S3-access-policy \
    --policy-document file://path/kendra-S3-access-policy.json
```

Where:

- **path/** is the filepath to **kendra-S3-access-policy.json** on your local device.

macOS

```
aws iam create-policy \
    --policy-name kendra-S3-access-policy \
    --policy-document file://path/kendra-S3-access-policy.json
```

Where:

- **path/** is the filepath to **kendra-S3-access-policy.json** on your local device.

Windows

```
aws iam create-policy ^
    --policy-name kendra-S3-access-policy ^
    --policy-document file://path/kendra-S3-access-policy.json
```

Where:

- **path/** is the filepath to **kendra-S3-access-policy.json** on your local device.

3. Copy the Amazon Resource Name (ARN) to your text editor and save it locally as **kendra-S3-access-arn**.

Note

The ARN has a format similar to **arn:aws:iam::123456789012:role/kendra-S3-access-policy**. You need the ARN you saved as **kendra-S3-access-arn** to attach the **kendra-S3-access-policy** to your IAM role.

4. To attach the **kendra-S3-access-policy** to your Amazon Kendra IAM role, use the [attach-role-policy](#) command:

Linux

```
aws iam attach-role-policy \
    --policy-arn policy-arn \
    --role-name kendra-role
```

Where:

- **policy-arn** is your saved **kendra-S3-access-arn**.

macOS

```
aws iam attach-role-policy \
    --policy-arn policy-arn \
    --role-name kendra-role
```

Where:

- *policy-arn* is your saved kendra-S3-access-arn.

Windows

```
aws iam attach-role-policy ^
    --policy-arn policy-arn ^
    --role-name kendra-role
```

Where:

- *policy-arn* is your saved kendra-S3-access-arn.

Creating Amazon Kendra custom search index fields

To prepare Amazon Kendra to recognize your metadata as custom document attributes, you create custom fields corresponding to Amazon Comprehend entity types. You input the following nine Amazon Comprehend entity types as custom fields:

- COMMERCIAL_ITEM
- DATE
- EVENT
- LOCATION
- ORGANIZATION
- OTHER
- PERSON
- QUANTITY
- TITLE

Important

Misspelled entity types will not be recognized by the index.

To create custom fields for your Amazon Kendra index (Console)

1. Open the Amazon Kendra console at <https://console.aws.amazon.com/kendra/>.
2. From the **Indexes** list, click on **kendra-index**.
3. From the left navigation panel, under **Data management**, choose **Facet definition**.
4. From the **Index fields** menu, choose **Add field**.
5. In the **Add index field** dialog box, do the following:
 - a. In **Field name**, enter **COMMERCIAL_ITEM**.

- b. In **Data type**, choose **String list**.
- c. In **Usage types**, select **Facetable**, **Searchable**, and **Displayable**, and then choose **Add**.
- d. Repeat steps a to c for each Amazon Comprehend entity type: COMMERCIAL_ITEM, DATE, EVENT, LOCATION, ORGANIZATION, OTHER, PERSON, QUANTITY, TITLE.

The console displays successful field addition messages. You can choose to close them before you proceed with the next step.

To create custom fields for your Amazon Kendra index (AWS CLI)

1. Save the following text as a JSON file called `custom-attributes.json` in a text editor on your local device.

```
[  
  {  
    "Name": "COMMERCIAL_ITEM",  
    "Type": "STRING_LIST_VALUE",  
    "Search": {  
      "Facetable": true,  
      "Searchable": true,  
      "Displayable": true  
    }  
  },  
  {  
    "Name": "DATE",  
    "Type": "STRING_LIST_VALUE",  
    "Search": {  
      "Facetable": true,  
      "Searchable": true,  
      "Displayable": true  
    }  
  },  
  {  
    "Name": "EVENT",  
    "Type": "STRING_LIST_VALUE",  
    "Search": {  
      "Facetable": true,  
      "Searchable": true,  
      "Displayable": true  
    }  
  },  
  {  
    "Name": "LOCATION",  
    "Type": "STRING_LIST_VALUE",  
    "Search": {  
      "Facetable": true,  
      "Searchable": true,  
      "Displayable": true  
    }  
  },  
  {  
    "Name": "ORGANIZATION",  
    "Type": "STRING_LIST_VALUE",  
    "Search": {  
      "Facetable": true,  
      "Searchable": true,  
      "Displayable": true  
    }  
  },  
  {  
    "Name": "OTHER",  
    "Type": "STRING_LIST_VALUE",  
  }
```

```
        "Search": {
            "Facetable": true,
            "Searchable": true,
            "Displayable": true
        }
    },
{
    "Name": "PERSON",
    "Type": "STRING_LIST_VALUE",
    "Search": {
        "Facetable": true,
        "Searchable": true,
        "Displayable": true
    }
},
{
    "Name": "QUANTITY",
    "Type": "STRING_LIST_VALUE",
    "Search": {
        "Facetable": true,
        "Searchable": true,
        "Displayable": true
    }
},
{
    "Name": "TITLE",
    "Type": "STRING_LIST_VALUE",
    "Search": {
        "Facetable": true,
        "Searchable": true,
        "Displayable": true
    }
}
]
```

2. To create custom fields in your index, use the [update-index](#) command:

Linux

```
aws kendra update-index \
    --id kendra-index-id \
    --document-metadata-configuration-updates file://path/custom-
attributes.json \
    --region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *path/* is the filepath to `custom-attributes.json` on your local device,
- *aws-region* is your AWS region.

macOS

```
aws kendra update-index \
    --id kendra-index-id \
    --document-metadata-configuration-updates file://path/custom-
attributes.json \
    --region aws-region
```

Where:

- *kendra-index-id* is your saved kendra-index-id,
- *path/* is the filepath to `custom-attributes.json` on your local device,
- *aws-region* is your AWS region.

Windows

```
aws kendra update-index ^
--id kendra-index-id ^
--document-metadata-configuration-updates file://path/custom-
attributes.json ^
--region aws-region
```

Where:

- *kendra-index-id* is your saved kendra-index-id,
- *path/* is the filepath to `custom-attributes.json` on your local device,
- *aws-region* is your AWS region.

3. To verify that the custom attributes have been added to your index, use the [describe-index](#) command:

Linux

```
aws kendra describe-index \
--id kendra-index-id \
--region aws-region
```

Where:

- *kendra-index-id* is your saved kendra-index-id,
- *aws-region* is your AWS region.

macOS

```
aws kendra describe-index \
--id kendra-index-id \
--region aws-region
```

Where:

- *kendra-index-id* is your saved kendra-index-id,
- *aws-region* is your AWS region.

Windows

```
aws kendra describe-index ^
--id kendra-index-id ^
--region aws-region
```

Where:

- *kendra-index-id* is your saved kendra-index-id,
- *aws-region* is your AWS region.

Adding the Amazon S3 bucket as a data source for the index

Before you can sync your index, you must connect your S3 data source to it.

To connect an S3 bucket to your Amazon Kendra index (Console)

1. Open the Amazon Kendra console at <https://console.aws.amazon.com/kendra/>.
2. From the **Indexes** list, click on **kendra-index**.
3. From the left navigation menu, under **Data management**, choose **Data sources**.
4. Under the **Select data source connector type** section, navigate to **Amazon S3**, and choose **Add connector**.
5. In the **Specify data source details** page, do the following:
 - a. Under **Name and description**, for **Data source name**, enter **S3-data-source**.
 - b. Keep the **Description** section blank.
 - c. Keep the default settings for **Tags**.
 - d. Choose **Next**.
6. On the **Configure sync settings** page, in the **Sync scope** section, do the following:
 - a. In **Enter the data source location**, choose **Browse S3**.
 - b. In **Choose resources**, select your S3 bucket and then choose **Choose**.
 - c. In **Metadata files prefix folder location**, choose **Browse S3**.
 - d. In **Choose resources**, click on the name of your bucket from the list of buckets.
 - e. For **Objects**, select the option box for **metadata** and choose **Choose**. The location field should now say **metadata/**.
 - f. Keep the default settings for **Access control list configuration file location**, **Select decryption key**, and **Additional configuration**.
7. For **IAM role**, on the **Configure sync settings** page, choose **kendra-role**.
8. On the **Configure sync settings** page, under **Sync run schedule**, for **Frequency**, choose **Run on demand** and then choose **Next**.
9. On the **Review and create** page, review your choices for the data source details and choose **Add data source**.

To connect an S3 bucket to your Amazon Kendra index (AWS CLI)

1. Save the following text as a JSON file called **S3-data-connector.json** in a text editor on your local device.

```
{  
    "S3Configuration":{  
        "BucketName":"DOC-EXAMPLE-BUCKET",  
        "DocumentsMetadataConfiguration":{  
            "S3Prefix":"metadata"  
        }  
    }  
}
```

Replace **DOC-EXAMPLE-BUCKET** with the name of your S3 bucket.

2. To connect your S3 bucket to your index, use the [create-data-source](#) command:

Linux

```
aws kendra create-data-source \
--index-id kendra-index-id \
--name S3-data-source \
--type S3 \
--configuration file://path/S3-data-connector.json \
--role-arn role-arn \
--region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *path/* is the filepath to `S3-data-connector.json` on your local device,
- *role-arn* is your saved `kendra-role-arn`,
- *aws-region* is your AWS region.

macOS

```
aws kendra create-data-source \
--index-id kendra-index-id \
--name S3-data-source \
--type S3 \
--configuration file://path/S3-data-connector.json \
--role-arn role-arn \
--region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *path/* is the filepath to `S3-data-connector.json` on your local device,
- *role-arn* is your saved `kendra-role-arn`,
- *aws-region* is your AWS region.

Windows

```
aws kendra create-data-source ^
--index-id kendra-index-id ^
--name S3-data-source ^
--type S3 ^
--configuration file://path/S3-data-connector.json ^
--role-arn role-arn ^
--region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *path/* is the filepath to `S3-data-connector.json` on your local device,
- *role-arn* is your saved `kendra-role-arn`,
- *aws-region* is your AWS region.

3. Copy the connector `Id` and save it in a text editor as `S3-connector-id`. The `Id` helps you track the status of the data-connection process.

4. To ensure that your S3 data source was connected successfully, use the [describe-data-source](#) command:

Linux

```
aws kendra describe-data-source \
    --id S3-connector-id \
    --index-id kendra-index-id \
    --region aws-region
```

Where:

- *S3-connector-id* is your saved S3-connector-id,
- *kendra-index-id* is your saved kendra-index-id,
- *aws-region* is your AWS region.

macOS

```
aws kendra describe-data-source \
    --id S3-connector-id \
    --index-id kendra-index-id \
    --region aws-region
```

Where:

- *S3-connector-id* is your saved S3-connector-id,
- *kendra-index-id* is your saved kendra-index-id,
- *aws-region* is your AWS region.

Windows

```
aws kendra describe-data-source ^
    --id S3-connector-id ^
    --index-id kendra-index-id ^
    --region aws-region
```

Where:

- *S3-connector-id* is your saved S3-connector-id,
- *kendra-index-id* is your saved kendra-index-id,
- *aws-region* is your AWS region.

At the end of this step, your Amazon S3 data source is connected to the index.

Syncing the Amazon Kendra index

With the Amazon S3 data source added, you now sync your Amazon Kendra index to it.

To sync your Amazon Kendra index (Console)

1. Open the Amazon Kendra console at <https://console.aws.amazon.com/kendra/>.
2. From the **Indexes** list, click on **kendra-index**.
3. From the left navigation menu, choose **Data sources**.

4. From **Data sources**, select `S3-data-source`.
5. From the top navigation bar, choose **Sync now**.

To sync your Amazon Kendra index (AWS CLI)

1. To sync your index, use the `start-data-source-sync-job` command:

Linux

```
aws kendra start-data-source-sync-job \
    --id S3-connector-id \
    --index-id kendra-index-id \
    --region aws-region
```

Where:

- `S3-connector-id` is your saved `S3-connector-id`,
- `kendra-index-id` is your saved `kendra-index-id`,
- `aws-region` is your AWS region.

macOS

```
aws kendra start-data-source-sync-job \
    --id S3-connector-id \
    --index-id kendra-index-id \
    --region aws-region
```

Where:

- `S3-connector-id` is your saved `S3-connector-id`,
- `kendra-index-id` is your saved `kendra-index-id`,
- `aws-region` is your AWS region.

Windows

```
aws kendra start-data-source-sync-job ^
    --id S3-connector-id ^
    --index-id kendra-index-id ^
    --region aws-region
```

Where:

- `S3-connector-id` is your saved `S3-connector-id`,
- `kendra-index-id` is your saved `kendra-index-id`,
- `aws-region` is your AWS region.

2. To check the status of the index sync, use the `list-data-source-sync-jobs` command:

Linux

```
aws kendra list-data-source-sync-jobs \
    --id S3-connector-id \
    --index-id kendra-index-id \
    --region aws-region
```

Where:

- *S3-connector-id* is your saved S3-connector-id,
- *kendra-index-id* is your saved kendra-index-id,
- *aws-region* is your AWS region.

macOS

```
aws kendra list-data-source-sync-jobs \
    --id S3-connector-id \
    --index-id kendra-index-id \
    --region aws-region
```

Where:

- *S3-connector-id* is your saved S3-connector-id,
- *kendra-index-id* is your saved kendra-index-id,
- *aws-region* is your AWS region.

Windows

```
aws kendra list-data-source-sync-jobs ^
    --id S3-connector-id ^
    --index-id kendra-index-id ^
    --region aws-region
```

Where:

- *S3-connector-id* is your saved S3-connector-id,
- *kendra-index-id* is your saved kendra-index-id,
- *aws-region* is your AWS region.

At the end of this step, you have created a searchable and filterable Amazon Kendra index for your dataset.

Step 5: Querying the Amazon Kendra index

Your Amazon Kendra index is now ready for natural language queries. When you search your index, Amazon Kendra uses all the data and metadata you provided to return the most accurate answers to your search query.

There are three kinds of queries that Amazon Kendra can answer:

- Factoid queries ("who", "what", "when", or "where" questions)
- Descriptive queries ("how" questions)
- Keyword searches (questions whose intent and scope are not clear)

Topics

- [Querying your Amazon Kendra index \(p. 218\)](#)

- Filtering your search results (p. 222)

Querying your Amazon Kendra index

You can query your Amazon Kendra index using questions that correspond to the three kinds of queries that Amazon Kendra supports. For more information, see [Queries](#).

The example questions in this section have been chosen based on the sample dataset.

To query your Amazon Kendra index (Console)

1. Open the Amazon Kendra console at <https://console.aws.amazon.com/kendra/>.
2. From the **Indexes** list, click on **kendra-index**.
3. From the left navigation menu, choose **Search console**.
4. To run a sample factoid query, enter **Who is Lewis Hamilton?** in the search box and press enter.

The first returned result is the Amazon Kendra suggested answer, together with the data file containing the answer. The rest of the results form the set of recommended documents.

The screenshot shows the Amazon Kendra Search Console interface. A search bar at the top contains the query "Who is Lewis Hamilton?". Below the search bar, a message says "▶ Test query with user name or groups" and "1-8 of 8 results". A section titled "Amazon Kendra suggested answers" displays the first result, which is a link to an S3 object named "7d87db6157b9a3142a96dd6f4a13f85b555c4f24". This result is titled "Formula One driver" and includes a snippet of text from CNN about Lewis Hamilton's tweet. At the bottom of this card, there are "Like" and "Dislike" buttons and a link to "https://s3.us-east-2.amazonaws.com/.../7d87db6157b9a3142a96dd6f4a13f85...". Below this card, there is another partial result snippet from CNN. A "Sort: Relevance" dropdown is visible at the bottom right.

5. To run a descriptive query, enter **How does Formula One work?** in the search box and press enter.

You will see another result returned by the Amazon Kendra console, this time with the relevant phrase highlighted.

The screenshot shows the Amazon Kendra search interface with the query "How does Formula One work?". The results are sorted by relevance. The first result is a document titled "cbc7bc8dc746845435064e23e1d6ae95dacc0b38" which discusses the Formula One circus taking over five continents. The second result is a CNN article about Sebastian Vettel's number choice. The third result is another CNN article about drivers' championship numbers. Each result includes a URL, a snippet of text, and like/dislike buttons.

- To run a keyword search, enter **Formula One** in the search box and press enter.

You will see another result returned by the Amazon Kendra console, followed by the results for all other mentions of the phrase in the dataset.

The screenshot shows the Amazon Kendra search interface with the query "Formula One". The results are sorted by relevance. The first result is a document titled "f2c943b3c66ea02d0dc6b346132c84dd6bc53ed4" which discusses Sebastian Vettel's number choice. The second result is a CNN article about drivers' championship numbers. The third result is another CNN article about Sebastian Vettel's number choice. Each result includes a URL, a snippet of text, and like/dislike buttons. A note at the bottom indicates that the suggested answers are from Amazon Kendra.

To query your Amazon Kendra index (AWS CLI)

- To run a sample factoid query, use the [query](#) command:

Linux

```
aws kendra query \
    --index-id kendra-index-id \
    --query-text "Who is Lewis Hamilton?" \
```

```
--region aws-region
```

Where:

- *kendra-index-id* is your saved kendra-index-id,
- *aws-region* is your AWS region.

macOS

```
aws kendra query \  
    --index-id kendra-index-id \  
    --query-text "Who is Lewis Hamilton?" \  
    --region aws-region
```

Where:

- *kendra-index-id* is your saved kendra-index-id,
- *aws-region* is your AWS region.

Windows

```
aws kendra query ^  
    --index-id kendra-index-id ^  
    --query-text "Who is Lewis Hamilton?" ^  
    --region aws-region
```

Where:

- *kendra-index-id* is your saved kendra-index-id,
- *aws-region* is your AWS region.

The AWS CLI displays the results of your query.

2. To run a sample descriptive query, use the [query](#) command:

Linux

```
aws kendra query \  
    --index-id kendra-index-id \  
    --query-text "How does Formula One work?" \  
    --region aws-region
```

Where:

- *kendra-index-id* is your saved kendra-index-id,
- *aws-region* is your AWS region.

macOS

```
aws kendra query \  
    --index-id kendra-index-id \  
    --query-text "How does Formula One work?" \  
    --region aws-region
```

Where:

- *kendra-index-id* is your saved kendra-index-id,
- *aws-region* is your AWS region.

Windows

```
aws kendra query ^
    --index-id kendra-index-id ^
    --query-text "How does Formula One work?" ^
    --region aws-region
```

Where:

- *kendra-index-id* is your saved kendra-index-id,
- *aws-region* is your AWS region.

The AWS CLI displays the results to your query.

3. To run a sample keyword search, use the [query](#) command:

Linux

```
aws kendra query \
    --index-id kendra-index-id \
    --query-text "Formula One" \
    --region aws-region
```

Where:

- *kendra-index-id* is your saved kendra-index-id,
- *aws-region* is your AWS region.

macOS

```
aws kendra query \
    --index-id kendra-index-id \
    --query-text "Formula One" \
    --region aws-region
```

Where:

- *kendra-index-id* is your saved kendra-index-id,
- *aws-region* is your AWS region.

Windows

```
aws kendra query ^
    --index-id kendra-index-id ^
    --query-text "Formula One" ^
    --region aws-region
```

Where:

- *kendra-index-id* is your saved kendra-index-id,
 - *aws-region* is your AWS region.

The AWS CLI displays the returned answers to your query.

Filtering your search results

You can filter and sort your search results using custom document attributes in the Amazon Kendra search console. For more information on how Amazon Kendra processes queries, see [Filtering queries](#).

To filter your search results (Console)

1. Open the Amazon Kendra console at <https://console.aws.amazon.com/kendra/>.
 2. From the **Indexes** list, click on kendra-index.
 3. From the left navigation menu, choose **Search console**.
 4. In the search box, enter **Soccer matches** as a query and press enter.
 5. From the left navigation menu, choose **Filter search results** to see a list of facets you can use to filter your search.
 6. Select the check box for "Champions League" under the **EVENT** subheading, to see your search results filtered only by the results containing "Champions League".

To filter your search results (AWS CLI)

1. To see the entities of a specific type (such as EVENT) that are available for a search, use the [query](#) command:

Linux

```
aws kendra query \
--index-id kendra-index-id \
--query-text "Soccer matches" \
--facets '[{"DocumentAttributeKey": "EVENT"}]' \
--region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *aws-region* is your AWS region.

macOS

```
aws kendra query \
--index-id kendra-index-id \
--query-text "Soccer matches" \
--facets '[{"DocumentAttributeKey": "EVENT"}]' \
--region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *aws-region* is your AWS region.

Windows

```
aws kendra query ^
--index-id kendra-index-id ^
--query-text "Soccer matches" ^
--facets '[{"DocumentAttributeKey": "EVENT"}]' ^
--region aws-region
```

Where:

- *kendra-index-id* is your saved `kendra-index-id`,
- *aws-region* is your AWS region.

The AWS CLI displays the search results. To get a list of facets of type EVENT, navigate to the "FacetResults" section of the AWS CLI output to see a list of filterable facets with their counts. For example, one of the facets is "Champions League".

Note

Instead of EVENT, you can choose any of the index fields you created in [the section called "Creating an Amazon Kendra index" \(p. 201\)](#) for the `DocumentAttributeKey` value.

2. To run the same search but filter only by the results containing "Champions League", use the [query](#) command:

Linux

```
aws kendra query \  
    --index-id kendra-index-id \  
    --query-text "Soccer matches" \  
    --attribute-filter '{"ContainsAny":{"Key":"EVENT","Value":  
        {"StringListValue":["Champions League"]}}}' \  
    --region aws-region
```

Where:

- **kendra-index-id** is your saved kendra-index-id,
- **aws-region** is your AWS region.

macOS

```
aws kendra query \  
    --index-id kendra-index-id \  
    --query-text "Soccer matches" \  
    --attribute-filter '{"ContainsAny":{"Key":"EVENT","Value":  
        {"StringListValue":["Champions League"]}}}' \  
    --region aws-region
```

Where:

- **kendra-index-id** is your saved kendra-index-id,
- **aws-region** is your AWS region.

Windows

```
aws kendra query ^  
    --index-id kendra-index-id ^  
    --query-text "Soccer matches" ^  
    --attribute-filter '{"ContainsAny":{"Key":"EVENT","Value":  
        {"StringListValue":["Champions League"]}}}' ^  
    --region aws-region
```

Where:

- **kendra-index-id** is your saved kendra-index-id,
- **aws-region** is your AWS region.

The AWS CLI displays the filtered search results.

Step 6: Cleaning up

Cleaning up your files

To stop incurring charges in your AWS account after you complete this tutorial, you can take the following steps:

1. **Delete your Amazon S3 bucket**

For information about deleting a bucket, see [Deleting a bucket](#).

2. Delete your Amazon Kendra index

For information about deleting an Amazon Kendra index, see [Deleting an index](#).

3. Delete converter.py

- **For Console:** Go to [AWS CloudShell](#), and make sure the region is set to your AWS region. After the bash shell has loaded, type the following command into the environment and press enter.

```
rm converter.py
```

- **For AWS CLI:** Run the following command on a terminal window.

Linux

```
rm file/converter.py
```

Where:

- *file/* is the filepath to converter.py on your local device.

macOS

```
rm file/converter.py
```

Where:

- *file/* is the filepath to converter.py on your local device.

Windows

```
rm file/converter.py
```

Where:

- *file/* is the filepath to converter.py on your local device.

Learn more

To learn more about integrating Amazon Kendra into your workflow, you can check out the following blogposts:

- [Content metadata tagging for enhanced search](#)
- [Build an intelligent search solution with automated content enrichment](#)

To learn more about Amazon Comprehend, you can look at the [Amazon Comprehend Developer Guide](#).

Monitoring and logging for Amazon Kendra

Topics

- [Monitoring your index \(Console\) \(p. 226\)](#)
- [Logging Amazon Kendra API calls with AWS CloudTrail logs \(p. 229\)](#)
- [Monitoring Amazon Kendra with Amazon CloudWatch \(p. 231\)](#)
- [Monitoring Amazon Kendra with Amazon CloudWatch Logs \(p. 235\)](#)

Monitoring your index (Console)

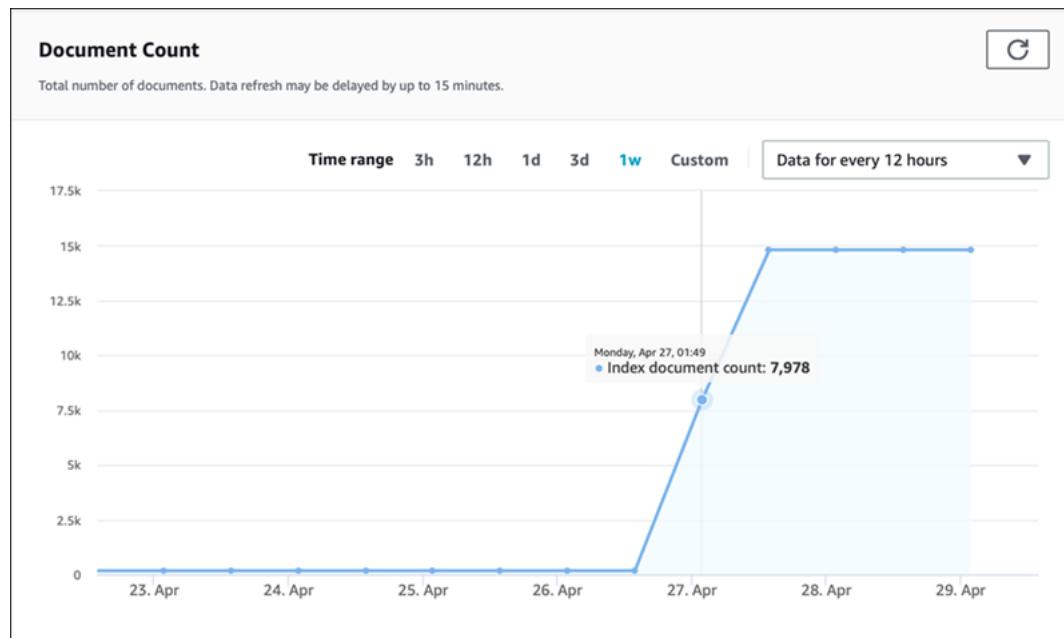
Use the Amazon Kendra console to monitor the state of indexes and data sources. You can use this information to track the size and storage requirements of your index and to monitor the progress and success of synchronization between your index and data sources.

To view index metrics (Console)

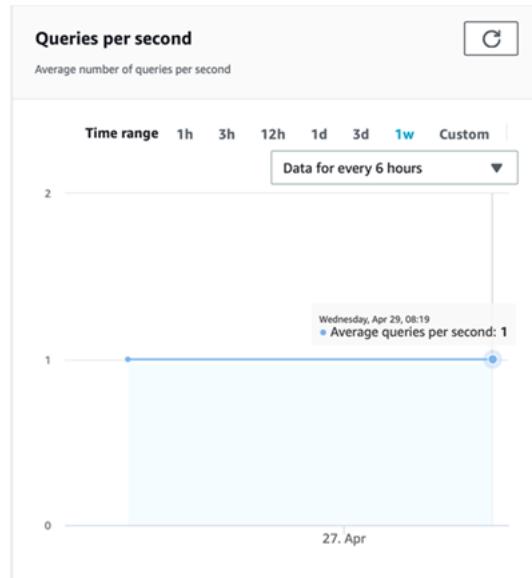
1. Sign into the AWS Management Console and open the Amazon Kendra console at <https://console.aws.amazon.com/kendra/home>.
2. From the list of indexes, choose the index to view.
3. Scroll the screen to see the index metrics.

You can see the following metrics about your index.

- **Document count** – The total number of documents indexed. This includes all documents from all data sources. Use this metric to determine if you need to purchase more or fewer storage units for your index.



- **Queries per second** – The number of index queries that are requested each second. Use this metric to determine if you need to purchase more or fewer query units for your index.



To monitor the progress and success of synchronization between your index and a data source, use the Amazon Kendra console. Use this information to help determine the health of your data source.

To view synchronization metrics (Console)

1. Sign into the AWS Management Console and open the Amazon Kendra console at <https://console.aws.amazon.com/kendra/home>.
2. From the list of indexes, choose the index to view synchronization metrics for.
3. From the left menu, choose **Data sources**.
4. From the list of data sources, choose the data source to view.

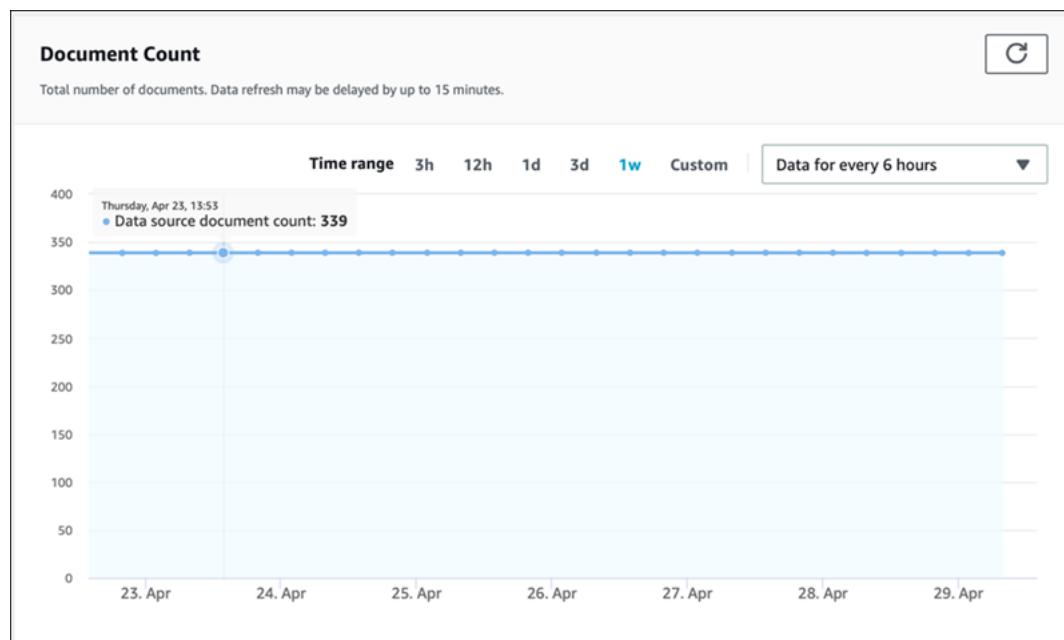
5. Scroll the screen to see the sync run metrics.

You can see the following information.

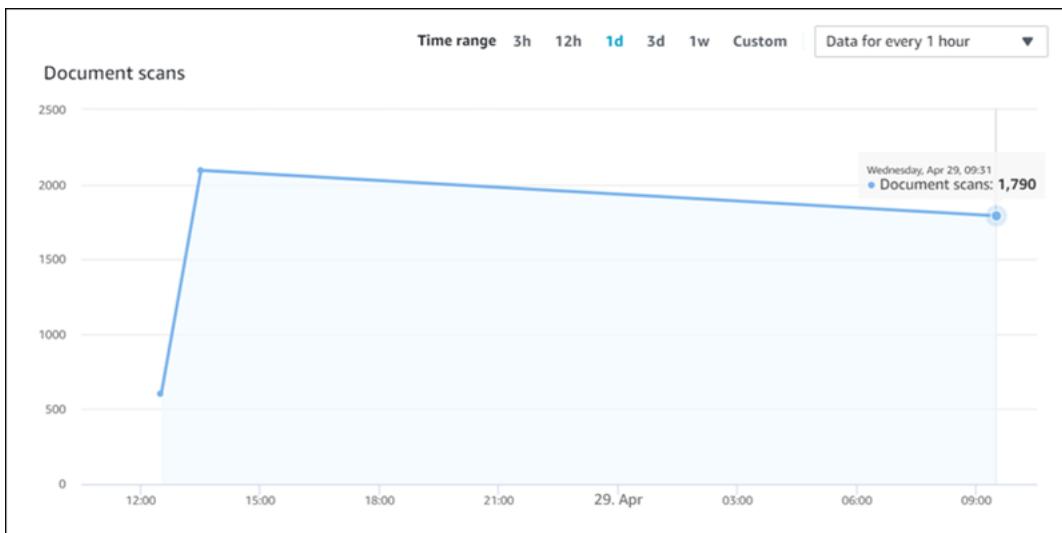
- **Sync run history** – Statistics about the synchronization run, including the start and end time, the number of documents added, deleted, and failed. If the sync run fails, there is a link to CloudWatch Logs with more information. Choose the settings icon in the upper left to change the columns that are displayed in the history. Use this information to determine the general health of your data source.

Sync run history (5)						
Status / Summary	Start time	End time	Added / Modified	Deleted	Failed	Details
Syncing - indexing	Apr 29, 2020, 9:53 AM PDT	Apr 29, 2020, 9:54 AM PDT	0	0	0	View in CloudWatch
Succeeded	Apr 28, 2020, 1:35 PM PDT	Apr 28, 2020, 1:37 PM PDT	1484	0	2	Service is operating normally []
Succeeded	Apr 28, 2020, 1:32 PM PDT	Apr 28, 2020, 1:32 PM PDT	0	0	0	Service is operating normally []
Succeeded	Apr 28, 2020, 1:05 PM PDT	Apr 28, 2020, 1:06 PM PDT	5	0	0	Service is operating normally []
Succeeded	Apr 28, 2020, 1:05 PM PDT	Apr 28, 2020, 1:05 PM PDT	298	0	1	Service is operating normally []

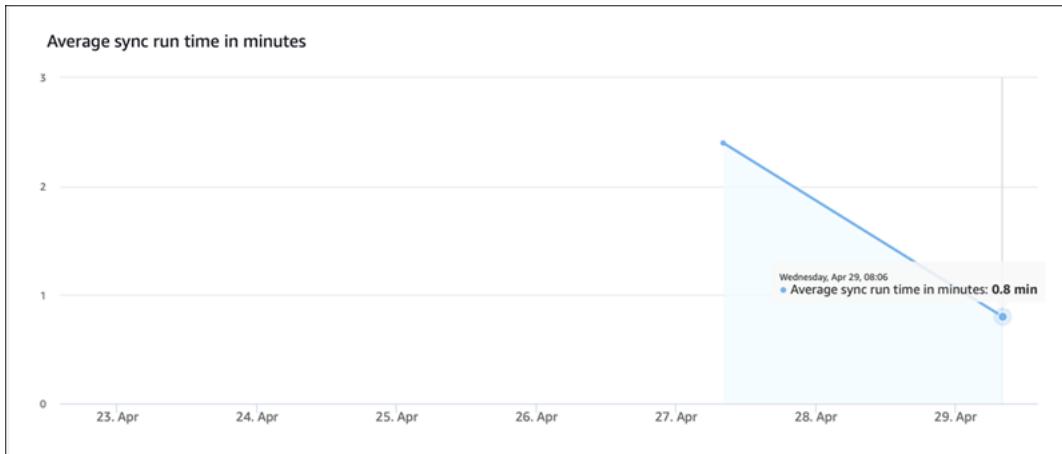
- **Document count** – The total number of documents indexed from this data source. This is the total of all documents added to the data source minus the total of all documents deleted from the data source. Use this information to determine how many documents from this data source are included in the index.



- **Document scans** – The total number of documents scanned during the sync run. This includes all documents in the data source, including those added, updated, deleted, or unchanged. Use this information to determine if Amazon Kendra is scanning all of the documents in the data source. The number of documents scanned affects the amount charged for the service.



- **Average sync run time in minutes** – The average length of time that it takes for a sync run to complete. The time that it takes to sync a data source affects the amount charged for the service.



Logging Amazon Kendra API calls with AWS CloudTrail logs

Amazon Kendra is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Amazon Kendra. CloudTrail captures all API calls from Amazon Kendra as events, including calls from the Amazon Kendra console and from code calls to the Amazon Kendra APIs. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Amazon Kendra. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to Amazon Kendra, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, including how to configure and enable it, see the [AWS CloudTrail User Guide](#).

Amazon Kendra Information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Amazon Kendra, that activity is recorded in a CloudTrail event along with other AWS service events in the CloudTrail **Event history**. You can view, search, and download recent events in your AWS account. For more information, see [Viewing Events with CloudTrail Event History](#).

For an ongoing record of events in your AWS account, including events for Amazon Kendra, create a trail. A *trail* is a configuration that enables CloudTrail to deliver events as log files to a specified S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see:

- [Overview for Creating a Trail](#)
- [CloudTrail Supported Services and Integrations](#)
- [Configuring Amazon SNS Notifications for CloudTrail](#)
- [Receiving CloudTrail Log Files from Multiple Regions](#) and [Receiving CloudTrail Log Files from Multiple Accounts](#)

CloudTrail logs all Amazon Kendra actions, which are documented in the [API Reference \(p. 269\)](#). For example, calls to the `CreateIndex`, `CreateDataSource`, and `Query` operations generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. For more information, see the [CloudTrail userIdentity Element](#).

Example: Amazon Kendra log file Entries

A *trail* is a configuration that enables delivery of events as log files to a specified S3 bucket. CloudTrail log files contain one or more log entries. An *event* represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

Calls to the `Query` operation creates the following entry.

```
{  
    "eventVersion": "1.05",  
    "userIdentity": {  
        "type": "AssumedRole | FederatedUser | IAMUser | Root | SAMLUser | WebIdentityUser",  
        "principalId": "principal ID",  
        "arn": "ARN",  
        "accountId": "account ID",  
        "accessKeyId": "access key ID",  
        "sessionContext": {  
            "sessionIssuer": {  
                "type": "Role",  
                "principalId": "principal ID",  
                "arn": "ARN",  
                "accountId": "account ID",  
                "userName": "user name"  
            },  
            "webIdFederationData": {  
            }  
        }  
    }  
}
```

```
        "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "timestamp"
        }
    },
    "eventTime": "timestamp",
    "eventSource": "kendra.amazonaws.com",
    "eventName": "Query",
    "awsRegion": "region",
    "sourceIPAddress": "source IP address",
    "userAgent": "user agent",
    "requestParameters": {
        "indexId": "index ID"
    },
    "responseElements": null,
    "requestID": "request ID",
    "eventID": "event ID",
    "eventType": "AwsApiCall",
    "recipientAccountId": "account ID"
},
```

Monitoring Amazon Kendra with Amazon CloudWatch

To track the health of your indexes, use Amazon CloudWatch. With CloudWatch, you can get metrics for document synchronization for your index. You can also set up CloudWatch alarms to be notified when one or more metrics exceeds a threshold that you define. For example, you can monitor the number of documents submitted to be indexed or the number of documents that failed to be indexed.

You must have the appropriate CloudWatch permissions to monitor Amazon Kendra with CloudWatch. For more information, see [Authentication and Access Control for Amazon CloudWatch](#) in the *Amazon CloudWatch User Guide*.

Viewing Amazon Kendra metrics

View Amazon Kendra metrics using the CloudWatch console.

To view metrics (CloudWatch console)

1. Sign in to the AWS Management Console and open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. Choose **Metrics**, choose **All Metrics** and then choose **Kendra**.
3. Choose the dimension, choose a metric name, then choose **Add to graph**.
4. Choose a value for the date range. The metric count for the selected date range is displayed in the graph.

Creating an alarm

A CloudWatch alarm watches a single metric over a specified time period and performs one or more actions: sending a notification to an Amazon Simple Notification Service (Amazon SNS) topic or Auto Scaling policy. The actions or notifications are based on the value of the metric relative to a given threshold over a number of time periods that you specify. CloudWatch can also send you an Amazon SNS message when the alarm changes state.

CloudWatch alarms invoke actions only when the state changes and has persisted for the period that you specify.

To set an alarm

1. Sign in to the AWS Management Console and open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. Choose **Alarms** and then choose **Create Alarm**.
3. Choose **Kendra metrics** and then choose a metric.
4. For **Time Range**, choose a time range to monitor, and then choose **Next**.
5. Enter a **Name** and **Description**.
6. For **Whenever**, choose \geq , and type a maximum value.
7. If you want CloudWatch to send an email when the alarm state is reached, in the **Actions** section, for **Whenever this alarm**, choose **State is ALARM**. For **Send notification to**, choose a mailing list or choose **New list** and create a new mailing list
8. Preview the alarm in the **Alarm Preview** section. If you are satisfied with the alarm, choose **Create Alarm**.

CloudWatch Metrics for index synchronization Jobs

The following table describes the Amazon Kendra metrics for data source synchronization jobs.

Metric	Description
DocumentsCrawled	<p>The number of documents that the synchronization job scanned or discovered during the run.</p> <p>Dimensions:</p> <ul style="list-style-type: none"> • IndexId • DataSourceld <p>Unit: Count</p>
DocumentsSubmittedForIndexing	<p>The number of documents that the synchronization job submitted to the index.</p> <p>Dimensions:</p> <ul style="list-style-type: none"> • IndexId • DataSourceld <p>Unit: Count</p>
DocumentsSubmittedForIndexingFailed	<p>The number of documents that failed indexing. Check the contents of the CloudWatch log for the synchronization job for details.</p> <p>Dimensions:</p> <ul style="list-style-type: none"> • IndexId • DataSourceld

Metric	Description
	Unit: Count
DocumentsSubmittedForDeletion	<p>The number of documents that the synchronization job asked to be removed from the index.</p> <p>Dimensions:</p> <ul style="list-style-type: none"> • IndexId • DataSourceId
	Unit: Count
DocumentsSubmittedForDeletionFailed	<p>The number of documents that failed to be deleted. Check the contents of the CloudWatch log for the synchronization job for details.</p> <p>Dimensions:</p> <ul style="list-style-type: none"> • IndexId • DataSourceId
	Unit: Count

Metrics for Amazon Kendra data sources

The following table describes the Amazon Kendra metrics for data source synchronization jobs. Metrics marked with an asterisk (*) are used only for Amazon S3 data sources.

Metric	Description
DocumentsSkippedNoChange *	<p>The number of documents examined and found not to have changed so they weren't submitted for indexing.</p> <p>Dimensions:</p> <ul style="list-style-type: none"> • IndexId • DataSourceId
	Unit: Count
DocumentsSkippedInvalidMetadata *	<p>The number of documents skipped because there was a problem with the associated metadata file. Check the contents of the CloudWatch log for the synchronization run for details.</p> <p>Dimensions:</p> <ul style="list-style-type: none"> • IndexId • DataSourceId

Metric	Description
	Unit: Count
DocumentsCrawled	<p>The number of document files examined.</p> <p>Dimensions:</p> <ul style="list-style-type: none"> • IndexId • DataSourceId
	Unit: Count
DocumentsSubmittedForDeletion	<p>The number of documents examined that were deleted from the data source and submitted for deletion.</p> <p>Dimensions:</p> <ul style="list-style-type: none"> • IndexId • DataSourceId
	Unit: Count
DocumentsSubmittedForDeletionFailed	<p>The number of documents that failed deletion from a data source.</p> <p>Dimensions:</p> <ul style="list-style-type: none"> • IndexId • DataSourceId
	Unit: Count
DocumentsSubmittedForIndexing	<p>The number of documents examined and submitted for indexing.</p> <p>Dimensions:</p> <ul style="list-style-type: none"> • IndexId • DataSourceId
	Unit: Count
DocumentsSubmittedForIndexingFailed	<p>The number of documents submitted for indexing that couldn't be indexed.</p> <p>Dimensions:</p> <ul style="list-style-type: none"> • IndexId • DataSourceId
	Unit: Count

Metrics for indexed documents

The following table describes the Amazon Kendra metrics for indexed documents. For documents that are indexed using the [BatchPutDocument \(p. 277\)](#) operation, only the `IndexId` dimension is supported.

Metric	Description
<code>DocumentsIndexed</code>	<p>The number of documents indexed.</p> <p>Dimensions:</p> <ul style="list-style-type: none"> • <code>IndexId</code> • <code>DataSourceId</code> <p>Unit: Count</p>
<code>DocumentsFailedToIndex</code>	<p>The number of documents that could not be indexed. Check the contents of the CloudWatch log for details.</p> <p>Dimensions:</p> <ul style="list-style-type: none"> • <code>IndexId</code> • <code>DataSourceId</code> <p>Unit: Count</p>
<code>IndexQueryCount</code>	<p>The number of index queries per minute.</p> <p>Dimensions:</p> <ul style="list-style-type: none"> • <code>IndexId</code> <p>Unit: Count</p>

Monitoring Amazon Kendra with Amazon CloudWatch Logs

Amazon Kendra uses Amazon CloudWatch Logs to give you insight into the operation of your data sources. Amazon Kendra logs process details for the documents that are indexed. It logs errors from your data source that occur while your documents are being indexed. You use CloudWatch Logs to monitor, store and access the log files.

CloudWatch Logs stores log events in a log stream that is part of a log group. Amazon Kendra uses these features as follows:

- Log groups – Amazon Kendra stores all of your log streams in a single log group for each index. Amazon Kendra creates the log group when the index is created. The log group identifier always begins with "aws/kendra/".

- Log stream – creates a new data source log stream in the log group for each index synchronization job that you run. It also creates a new document log stream when a stream reaches approximately 500 entries.
- Log entries – Amazon Kendra creates a log entry in the log stream as it indexes documents. Each entry provides information about processing the document or any errors that are encountered.

For more information about using CloudWatch Logs, see [What Is Amazon Cloud Watch Logs in the Amazon Cloud Watch Logs User Guide](#).

Amazon Kendra creates two types of log streams:

- [Data source log streams \(p. 236\)](#)
- [Document log streams \(p. 237\)](#)

Data source log streams

Data source log streams publish entries about your index synchronization jobs. Each synchronization job creates a new log stream that it uses to publish entries. The log stream name is:

```
data source id/YYYY-MM-DD-HH/data source sync job ID
```

A new log stream is created for each synchronization job run.

There are three types of log messages published to a data source log stream:

- A log message for a document that failed to be sent for indexing. The following is an example of this message for a document in an S3 data source:

```
{  
    "DocumentId": "document ID",  
    "S3Path": "s3://bucket/prefix/object",  
    "Message": "Failed to ingest document via BatchPutDocument.",  
    "ErrorCode": "InvalidRequest",  
    "ErrorMessage": "No document metadata configuration found for document attribute key city."  
}
```

- A log message for a document that failed to be sent for deletion. The following is an example of this message:

```
{  
    "DocumentId": "document ID",  
    "Message": "Failed to delete document via BatchDeleteDocument.",  
    "ErrorCode": "InvalidRequest",  
    "ErrorMessage": "Document can't be deleted because it doesn't exist."  
}
```

- A log message when an invalid metadata file for a document in an Amazon S3 bucket is found. The following is an example of this message.

```
{  
    "Message": "Found invalid metadata file bucket/prefix/filename.extension.metadata.json."  
}
```

- For SharePoint and database connectors, Amazon Kendra only writes messages to the log stream if a document can't be indexed. The following is an example of the error message that Amazon Kendra logs.

```
{  
    "DocumentID": "document ID",  
    "IndexID": "index ID",  
    "SourceURI": "",  
    "CrawlStatus": "FAILED",  
    "ErrorCode": "403",  
    "ErrorMessage": "Access Denied",  
    "DataSourceErrorCode": "403"  
}
```

Document log streams

Amazon Kendra logs information about processing documents while they are being indexed. It logs a set of messages for documents stored in an Amazon S3 data source. It logs errors only for documents stored in a Microsoft SharePoint or a database data source.

If the documents were added to the index using the [BatchPutDocument \(p. 277\)](#) operation, the log stream is named as follows:

```
YYYY-MM-DD-HH/UUID
```

If the documents were added to the index using a datasource, the log stream is named as follows:

```
dataSourceId/YYYY-MM-DD-HH/UUID
```

Each log stream contains up to 500 messages.

If indexing a document fails, this message is output to the log stream:

```
{  
    "DocumentId": "document ID",  
    "IndexName": "index name",  
    "IndexId": "index ID",  
    "SourceURI": "source URI",  
    "IndexingStatus": "DocumentFailedToIndex",  
    "ErrorCode": "400 | 500",  
    "ErrorMessage": "message"  
}
```

Security in Amazon Kendra

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The [shared responsibility model](#) describes this as security of the cloud and security *in the cloud*:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the [AWS Compliance Programs](#). To learn about the compliance programs that apply to Amazon Kendra, see [AWS Services in Scope by Compliance Program](#).
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using Amazon Kendra. The following topics show you how to configure Amazon Kendra to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Amazon Kendra resources.

Topics

- [Data protection in Amazon Kendra \(p. 238\)](#)
- [Amazon Kendra and interface VPC endpoints \(AWS PrivateLink\) \(p. 240\)](#)
- [Identity and access management for Amazon Kendra \(p. 241\)](#)
- [Logging and monitoring in Amazon Kendra \(p. 257\)](#)
- [Compliance validation for Amazon Kendra \(p. 258\)](#)
- [Resilience in Amazon Kendra \(p. 258\)](#)
- [Infrastructure security in Amazon Kendra \(p. 258\)](#)

Data protection in Amazon Kendra

The AWS [shared responsibility model](#) applies to data protection in Amazon Kendra. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. This content includes the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the [Data Privacy FAQ](#). For information about data protection in Europe, see the [AWS Shared Responsibility Model and GDPR](#) blog post on the [AWS Security Blog](#).

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user accounts with AWS Identity and Access Management (IAM). That way each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We recommend TLS 1.2 or later.

- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see [Federal Information Processing Standard \(FIPS\) 140-2](#).

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form fields such as a **Name** field. This includes when you work with Amazon Kendra or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Encryption at rest

Amazon Kendra encrypts your data at rest with your choice of an encryption key. You can choose one of the following:

- An AWS owned customer master key (CMK). If you don't specify an encryption key your data is encrypted with this key by default.
- An AWS managed CMK in your account. This key is created, managed, and used on your behalf by Amazon Kendra. The key name is `aws/kendra`.
- A customer managed CMK. You can provide the ARN of an encryption key that you created in your account. When you use a customer managed CMK, you must give the key a key policy that enables Amazon Kendra to use the key. Select a symmetric customer managed CMK, Amazon Kendra does not support asymmetric CMKs. For more information, see [Key management \(p. 239\)](#).

Encryption in transit

Amazon Kendra uses the HTTPS protocol to communicate with your client application. It uses HTTPS and AWS signatures to communicate with other services on your application's behalf. If you use a VPC, you can use AWS PrivateLink to establish a private connection between your VPC and Amazon Kendra.

Key management

Amazon Kendra encrypts the contents of your index using one of three types of keys. You can choose one of the following:

- An AWS owned customer master key (CMK). This is the default.
- An AWS managed CMK. This key is created in your account and is managed and used on your behalf by Amazon Kendra.
- A customer managed CMK. You can create the key when you are creating an Amazon Kendra index or data source, or you can create the key using the AWS KMS console. Select a symmetric customer managed CMK, Amazon Kendra does not support asymmetric CMKs. For more information, see [Using Symmetric and Asymmetric Keys](#) in the *AWS Key Management Service Developer Guide*.

When you create a key using the AWS KMS console, you must give the key the following policy that enables Amazon Kendra to use the key. If you create a key with the Amazon Kendra console, the policy is applied to the key for you. For more information, see [Using Key Policies in AWS KMS](#) in the *AWS Key Management Service Developer Guide*.

You should specify only the CMKs that Amazon Kendra needs to use in the Resource element to ensure permissions or actions are limited only to the specified CMKs.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "kendra.amazonaws.com"  
            },  
            "Action": [  
                "kms:Encrypt",  
                "kms:Decrypt",  
                "kms:ReEncrypt*",  
                "kms:GenerateDataKey*",  
                "kms:DescribeKey",  
                "kms>CreateGrant",  
                "kms:RetireGrant"  
            ],  
            "Resource": [  
                "arn:aws:kms:region:account ID:key/key ID"]  
        }  
    ]  
}
```

Amazon Kendra and interface VPC endpoints (AWS PrivateLink)

You can establish a private connection between your VPC and Amazon Kendra by creating an *interface VPC endpoint*. Interface endpoints are powered by [AWS PrivateLink](#), a technology that enables you to privately access Amazon Kendra APIs without an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC don't need public IP addresses to communicate with Amazon Kendra APIs. Traffic between your VPC and Amazon Kendra does not leave the Amazon network.

Each interface endpoint is represented by one or more [Elastic Network Interfaces](#) in your subnets.

For more information, see [Interface VPC endpoints \(AWS PrivateLink\)](#) in the *Amazon VPC User Guide*.

Considerations for Amazon Kendra VPC endpoints

Before you set up an interface VPC endpoint for Amazon Kendra, make sure that you review [Interface endpoint properties and limitations](#) in the *Amazon VPC User Guide*.

Amazon Kendra supports making calls to all of its API actions from your VPC.

Creating an interface VPC endpoint for Amazon Kendra

You can create a VPC endpoint for the Amazon Kendra service using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI). For more information, see [Creating an interface endpoint](#) in the *Amazon VPC User Guide*.

Create a VPC endpoint for Amazon Kendra using the following service name:

- com.amazonaws.*region*.kendra

After you create a VPC endpoint, you can use the following example AWS CLI command that uses the `endpoint-url` parameter to specify an interface endpoint to the Amazon Kendra API:

```
aws kendra list-indices --endpoint-url https://VPC endpoint
```

where *VPC endpoint* is the DNS name generated when the interface endpoint is created. This name includes the VPC endpoint ID, Amazon Kendra service name and Region name. For example, `vpce-1234-abcdef.kendra.us-west-2.vpce.amazonaws.com`.

If you enable private DNS for the endpoint, you can make API requests to Amazon Kendra using its default DNS name for the Region, for example, `kendra.us-east-1.amazonaws.com`.

For more information, see [Accessing a service through an interface endpoint](#) in the *Amazon VPC User Guide*.

Creating a VPC endpoint policy for Amazon Kendra

You can attach an endpoint policy to your VPC endpoint that controls access to Amazon Kendra. The policy specifies the following information:

- The principal that can perform actions.
- The actions that can be performed.
- The resources on which actions can be performed.

For more information, see [Controlling access to services with VPC endpoints](#) in the *Amazon VPC User Guide*.

Example: VPC endpoint policy for Amazon Kendra actions

The following is an example of an endpoint policy for Amazon Kendra. When attached to an endpoint, this policy grants access to the listed Amazon Kendra actions for all principals on all resources.

```
{  
    "Statement": [  
        {  
            "Principal": "*",
            "Effect": "Allow",
            "Action": [  
                "Query"
            ],
            "Resource": "*"
        }
    ]
}
```

Identity and access management for Amazon Kendra

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use Amazon Kendra resources. IAM is an AWS service that you can use with no additional charge.

Topics

- [Audience \(p. 242\)](#)
- [Authenticating with identities \(p. 242\)](#)
- [Managing access using policies \(p. 244\)](#)
- [How Amazon Kendra works with IAM \(p. 246\)](#)
- [Amazon Kendra Identity-based policy examples \(p. 249\)](#)
- [AWS managed policies for Amazon Kendra \(p. 253\)](#)
- [Troubleshooting Amazon Kendra Identity and Access \(p. 256\)](#)

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in Amazon Kendra.

Service user – If you use the Amazon Kendra service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Amazon Kendra features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in Amazon Kendra, see [Troubleshooting Amazon Kendra Identity and Access \(p. 256\)](#).

Service administrator – If you're in charge of Amazon Kendra resources at your company, you probably have full access to Amazon Kendra. It's your job to determine which Amazon Kendra features and resources your employees should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Amazon Kendra, see [How Amazon Kendra works with IAM \(p. 246\)](#).

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to Amazon Kendra. To view example Amazon Kendra identity-based policies that you can use in IAM, see [Amazon Kendra Identity-based policy examples \(p. 249\)](#).

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. For more information about signing in using the AWS Management Console, see [Signing in to the AWS Management Console as an IAM user or root user](#) in the *IAM User Guide*.

You must be *authenticated* (signed in to AWS) as the AWS account root user, an IAM user, or by assuming an IAM role. You can also use your company's single sign-on authentication or even sign in using Google or Facebook. In these cases, your administrator previously set up identity federation using IAM roles. When you access AWS using credentials from another company, you are assuming a role indirectly.

To sign in directly to the [AWS Management Console](#), use your password with your root user email address or your IAM user name. You can access AWS programmatically using your root user or IAM users access keys. AWS provides SDK and command line tools to cryptographically sign your request using your credentials. If you don't use AWS tools, you must sign the request yourself. Do this using *Signature Version 4*, a protocol for authenticating inbound API requests. For more information about authenticating requests, see [Signature Version 4 signing process](#) in the *AWS General Reference*.

Regardless of the authentication method that you use, you might also be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see [Using multi-factor authentication \(MFA\) in AWS](#) in the *IAM User Guide*.

AWS account root user

When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the [best practice of using the root user only to create your first IAM user](#). Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

IAM Users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. An IAM user can have long-term credentials such as a user name and password or a set of access keys. To learn how to generate access keys, see [Managing access keys for IAM users](#) in the *IAM User Guide*. When you generate access keys for an IAM user, make sure you view and securely save the key pair. You cannot recover the secret access key in the future. Instead, you must generate a new access key pair.

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see [When to create an IAM user \(instead of a role\)](#) in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by [switching roles](#). You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see [Using IAM roles](#) in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Temporary IAM user permissions** – An IAM user can assume an IAM role to temporarily take on different permissions for a specific task.
- **Federated user access** – Instead of creating an IAM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as *federated users*. AWS assigns a role to a federated user when access is requested through an *identity provider*. For more information about federated users, see [Federated users and roles](#) in the *IAM User Guide*.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see [How IAM roles differ from resource-based policies](#) in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.

- **Principal permissions** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see [Actions, Resources, and Condition Keys for Amazon Kendra](#) in the *Service Authorization Reference*.
- **Service role** – A service role is an [IAM role](#) that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see [Creating a role to delegate permissions to an AWS service](#) in the *IAM User Guide*.
- **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see [Using an IAM role to grant permissions to applications running on Amazon EC2 instances](#) in the *IAM User Guide*.

To learn whether to use IAM roles or IAM users, see [When to create an IAM role \(instead of a user\)](#) in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to IAM identities or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. You can sign in as the root user or an IAM user, or you can assume an IAM role. When you then make a request, AWS evaluates the related identity-based or resource-based policies. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see [Overview of JSON policies](#) in the *IAM User Guide*.

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

Every IAM entity (user or role) starts with no permissions. In other words, by default, users can do nothing, not even change their own password. To give a user permission to do something, an administrator must attach a permissions policy to a user. Or the administrator can add the user to a group that has the intended permissions. When an administrator gives permissions to a group, all users in that group are granted those permissions.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see [Creating IAM policies](#) in the *IAM User Guide*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that

you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see [Choosing between managed policies and inline policies](#) in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must [specify a principal](#) in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Access control lists (ACLs)

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see [Access control list \(ACL\) overview](#) in the *Amazon Simple Storage Service Developer Guide*.

Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see [Permissions boundaries for IAM entities](#) in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see [How SCPs work](#) in the *AWS Organizations User Guide*.
- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see [Session policies](#) in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see [Policy evaluation logic](#) in the *IAM User Guide*.

How Amazon Kendra works with IAM

Before you use IAM to manage access to Amazon Kendra, you should understand what IAM features are available to use with Amazon Kendra. To get a high-level view of how Amazon Kendra and other AWS services work with IAM, see [AWS Services That Work with IAM](#) in the *IAM User Guide*.

Topics

- [Amazon Kendra identity-based policies \(p. 246\)](#)
- [Amazon Kendra Resource-based policies \(p. 248\)](#)
- [Access control lists \(ACLs\) \(p. 248\)](#)
- [Authorization based on Amazon Kendra tags \(p. 248\)](#)
- [Amazon Kendra IAM Roles \(p. 248\)](#)

Amazon Kendra identity-based policies

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. Amazon Kendra supports specific actions, resources, and condition keys. To learn about all of the elements that you use in a JSON policy, see [IAM JSON Policy Elements Reference](#) in the *IAM User Guide*.

Actions

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The **Action** element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

Policy actions in Amazon Kendra use the following prefix before the action: `kendra:`. For example, to grant someone permission to list Amazon Kendra indexes with the [ListIndices \(p. 370\)](#) API operation, you include the `kendra>ListIndices` action in their policy. Policy statements must include either an **Action** or **NotAction** element. Amazon Kendra defines its own set of actions that describe tasks that you can perform with this service.

To specify multiple actions in a single statement, separate them with commas as follows:

```
"Action": [  
    "kendra:action1",  
    "kendra:action2"]
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word `Describe`, include the following action:

```
"Action": "kendra:Describe*"
```

To see a list of Amazon Kendra actions, see [Actions Defined by Amazon Kendra](#) in the *IAM User Guide*.

Resources

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The Resource JSON policy element specifies the object or objects to which the action applies. Statements must include either a Resource or a NotResource element. As a best practice, specify a resource using its [Amazon Resource Name \(ARN\)](#). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

The Amazon Kendra index resource has the following ARN:

```
arn:${Partition}:kendra:${Region}:${Account}:index/${IndexId}
```

For more information about the format of ARNs, see [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#).

For example, to specify an index in your statement, use the GUID of the index in the following ARN:

```
"Resource": "arn:aws:kendra:${Region}:${Account}:index/${GUID}"
```

To specify all indexes that belong to a specific account, use the wildcard (*):

```
"Resource": "arn:aws:${Region}:${Account}:index/*"
```

Some Amazon Kendra actions, such as those for creating resources, cannot be performed on a specific resource. In those cases, you must use the wildcard (*).

```
"Resource": "*"
```

To see a list of Amazon Kendra resource types and their ARNs, see [Resources Defined by Amazon Kendra](#) in the *IAM User Guide*. To learn with which actions you can specify the ARN of each resource, see [Actions Defined by Amazon Kendra](#).

Condition keys

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The Condition element (or Condition block) lets you specify conditions in which a statement is in effect. The Condition element is optional. You can create conditional expressions that use [condition operators](#), such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple Condition elements in a statement, or multiple keys in a single Condition element, AWS evaluates them using a logical AND operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical OR operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see [IAM policy elements: variables and tags](#) in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see [AWS global condition context keys](#) in the *IAM User Guide*.

Amazon Kendra does not provide any service-specific condition keys, but it does support using some global condition keys. To see all AWS global condition keys, see [AWS Global Condition Context Keys](#) in the *IAM User Guide*.

Examples

To view examples of Amazon Kendra identity-based policies, see [Amazon Kendra Identity-based policy examples \(p. 249\)](#).

Amazon Kendra Resource-based policies

Amazon Kendra does not support resource-based policies.

Access control lists (ACLs)

Amazon Kendra does not support access control lists (ACLs) for access to AWS services and resources.

Authorization based on Amazon Kendra tags

You can associate tags with certain types of Amazon Kendra resources to authorize access to those resources. To control access based on tags, provide tag information in the condition element of a policy by using the `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

The following table lists the actions, corresponding resource types, and condition keys for tag-based access control. Each action is authorized based on the tags associated with the corresponding resource type.

Action	Resource type	Condition keys
CreateDataSource (p. 283)		<code>aws:RequestTag</code> , <code>aws:TagKeys</code>
CreateFaq (p. 292)		<code>aws:RequestTag</code> , <code>aws:TagKeys</code>
CreateIndex (p. 296)		<code>aws:RequestTag</code> , <code>aws:TagKeys</code>
ListTagsForResource (p. 375)	data source, FAQ, index	
TagResource (p. 399)	data source, FAQ, index	<code>aws:RequestTag</code> , <code>aws:TagKeys</code>
UntagResource (p. 401)	data source, FAQ, index	<code>aws:TagKeys</code>

For information about tagging Amazon Kendra resources, see [Tags \(p. 8\)](#). For an example identity-based policy that limits access to a resource based on resource tags, see [Tag-based policy examples \(p. 251\)](#). For more information about using tags to limit access to resources, see [Controlling access using tags](#) in the *IAM User Guide*.

Amazon Kendra IAM Roles

An [IAM role](#) is an entity within your AWS account that has specific permissions.

Using temporary credentials with Amazon Kendra

You can use temporary credentials to sign in with federation, assume an IAM role, or to assume a cross-account role. You obtain temporary security credentials by calling AWS STS API operations such as [AssumeRole](#) or [GetFederationToken](#).

Amazon Kendra supports using temporary credentials.

Service roles

This feature allows a service to assume a [service role](#) on your behalf. This role allows the service to access resources in other services to complete an action on your behalf. Service roles appear in your IAM account and are owned by the account. This means that an IAM administrator can change the permissions for this role. However, doing so might break the functionality of the service.

Amazon Kendra supports service roles.

Choosing an IAM role in Amazon Kendra

When you create an index, call the `BatchPutDocument` operation, create a data source or create an FAQ, you must provide an access role Amazon Resource Name (ARN) that Amazon Kendra uses to access the required resources on your behalf. If you have previously created a role, then the Amazon Kendra console provides you with a list of roles to choose from. It's important to choose a role that allows access to the resources that you require. For more information, see [IAM access roles for Amazon Kendra \(p. 12\)](#).

Amazon Kendra Identity-based policy examples

By default, IAM users and roles don't have permission to create or modify Amazon Kendra resources. They also can't perform tasks using the AWS Management Console, AWS CLI, or AWS API. An IAM administrator must create IAM policies that grant users and roles permission to perform specific API operations on the specified resources they need. The administrator must then attach those policies to the IAM users or groups that require those permissions.

To learn how to create an IAM identity-based policy using these example JSON policy documents, see [Creating Policies on the JSON Tab](#) in the *IAM User Guide*.

Topics

- [Policy best practices \(p. 249\)](#)
- [AWS Managed \(Predefined\) Policies for Amazon Kendra \(p. 250\)](#)
- [Allow users to view their own permissions \(p. 250\)](#)
- [Accessing one Amazon Kendra index \(p. 251\)](#)
- [Tag-based policy examples \(p. 251\)](#)

Policy best practices

Identity-based policies are very powerful. They determine whether someone can create, access, or delete Amazon Kendra resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started using AWS managed policies** – To start using Amazon Kendra quickly, use AWS managed policies to give your employees the permissions they need. These policies are already available in your account and are maintained and updated by AWS. For more information, see [Get started using permissions with AWS managed policies](#) in the *IAM User Guide*.
- **Grant least privilege** – When you create custom policies, grant only the permissions required to perform a task. Start with a minimum set of permissions and grant additional permissions as

necessary. Doing so is more secure than starting with permissions that are too lenient and then trying to tighten them later. For more information, see [Grant least privilege](#) in the *IAM User Guide*.

- **Enable MFA for sensitive operations** – For extra security, require IAM users to use multi-factor authentication (MFA) to access sensitive resources or API operations. For more information, see [Using multi-factor authentication \(MFA\) in AWS](#) in the *IAM User Guide*.
- **Use policy conditions for extra security** – To the extent that it's practical, define the conditions under which your identity-based policies allow access to a resource. For example, you can write conditions to specify a range of allowable IP addresses that a request must come from. You can also write conditions to allow requests only within a specified date or time range, or to require the use of SSL or MFA. For more information, see [IAM JSON policy elements: Condition](#) in the *IAM User Guide*.

AWS Managed (Predefined) Policies for Amazon Kendra

AWS addresses many common use cases by providing standalone IAM policies that are created and administered by AWS. These policies are called AWS managed policies. AWS managed policies make it easier for you to assign permissions to users, groups, and roles than if you had to write the policies yourself. For more information, see [Adding Permissions to a User](#) in the *IAM User Guide*.

The following AWS managed policies, which you can attach to groups and roles in your account, are specific to Amazon Kendra:

- **AmazonKendraReadOnly** — Grants read-only access to Amazon Kendra resources.
- **AmazonKendraFullAccess** — Grants full access to create, read, update, delete, tag, and run all Amazon Kendra resources.

For the console, your role must also have `iam:CreateRole`, `iam:CreatePolicy`, `iam:AttachRolePolicy`, and `s3>ListBucket` permissions.

Note

You can review these permissions by signing in to the IAM console and searching for specific policies.

You can also create your own custom policies to allow permissions for Amazon Kendra API actions. You can attach these custom policies to the IAM roles or groups that require those permissions. For examples of IAM policies for Amazon Kendra, see [Amazon Kendra Identity-based policy examples \(p. 249\)](#).

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "ViewOwnUserInfo",  
            "Effect": "Allow",  
            "Action": [  
                "iam:GetUserPolicy",  
                "iam>ListGroupsForUser",  
                "iam>ListAttachedUserPolicies",  
                "iam>ListUserPolicies",  
                "iam GetUser"  
            ],  
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]  
        },  
    ]  
}
```

```
{
    "Sid": "NavigateInConsole",
    "Effect": "Allow",
    "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam>ListAttachedGroupPolicies",
        "iam>ListGroupPolicies",
        "iam>ListPolicyVersions",
        "iam>ListPolicies",
        "iam>ListUsers"
    ],
    "Resource": "*"
}
]
```

Accessing one Amazon Kendra index

In this example, you want to grant an IAM user in your AWS account access to query an index.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "QueryIndex",
            "Effect": "Allow",
            "Action": [
                "kendra:Query"
            ],
            "Resource": "arn:aws:kendra:${Region}:${Account}:index/${Index ID}"
        }
    ]
}
```

Tag-based policy examples

Tag-based policies are JSON policy documents that specify the actions that a principal can perform on tagged resources.

Example: Use a tag to access a resource

This example policy grants an IAM user or role in your AWS account permission to use the `Query` operation with any resource tagged with the key `department` and the value `finance`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kendra:Query"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/department": "finance"
                }
            }
        }
    ]
}
```

```
    ]  
}
```

Example: Use a tag to enable Amazon Kendra operations

This example policy grants an IAM user or role in your AWS account permission to use any Amazon Kendra operation except TagResource operation with any resource tagged with the key **department** and the value **finance**.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "kendra:*",  
            "Resource": "*"  
        },  
        {  
            "Effect": "Deny",  
            "Action": [  
                "kendra:TagResource"  
            ],  
            "Resource": "*",  
            "Condition": {  
                "StringEquals": {  
                    "aws:ResourceTag/department": "finance"  
                }  
            }  
        }  
    ]  
}
```

Example: Use a tag to restrict access to an operation

This example policy restricts access for an IAM user or role in your AWS account to use the `CreateIndex` operation unless the user provides the **department** tag and it has the allowed values **finance** and **IT**.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "kendra:CreateIndex",  
            "Resource": "*"  
        },  
        {  
            "Effect": "Deny",  
            "Action": "kendra:CreateIndex",  
            "Resource": "*",  
            "Condition": {  
                "Null": {  
                    "aws:RequestTag/department": "true"  
                }  
            }  
        },  
        {  
            "Effect": "Deny",  
            "Action": "kendra:CreateIndex",  
            "Resource": "*",  
            "Condition": {  
                "ForAnyValue:StringNotEquals": {  
                    "aws:RequestTag/department": [  
                        "finance",  
                        "IT"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

```
        "finance",
        "IT"
    ]
}
]
```

AWS managed policies for Amazon Kendra

To add permissions to users, groups, and roles, it is easier to use AWS managed policies than to write policies yourself. It takes time and expertise to [create IAM customer managed policies](#) that provide your team with only the permissions they need. To get started quickly, you can use our AWS managed policies. These policies cover common use cases and are available in your AWS account. For more information about AWS managed policies, see [AWS managed policies](#) in the *IAM User Guide*.

AWS services maintain and update AWS managed policies. You can't change the permissions in AWS managed policies. Services occasionally add additional permissions to an AWS managed policy to support new features. This type of update affects all identities (users, groups, and roles) where the policy is attached. Services are most likely to update an AWS managed policy when a new feature is launched or when new operations become available. Services do not remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

Additionally, AWS supports managed policies for job functions that span multiple services. For example, the **ReadOnlyAccess** AWS managed policy provides read-only access to all AWS services and resources. When a service launches a new feature, AWS adds read-only permissions for new operations and resources. For a list and descriptions of job function policies, see [AWS managed policies for job functions](#) in the *IAM User Guide*.

AWS managed policy: AmazonKendraReadOnly

Grants read-only access to Amazon Kendra resources. This policy includes the following permissions.

- **kendra** – Allows users to perform actions that return either a list of items or details about an item. This includes API operations that start with `Describe`, `List`, `Query`, or `GetQuerySuggestions`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "kendra:Describe*",
                "kendra>List*",
                "kendra:Query",
                "kendra:GetQuerySuggestions"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

}

AWS managed policy: AmazonKendraFullAccess

Grants full access to create, read, update, delete, tag, and run all Amazon Kendra resources. This policy includes the following permissions.

- **kendra** – Allows principals read and write access to all actions in the Amazon Kendra.
- **s3** – Allows principals get Amazon S3 bucket locations and list buckets.
- **iam** – Allows principals to pass and list roles.
- **kms** – Allows principals to describe and list AWS KMS keys and aliases.
- **secretsmanager** – Allows principals to create, describe, and list secrets.
- **ec2** – Allows principals to describe security groups, VCPs (Virtual Private Cloud), and subnets.
- **cloudwatch** – Allows principals to view Cloud Watch metrics.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "kendra.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam>ListRoles"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2>DescribeSecurityGroups",
                "ec2>DescribeVpcs",
                "ec2>DescribeSubnets"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms>ListKeys",
                "kms>ListAliases",
                "kms>DescribeKey"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3>ListAllMyBuckets",
                "s3>GetBucketLocation"
            ],
            "Resource": "*"
        }
    ]
}
```

```

        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "secretsmanager>ListSecrets"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "cloudwatch:GetMetricData"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "secretsmanager>CreateSecret",
            "secretsmanager>DescribeSecret"
        ],
        "Resource": "arn:aws:secretsmanager:*::secret:AmazonKendra-*"
    },
    {
        "Effect": "Allow",
        "Action": "kendra:*",
        "Resource": "*"
    }
]
}

```

Amazon Kendra updates to AWS managed policies

View details about updates to AWS managed policies for Amazon Kendra since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the [Amazon Kendra Document history](#) page.

Change	Description	Date
AmazonKendraReadOnly – Add permission to support GetQuerySuggestions operation	Amazon Kendra added a new action <code>GetQuerySuggestions</code> that allows access to get query suggestions for popular search queries, helping guide your users' search. When users type their search query, the suggested query helps autocomplete their search. This action is associated with the <code>GetQuerySuggestions</code> API operation.	May 27, 2021
Amazon Kendra started tracking changes	Amazon Kendra started tracking changes for its AWS managed policies.	May 27, 2021

Troubleshooting Amazon Kendra Identity and Access

Use the following information to help you diagnose and fix common issues that you might encounter when working with Amazon Kendra and IAM.

Topics

- [I am not authorized to perform an action in Amazon Kendra \(p. 256\)](#)
- [I am not authorized to perform iam:PassRole \(p. 256\)](#)
- [I want to view my access keys \(p. 256\)](#)
- [I'm an administrator and I want to allow others to access Amazon Kendra \(p. 257\)](#)
- [I want to allow people outside of my AWS account to access my Amazon Kendra resources \(p. 257\)](#)

I am not authorized to perform an action in Amazon Kendra

If the AWS Management Console tells you that you're not authorized to perform an action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about an index but does not have `kendra:DescribeIndex` permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:  
kendra:DescribeIndex on resource: index ARN
```

In this case, Mateo asks his administrator to update his policies to allow him to access the `index` resource using the `kendra:DescribeIndex` action.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password. Ask that person to update your policies to allow you to pass a role to Amazon Kendra.

Some AWS services allow you to pass an existing role to that service, instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in Amazon Kendra. However, the action requires the service to have permissions granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform: iam:PassRole
```

In this case, Mary asks her administrator to update her policies to allow her to perform the `iam:PassRole` action.

I want to view my access keys

After you create your IAM user access keys, you can view your access key ID at any time. However, you can't view your secret access key again. If you lose your secret key, you must create a new access key pair.

Access keys consist of two parts: an access key ID (for example, `AKIAIOSFODNN7EXAMPLE`) and a secret access key (for example, `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`). Like a user name and

password, you must use both the access key ID and secret access key together to authenticate your requests. Manage your access keys as securely as you do your user name and password.

Important

Do not provide your access keys to a third party, even to help [find your canonical user ID](#). By doing this, you might give someone permanent access to your account.

When you create an access key pair, you are prompted to save the access key ID and secret access key in a secure location. The secret access key is available only at the time you create it. If you lose your secret access key, you must add new access keys to your IAM user. You can have a maximum of two access keys. If you already have two, you must delete one key pair before creating a new one. To view instructions, see [Managing access keys](#) in the *IAM User Guide*.

I'm an administrator and I want to allow others to access Amazon Kendra

To allow others to access Amazon Kendra, you must create an IAM entity (user or role) for the person or application that needs access. They will use the credentials for that entity to access AWS. You must then attach a policy to the entity that grants them the correct permissions in Amazon Kendra.

To get started right away, see [Creating your first IAM delegated user and group](#) in the *IAM User Guide*.

I want to allow people outside of my AWS account to access my Amazon Kendra resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether Amazon Kendra supports these features, see [How Amazon Kendra works with IAM \(p. 246\)](#).
- To learn how to provide access to your resources across AWS accounts that you own, see [Providing access to an IAM user in another AWS account that you own](#) in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see [Providing access to AWS accounts owned by third parties](#) in the *IAM User Guide*.
- To learn how to provide access through identity federation, see [Providing access to externally authenticated users \(identity federation\)](#) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see [How IAM roles differ from resource-based policies](#) in the *IAM User Guide*.

Logging and monitoring in Amazon Kendra

Monitoring is an important part of maintaining the reliability, availability, and performance of your Amazon Kendra applications. To monitor Amazon Kendra API calls, you can use AWS CloudTrail. To monitor the status of your jobs, use Amazon CloudWatch Logs.

- **Amazon CloudWatch Alarms** — Using CloudWatch alarms, you watch a single metric over a time period that you specify. If the metric exceeds a policy, CloudWatch alarms do not invoke actions when a metric is in a particular state. Rather the state must have changed and been maintained for a specified number of periods. For more information, see [Monitoring Amazon Kendra with Amazon CloudWatch \(p. 231\)](#).

- **AWS CloudTrail Logs** — CloudTrail provides a record of actions taken by a user, role, or an AWS service in Amazon Kendra. Using the information collected by CloudTrail, you can determine the request that was made to Amazon Kendra, the IP address from which the request was made, who made the request, when it was made, and additional details. For more information, see [Logging Amazon Kendra API calls with AWS CloudTrail logs \(p. 229\)](#).

Compliance validation for Amazon Kendra

Third-party auditors assess the security and compliance of Amazon Kendra as part of multiple Amazon Kendra compliance programs. Amazon Kendra is HIPAA compliant, SOC 2 compliant, and IRAP compliant.

For a list of AWS services in scope of specific compliance programs, see [AWS Services in Scope by Compliance Program](#). For general information, see [AWS Compliance Programs](#).

You can download third-party audit reports using AWS Artifact. For more information, see [Downloading Reports in AWS Artifact](#).

Your compliance responsibility when using Amazon Kendra is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- [Security and Compliance Quick Start Guides](#) – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- [Architecting for HIPAA Security and Compliance Whitepaper](#) – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- [AWS Compliance Resources](#) – This collection of workbooks and guides might apply to your industry and location.
- [Evaluating Resources with Rules](#) in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- [AWS Security Hub](#) – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

Resilience in Amazon Kendra

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see [AWS Global Infrastructure](#).

With AWS global infrastructure, Amazon Kendra Enterprise Edition is fault tolerant, scalable, and highly available. Rolling back to previous versions of an index is not currently supported, but you can refresh or recreate portions of your index by [deleting](#) and [adding](#) existing data sources back into your index.

Infrastructure security in Amazon Kendra

As a managed service, Amazon Kendra is protected by the AWS global network security procedures that are described in the [Amazon Web Services: Overview of Security Processes](#) whitepaper.

You use AWS published API calls to access Amazon Kendra through the network. Clients must support Transport Layer Security (TLS) 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the [AWS Security Token Service](#) (AWS STS) to generate temporary security credentials to sign requests.

Quotas for Amazon Kendra

Supported regions

For a list of AWS Regions where Amazon Kendra is available, see [AWS Regions and Endpoints](#) in the *Amazon Web Services General Reference*.

Quotas

Service quotas, also referred to as limits, are the maximum number of service resources for your AWS account. For more information, see [AWS Service Quotas](#) in the *AWS General Reference*.

Description	Default	Edition	Adjustable
Maximum number of indexes per account	5	Developer	Yes
Maximum number of indexes per account	5	Enterprise	Yes
Maximum number of data sources per index	5	Developer	No
Maximum number of data sources per index	50	Enterprise	Yes
Maximum amount of text extracted for an index or a single unit of storage capacity	3 GB	Developer	Yes
Maximum amount of text extracted for an index or a single unit of storage capacity	30 GB	Enterprise	Yes
Maximum number of documents for an index or a single unit of storage capacity. You must also not exceed the maximum text extracted.	10,000	Developer	Yes
Maximum number of documents for an index or a single unit of storage capacity. You must also not exceed the maximum text extracted.	100,000	Enterprise	Yes

Description	Default	Edition	Adjustable
Maximum number of queries per second for an index or a single unit of query capacity	0.05	Developer	Yes
Maximum number of queries per second for an index or a single unit of query capacity	0.1	Enterprise	Yes
Maximum number of storage capacity units per index	100	Enterprise	Yes
Maximum number of query capacity units per index	100	Enterprise	Yes
Maximum number of search results retrieved for a query	100	All editions	Yes
Maximum size of a single document	50 MB	All editions	Yes
Maximum amount of text extracted from a single document	5 MB	All editions	Yes
Maximum user group list size per query attribute	10	All editions	Yes
Maximum string list size per query attribute	10	All editions	Yes
Maximum number of custom attributes per index	500	All editions	No
Maximum number of FAQs per index	30	All editions	Yes
Maximum size of 1 FAQ	1 MB	All editions	Yes
Maximum number of results returned for FAQ	4	All editions	Yes
Maximum number of thesauri per index	1	All editions	No
Maximum size of a thesaurus file	5 MB	All editions	Yes
Maximum number of synonym rules per thesaurus	10,000	All editions	Yes

Description	Default	Edition	Adjustable
Maximum number of synonyms per term in all thesauri in an index	10	All editions	No
Maximum number of block lists per index	1	All editions	No
Maximum size of a block list text file	2 MB	All editions	Yes
Maximum number of items (words or phrases) in a block list	20,000	All editions	Yes

For more information about Amazon Kendra service quotas and to request a quota increase, see [Service Quotas](#).

Troubleshooting

This section can help you solve common problems you might find when working with Amazon Kendra.

Topics

- [Troubleshooting data sources \(p. 263\)](#)
- [Troubleshooting document search results \(p. 264\)](#)
- [Troubleshooting general issues \(p. 265\)](#)

Troubleshooting data sources

This section can help you fix issues with Amazon Kendra data sources.

My documents were not indexed

When you synchronize your Kendra index with a data source, you may run into issues that prevent the documents from being indexed. Indexing is a two step process. First, there is the data source level process of crawling the data source to find the new and updated documents to index, and to find any documents to remove from the index. Second, there is the document level process where each document is accessed and indexed.

An error can occur in either of these steps. Data source level errors are reported in the console in the **Sync run history** section of the data source details page. The status of the synchronization job can be **Succeeded**, **Incomplete**, or **Failed**. You can also see the number of documents indexed and deleted during the job. If the status is **Failed**, a message is shown in the **Details** column.

Document level errors are reported in Amazon CloudWatch Logs. You can see the errors using the CloudWatch console.

My synchronization job failed

A synchronization job typically fails when there is a configuration error in the index or the data source. The error message in the Details column of the data source gives information about what went wrong. The problem is usually that the index or the data source does not have the proper IAM permissions. The error message describes the permissions that are missing. Here are some of the error messages that you can receive:

Failed to create log group for job. Please make sure that the IAM role provided has sufficient permissions.

If your index role does not have permission to use CloudWatch, the data source will not be able to create a CloudWatch log. If you get this error you need to add CloudWatch permissions to the index role.

Failed to access S3 file prefix (`bucket name`) while trying to crawl your metadata files. Please make sure the IAM Role (`role ARN`) provided has sufficient permissions.

When you are using an Amazon S3 data source, Kendra must have permission to access the bucket that contains the documents. You need to add permission for Kendra to read the bucket to the data source IAM role.

The provided IAM Role (`role ARN`) could not be assumed. Please make sure Amazon Kendra is a trusted entity that is allowed to assume the role.

Kendra needs permission to assume the index and data source IAM roles. You need to add a trust policy to the roles with permission for the `sts:AssumeRole` action.

For the IAM policies that Kendra needs to index a data source, see [IAM access roles for Amazon Kendra \(p. 12\)](#).

My synchronization job is incomplete

Jobs are generally incomplete when they have completed the data source level process but have had some error during the document level process. When a job is incomplete, some of the documents may have been successfully indexed. For an Amazon S3 data source, an incomplete job is typically caused by:

- The metadata for one or more documents was invalid.
- When there are documents to submit for indexing, at least one document was not submitted.
- When there are documents to submit for deleting from the index, at least one document was not submitted.

To troubleshoot an incomplete synchronization job, look first to your CloudWatch logs.

1. From the details column, choose **View details in CloudWatch**.
2. Review the error messages to see what caused the document to fail.

My synchronization job succeeded but there are no indexed documents

Occasionally you will have a index synchronization job run that is marked as **Succeeded** but there are no new or updated documents indexed when you expect there to be. Here are some reasons:

- Check CloudWatch `DocumentsSubmittedForIndexingFailed` metric to see if there were any documents that failed to synchronize. Check your CloudWatch logs for details.
- For an Amazon S3 data source, you may have given Kendra the wrong bucket name or prefix. Make sure that the bucket that Kendra is using is the one that contains the documents to index.
- If you are re-indexing a document that failed to be indexed in an earlier job, Kendra won't index it unless you make a change to the document or its associated metadata file.

Troubleshooting document search results

This section can help you fix issues in your Amazon Kendra search results.

Why do I only see 100 results?

Amazon Kendra returns the total count of related documents. The top 100 are returned. The results are paginated. You can use `PageNumber` to access different pages.

Pagination stops at 100 documents even if there are more than 100 search results.

Why are documents I expect to see missing?

Amazon Kendra supports access control lists (ACLs) based on user and groups. Amazon Kendra ingests ACL policies via connectors. If an index does not configure an ACL, only documents matching the

attribute filter for user and group will be shown. If a user or group attribute filter is provided, documents without an ACL will not be shown.

If you are using token-based access control, documents without an ACL policy and documents that match the user and groups will be shown.

Why do I see documents that have an ACL policy?

If an index does not configure an access control policy, then user and groups can be provided by the filter. If no user and group filter is applied then all related documents will be returned. Any ACL policy will be ignored.

Troubleshooting general issues

Kendra uses CloudWatch metrics and logs to provide you with insight into synchronizing your data sources. You can use the metrics and logs to determine what went wrong with a synchronization run and to determine what you need to do to fix it.

For general troubleshooting, start with your CloudWatch metrics.

- Check the `DocumentsCrawled` metric to see how many documents your data source checked. For an Amazon S3 bucket, if the number is less than you expect, check to be sure that your data source is pointing to the right bucket.
- Check the `DocumentsSkippedNoChange` metric to see how many documents were skipped because they haven't changed since the last synchronization. If the number does not match what you expect, check to make sure that your repository was updated correctly.
- Check the `DocumentsSkippedInvalidMetadata` metric to see how many documents had invalid metadata. Check your CloudWatch logs to see the specific errors that occurred.
- Check the `DocumentsSubmittedForIndexingFailed` metric to see the number of documents that were sent from the data source to the index but failed to be indexed for some reason. For example, if you use a metadata attribute in an Amazon S3 data source that hasn't been defined as a custom index field the document will not be indexed. Check your CloudWatch logs to see the specific errors that occurred.
- Check the `DocumentsSubmittedForDeletionFailed` metric to see how many documents that the data source attempted to remove from the index failed to be deleted from the index. Check your CloudWatch logs to see the specific errors that occurred.

You can look at the CloudWatch logs for a particular synchronization run to get details of the errors that occurred during the run. For more information about CloudWatch logs with Kendra, see [Monitoring Amazon Kendra with Amazon CloudWatch Logs \(p. 235\)](#).

Document history for Amazon Kendra

- **Latest documentation update:** October 13, 2021

The following table describes important changes in each release of Amazon Kendra. For notification about updates to this documentation, you can subscribe to an RSS feed.

update-history-change	update-history-description	update-history-date
Region expansion	Amazon Kendra is now available in AWS GovCloud (US-West) (us-gov-west-1).	October 13, 2021
New feature	Amazon Kendra can now index documents in multiple languages and filter search results by language. See Adding documents in languages other than English and Searching in languages .	October 7, 2021
New feature	Amazon Kendra now integrates with AWS Single Sign-On identity store to fetch access levels of groups and users for user context filtering . See User-group configuration for AWS SSO .	October 6, 2021
New tutorial	Amazon Kendra now provides a tutorial that walks you through how to build a metadata-enriched search solution. See Building an intelligent search solution .	August 13, 2021
New feature	Amazon Kendra now provides a data source connector for Amazon WorkDocs. For more information, see Using an Amazon WorkDocs data source .	July 20, 2021
New feature	Amazon Kendra now provides a web crawler to crawl and index webpages. For more information, see Using a web crawler data source .	June 17, 2021
Region expansion	Amazon Kendra is now available in Canada (Central) (ca-central-1).	June 16, 2021

Region expansion	Amazon Kendra is now available in US East (Ohio) (us-east-2).	June 7, 2021
New feature	Amazon Kendra now supports query suggestions, where users are suggested popular queries relevant to their search. For more information, see Suggesting popular search queries .	May 27, 2021
AWS managed policy updates – New policies (p. 266)	Amazon Kendra added two new AWS managed policies. For more information, see AWS Managed policies for Amazon Kendra .	May 27, 2021
Region expansion	Amazon Kendra is now available in Asia Pacific (Singapore) (ap-southeast-1).	May 5, 2021
New feature	Amazon Kendra now supports tuning search relevance in the query by overriding tuning configurations set at the index level. For more information, see Tuning search relevance and Tuning responses .	April 20, 2021
New feature	Amazon Kendra now supports OAuth 2.0 authentication and using ServiceNow queries to select documents for indexing. For more information, see Using a ServiceNow data source .	April 1, 2021
New feature	Amazon Kendra now supports incremental learning for FAQ documents. For more information, see Submitting feedback for incremental learning .	February 17, 2021
New feature	Amazon Kendra now supports index synonyms. For more information, see Adding synonyms to an index .	December 10, 2020
New feature	Amazon Kendra now provides a data base connector for Google Workspace Drive. For more information, see Using a Google Workspace Drive data source .	December 8, 2020
New feature	Amazon Kendra now provides a JavaScript library that makes it easier for you to provide query feedback to Amazon Kendra. For more information, see Submitting feedback .	December 8, 2020

New feature	Amazon Kendra now supports token-based user access control. For more information, see Controlling access to documents in an index .	November 5, 2020
New feature	The Amazon Kendra Confluence data source connector now works with Confluence cloud. For more information, see Using a Confluence data source .	November 5, 2020
New feature	Amazon Kendra is now HIPAA compliant. For more information, see Compliance validation for Amazon Kendra .	November 5, 2020
Region expansion	Amazon Kendra is now available in Asia Pacific (Sydney) (ap-southeast-2).	November 2, 2020
New feature	Amazon Kendra now provides a data source connector for Confluence server. For more information, see Using a Confluence data source .	October 26, 2020
New feature	Amazon Kendra now provides a data source that you can use to generate statistics for your custom connectors. For more information, see Using a custom data source .	October 21, 2020
New feature	Amazon Kendra now supports custom attributes for frequently asked questions. For more information, see Adding questions and answers .	September 17, 2020
New feature	Amazon Kendra now returns confidence scores for query results. For more information, see QueryResultItem .	September 15, 2020
New feature	AWS CloudFormation now supports Amazon Kendra. For more information, see Amazon Kendra resource type reference - AWS CloudFormation .	September 10, 2020
New feature	Amazon Kendra adds support for AWS PrivateLink. For more information, see Amazon Kendra and interface VPC endpoints (AWS PrivateLink) .	July 7, 2020
New guide	This is the first release of the <i>Amazon Kendra Developer Guide</i> .	May 11, 2020

API Reference

This section contains the API Reference documentation.

Actions

The following actions are supported:

- [BatchDeleteDocument \(p. 271\)](#)
- [BatchGetDocumentStatus \(p. 274\)](#)
- [BatchPutDocument \(p. 277\)](#)
- [ClearQuerySuggestions \(p. 281\)](#)
- [CreateDataSource \(p. 283\)](#)
- [CreateFaq \(p. 292\)](#)
- [CreateIndex \(p. 296\)](#)
- [CreateQuerySuggestionsBlockList \(p. 301\)](#)
- [CreateThesaurus \(p. 305\)](#)
- [DeleteDataSource \(p. 309\)](#)
- [DeleteFaq \(p. 311\)](#)
- [DeleteIndex \(p. 313\)](#)
- [DeletePrincipalMapping \(p. 315\)](#)
- [DeleteQuerySuggestionsBlockList \(p. 318\)](#)
- [DeleteThesaurus \(p. 320\)](#)
- [DescribeDataSource \(p. 322\)](#)
- [DescribeFaq \(p. 331\)](#)
- [DescribeIndex \(p. 335\)](#)
- [DescribePrincipalMapping \(p. 340\)](#)
- [DescribeQuerySuggestionsBlockList \(p. 343\)](#)
- [DescribeQuerySuggestionsConfig \(p. 347\)](#)
- [DescribeThesaurus \(p. 350\)](#)
- [GetQuerySuggestions \(p. 354\)](#)
- [ListDataSources \(p. 357\)](#)
- [ListDataSourceSyncJobs \(p. 360\)](#)
- [ListFaqs \(p. 364\)](#)
- [ListGroupsOlderThanOrderingId \(p. 367\)](#)
- [ListIndices \(p. 370\)](#)
- [ListQuerySuggestionsBlockLists \(p. 372\)](#)
- [ListTagsForResource \(p. 375\)](#)
- [ListThesauri \(p. 377\)](#)
- [PutPrincipalMapping \(p. 380\)](#)
- [Query \(p. 384\)](#)
- [StartDataSourceSyncJob \(p. 392\)](#)
- [StopDataSourceSyncJob \(p. 394\)](#)

- [SubmitFeedback \(p. 396\)](#)
- [TagResource \(p. 399\)](#)
- [UntagResource \(p. 401\)](#)
- [UpdateDataSource \(p. 403\)](#)
- [UpdateIndex \(p. 411\)](#)
- [UpdateQuerySuggestionsBlockList \(p. 415\)](#)
- [UpdateQuerySuggestionsConfig \(p. 418\)](#)
- [UpdateThesaurus \(p. 421\)](#)

BatchDeleteDocument

Removes one or more documents from an index. The documents must have been added with the [BatchPutDocument](#) operation.

The documents are deleted asynchronously. You can see the progress of the deletion by using AWS CloudWatch. Any error messages related to the processing of the batch are sent to you CloudWatch log.

Request Syntax

```
{  
    "DataSourceSyncJobMetricTarget": {  
        "DataSourceId": "string",  
        "DataSourceSyncJobId": "string"  
    },  
    "DocumentIdList": [ "string" ],  
    "IndexId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

[DataSourceSyncJobMetricTarget \(p. 271\)](#)

Maps a particular data source sync job to a particular data source.

Type: [DataSourceSyncJobMetricTarget \(p. 466\)](#) object

Required: No

[DocumentIdList \(p. 271\)](#)

One or more identifiers for documents to delete from the index.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 10 items.

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

[IndexId \(p. 271\)](#)

The identifier of the index that contains the documents to delete.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

Response Syntax

```
{
```

```
"FailedDocuments": [  
    {  
        "ErrorCode": "string",  
        "ErrorMessage": "string",  
        "Id": "string"  
    }  
]
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

FailedDocuments ([p. 271](#))

A list of documents that could not be removed from the index. Each entry contains an error message that indicates why the document couldn't be removed from the index.

Type: Array of [BatchDeleteDocumentResponseFailedDocument](#) ([p. 434](#)) objects

Errors

For information about the errors that are common to all actions, see [Common Errors](#) ([p. 563](#)).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

BatchGetDocumentStatus

Returns the indexing status for one or more documents submitted with the [BatchPutDocument](#) operation.

When you use the `BatchPutDocument` operation, documents are indexed asynchronously. You can use the `BatchGetDocumentStatus` operation to get the current status of a list of documents so that you can determine if they have been successfully indexed.

You can also use the `BatchGetDocumentStatus` operation to check the status of the [BatchDeleteDocument](#) operation. When a document is deleted from the index, Amazon Kendra returns `NOT_FOUND` as the status.

Request Syntax

```
{  
    "DocumentInfoList": [  
        {  
            "Attributes": [  
                {  
                    "Key": "string",  
                    "Value": {  
                        "DateValue": number,  
                        "LongValue": number,  
                        "StringListValue": [ "string" ],  
                        "StringValue": "string"  
                    }  
                }  
            ],  
            "DocumentId": "string"  
        }  
    ],  
    "IndexId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

[DocumentInfoList \(p. 274\)](#)

A list of `DocumentInfo` objects that identify the documents for which to get the status. You identify the documents by their document ID and optional attributes.

Type: Array of [DocumentInfo \(p. 474\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 10 items.

Required: Yes

[IndexId \(p. 274\)](#)

The identifier of the index to add documents to. The index ID is returned by the [CreateIndex](#) operation.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

Response Syntax

```
{  
    "DocumentStatusList": [  
        {  
            "DocumentId": "string",  
            "DocumentStatus": "string",  
            "FailureCode": "string",  
            "FailureReason": "string"  
        }  
    ],  
    "Errors": [  
        {  
            "DocumentId": "string",  
            "ErrorCode": "string",  
            "ErrorMessage": "string"  
        }  
    ]  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

DocumentStatusList ([p. 275](#))

The status of documents. The status indicates if the document is waiting to be indexed, is in the process of indexing, has completed indexing, or failed indexing. If a document failed indexing, the status provides the reason why.

Type: Array of [Status](#) ([p. 543](#)) objects

Errors ([p. 275](#))

A list of documents that Amazon Kendra couldn't get the status for. The list includes the ID of the document and the reason that the status couldn't be found.

Type: Array of [BatchGetDocumentStatusResponseError](#) ([p. 435](#)) objects

Errors

For information about the errors that are common to all actions, see [Common Errors](#) ([p. 563](#)).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerException

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

BatchPutDocument

Adds one or more documents to an index.

The `BatchPutDocument` operation enables you to ingest inline documents or a set of documents stored in an Amazon S3 bucket. Use this operation to ingest your text and unstructured text into an index, add custom attributes to the documents, and to attach an access control list to the documents added to the index.

The documents are indexed asynchronously. You can see the progress of the batch using AWS CloudWatch. Any error messages related to processing the batch are sent to your AWS CloudWatch log.

Request Syntax

```
{  
    "Documents": [  
        {  
            "AccessControlList": [  
                {  
                    "Access": "string",  
                    "DataSourceId": "string",  
                    "Name": "string",  
                    "Type": "string"  
                }  
            ],  
            "Attributes": [  
                {  
                    "Key": "string",  
                    "Value": {  
                        "DateValue": number,  
                        "LongValue": number,  
                        "StringListValue": [ "string" ],  
                        "StringValue": "string"  
                    }  
                }  
            ],  
            "Blob": blob,  
            "ContentType": "string",  
            "HierarchicalAccessControlList": [  
                {  
                    "PrincipalList": [  
                        {  
                            "Access": "string",  
                            "DataSourceId": "string",  
                            "Name": "string",  
                            "Type": "string"  
                        }  
                    ]  
                }  
            ],  
            "Id": "string",  
            "S3Path": {  
                "Bucket": "string",  
                "Key": "string"  
            },  
            "Title": "string"  
        }  
    ],  
    "IndexId": "string",  
    "RoleArn": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

[Documents \(p. 277\)](#)

One or more documents to add to the index.

Documents can include custom attributes. For example, 'DataSourceId' and 'DataSourceSyncJobId' are custom attributes that provide information on the synchronization of documents running on a data source. Note, 'DataSourceSyncJobId' could be an optional custom attribute as Amazon Kendra will use the ID of a running sync job.

Documents have the following file size limits.

- 5 MB total size for inline documents
- 50 MB total size for files from an S3 bucket
- 5 MB extracted text for any file

For more information about file size and transaction per second quotas, see [Quotas](#).

Type: Array of [Document \(p. 469\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 10 items.

Required: Yes

[IndexId \(p. 277\)](#)

The identifier of the index to add the documents to. You need to create the index first using the [CreateIndex](#) operation.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

[RoleArn \(p. 277\)](#)

The Amazon Resource Name (ARN) of a role that is allowed to run the [BatchPutDocument](#) operation. For more information, see [IAM Roles for Amazon Kendra](#).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: arn:[a-zA-Z0-9-\.\-]{1,63}:[a-zA-Z0-9-\.\-]{0,63}:[a-zA-Z0-9-\.\-]{0,63}:[a-zA-Z0-9-\.\-]{0,63}:[^/]{0,1023}

Required: No

Response Syntax

```
{
```

```
"FailedDocuments": [  
    {  
        "ErrorCode": "string",  
        "ErrorMessage": "string",  
        "Id": "string"  
    }  
]
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

FailedDocuments (p. 278)

A list of documents that were not added to the index because the document failed a validation check. Each document contains an error message that indicates why the document couldn't be added to the index.

If there was an error adding a document to an index the error is reported in your AWS CloudWatch log. For more information, see [Monitoring Amazon Kendra with Amazon CloudWatch Logs](#)

Type: Array of [BatchPutDocumentResponseFailedDocument](#) (p. 436) objects

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ServiceQuotaExceededException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ClearQuerySuggestions

Clears existing query suggestions from an index.

This deletes existing suggestions only, not the queries in the query log. After you clear suggestions, Amazon Kendra learns new suggestions based on new queries added to the query log from the time you cleared suggestions. If you do not see any new suggestions, then please allow Amazon Kendra to collect enough queries to learn new suggestions.

`ClearQuerySuggestions` is currently not supported in the AWS GovCloud (US-West) region.

Request Syntax

```
{  
    "IndexId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

IndexId (p. 281)

The identifier of the index you want to clear query suggestions from.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

CreateDataSource

Creates a data source that you want to use with an Amazon Kendra index.

You specify a name, data source connector type and description for your data source. You also specify configuration information for the data source connector.

`CreateDataSource` is a synchronous operation. The operation returns 200 if the data source was successfully created. Otherwise, an exception is raised.

Amazon S3 and [custom](#) data sources are the only supported data sources in the AWS GovCloud (US-West) region.

Request Syntax

```
{
    "ClientToken": "string",
    "Configuration": {
        "ConfluenceConfiguration": {
            "AttachmentConfiguration": {
                "AttachmentFieldMappings": [
                    {
                        "DataSourceFieldName": "string",
                        "DateFormat": "string",
                        "IndexFieldName": "string"
                    }
                ],
                "CrawlAttachments": boolean
            },
            "BlogConfiguration": {
                "BlogFieldMappings": [
                    {
                        "DataSourceFieldName": "string",
                        "DateFormat": "string",
                        "IndexFieldName": "string"
                    }
                ]
            },
            "ExclusionPatterns": [ "string" ],
            "InclusionPatterns": [ "string" ],
            "PageConfiguration": {
                "PageFieldMappings": [
                    {
                        "DataSourceFieldName": "string",
                        "DateFormat": "string",
                        "IndexFieldName": "string"
                    }
                ]
            },
            "SecretArn": "string",
            "ServerUrl": "string",
            "SpaceConfiguration": {
                "CrawlArchivedSpaces": boolean,
                "CrawlPersonalSpaces": boolean,
                "ExcludeSpaces": [ "string" ],
                "IncludeSpaces": [ "string" ],
                "SpaceFieldMappings": [
                    {
                        "DataSourceFieldName": "string",
                        "DateFormat": "string",
                        "IndexFieldName": "string"
                    }
                ]
            }
        }
    }
}
```

```

},
"Version": "string",
"VpcConfiguration": {
    "SecurityGroupIds": [ "string" ],
    "SubnetIds": [ "string" ]
}
},
"DatabaseConfiguration": {
    "AclConfiguration": {
        "AllowedGroupsColumnName": "string"
    },
    "ColumnConfiguration": {
        "ChangeDetectingColumns": [ "string" ],
        "DocumentDataColumnName": "string",
        "DocumentIdColumnName": "string",
        "DocumentTitleColumnName": "string",
        "FieldMappings": [
            {
                "DataSourceFieldName": "string",
                "DateFormat": "string",
                "IndexFieldName": "string"
            }
        ]
    },
    "ConnectionConfiguration": {
        "DatabaseHost": "string",
        "DatabaseName": "string",
        "DatabasePort": number,
        "SecretArn": "string",
        "TableName": "string"
    },
    "DatabaseEngineType": "string",
    "SqlConfiguration": {
        "QueryIdentifiersEnclosingOption": "string"
    },
    "VpcConfiguration": {
        "SecurityGroupIds": [ "string" ],
        "SubnetIds": [ "string" ]
    }
},
"GoogleDriveConfiguration": {
    "ExcludeMimeTypes": [ "string" ],
    "ExcludeSharedDrives": [ "string" ],
    "ExcludeUserAccounts": [ "string" ],
    "ExclusionPatterns": [ "string" ],
    "FieldMappings": [
        {
            "DataSourceFieldName": "string",
            "DateFormat": "string",
            "IndexFieldName": "string"
        }
    ],
    "InclusionPatterns": [ "string" ],
    "SecretArn": "string"
},
"OneDriveConfiguration": {
    "DisableLocalGroups": boolean,
    "ExclusionPatterns": [ "string" ],
    "FieldMappings": [
        {
            "DataSourceFieldName": "string",
            "DateFormat": "string",
            "IndexFieldName": "string"
        }
    ],
    "InclusionPatterns": [ "string" ],
}

```

```

    "OneDriveUsers": {
        "OneDriveUserList": [ "string" ],
        "OneDriveUserS3Path": {
            "Bucket": "string",
            "Key": "string"
        }
    },
    "SecretArn": "string",
    "TenantDomain": "string"
},
"S3Configuration": {
    "AccessControlListConfiguration": {
        "KeyPath": "string"
    },
    "BucketName": "string",
    "DocumentsMetadataConfiguration": {
        "S3Prefix": "string"
    },
    "ExclusionPatterns": [ "string" ],
    "InclusionPatterns": [ "string" ],
    "InclusionPrefixes": [ "string" ]
},
"SalesforceConfiguration": {
    "ChatterFeedConfiguration": {
        "DocumentDataFieldName": "string",
        "DocumentTitleFieldName": "string",
        "FieldMappings": [
            {
                "DataSourceFieldName": "string",
                "DateFormat": "string",
                "IndexFieldName": "string"
            }
        ],
        "IncludeFilterTypes": [ "string" ]
    },
    "CrawlAttachments": boolean,
    "ExcludeAttachmentFilePatterns": [ "string" ],
    "IncludeAttachmentFilePatterns": [ "string" ],
    "KnowledgeArticleConfiguration": {
        "CustomKnowledgeArticleTypeConfigurations": [
            {
                "DocumentDataFieldName": "string",
                "DocumentTitleFieldName": "string",
                "FieldMappings": [
                    {
                        "DataSourceFieldName": "string",
                        "DateFormat": "string",
                        "IndexFieldName": "string"
                    }
                ],
                "Name": "string"
            }
        ],
        "IncludedStates": [ "string" ],
        "StandardKnowledgeArticleTypeConfiguration": {
            "DocumentDataFieldName": "string",
            "DocumentTitleFieldName": "string",
            "FieldMappings": [
                {
                    "DataSourceFieldName": "string",
                    "DateFormat": "string",
                    "IndexFieldName": "string"
                }
            ]
        }
    }
},
]
}

```

```

"SecretArn": "string",
"ServerUrl": "string",
"StandardObjectAttachmentConfiguration": {
    "DocumentTitleFieldName": "string",
    "FieldMappings": [
        {
            "DataSourceFieldName": "string",
            "DateFormat": "string",
            "IndexFieldName": "string"
        }
    ]
},
"StandardObjectConfigurations": [
    {
        "DocumentDataFieldName": "string",
        "DocumentTitleFieldName": "string",
        "FieldMappings": [
            {
                "DataSourceFieldName": "string",
                "DateFormat": "string",
                "IndexFieldName": "string"
            }
        ],
        "Name": "string"
    }
]
},
"ServiceNowConfiguration": {
    "AuthenticationType": "string",
    "HostUrl": "string",
    "KnowledgeArticleConfiguration": {
        "CrawlAttachments": boolean,
        "DocumentDataFieldName": "string",
        "DocumentTitleFieldName": "string",
        "ExcludeAttachmentFilePatterns": [ "string" ],
        "FieldMappings": [
            {
                "DataSourceFieldName": "string",
                "DateFormat": "string",
                "IndexFieldName": "string"
            }
        ],
        "FilterQuery": "string",
        "IncludeAttachmentFilePatterns": [ "string" ]
    },
    "SecretArn": "string",
    "ServiceCatalogConfiguration": {
        "CrawlAttachments": boolean,
        "DocumentDataFieldName": "string",
        "DocumentTitleFieldName": "string",
        "ExcludeAttachmentFilePatterns": [ "string" ],
        "FieldMappings": [
            {
                "DataSourceFieldName": "string",
                "DateFormat": "string",
                "IndexFieldName": "string"
            }
        ],
        "IncludeAttachmentFilePatterns": [ "string" ]
    },
    "ServiceNowBuildVersion": "string"
},
"SharePointConfiguration": {
    "CrawlAttachments": boolean,
    "DisableLocalGroups": boolean,
    "DocumentTitleFieldName": "string",

```

```

    "ExclusionPatterns": [ "string" ],
    "FieldMappings": [
        {
            "DataSourceFieldName": "string",
            "DateFormat": "string",
            "IndexFieldName": "string"
        }
    ],
    "InclusionPatterns": [ "string" ],
    "SecretArn": "string",
    "SharePointVersion": "string",
    "SslCertificateS3Path": {
        "Bucket": "string",
        "Key": "string"
    },
    "Urls": [ "string" ],
    "UseChangeLog": boolean,
    "VpcConfiguration": {
        "SecurityGroupIds": [ "string" ],
        "SubnetIds": [ "string" ]
    }
},
"WebCrawlerConfiguration": {
    "AuthenticationConfiguration": {
        "BasicAuthentication": [
            {
                "Credentials": "string",
                "Host": "string",
                "Port": number
            }
        ]
    },
    "CrawlDepth": number,
    "MaxContentSizePerPageInMegaBytes": number,
    "MaxLinksPerPage": number,
    "MaxUrlsPerMinuteCrawlRate": number,
    "ProxyConfiguration": {
        "Credentials": "string",
        "Host": "string",
        "Port": number
    },
    "UrlExclusionPatterns": [ "string" ],
    "UrlInclusionPatterns": [ "string" ],
    "Urls": [
        "SeedUrlConfiguration": {
            "SeedUrls": [ "string" ],
            "WebCrawlerMode": "string"
        },
        "SiteMapsConfiguration": {
            "SiteMaps": [ "string" ]
        }
    ]
},
"WorkDocsConfiguration": {
    "CrawlComments": boolean,
    "ExclusionPatterns": [ "string" ],
    "FieldMappings": [
        {
            "DataSourceFieldName": "string",
            "DateFormat": "string",
            "IndexFieldName": "string"
        }
    ],
    "InclusionPatterns": [ "string" ],
    "OrganizationId": "string",
    "UseChangeLog": boolean
}

```

```
        },
        "Description": "string",
        "IndexId": "string",
        "LanguageCode": "string",
        "Name": "string",
        "RoleArn": "string",
        "Schedule": "string",
        "Tags": [
            {
                "Key": "string",
                "Value": "string"
            }
        ],
        "Type": "string"
    }
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

[ClientToken \(p. 283\)](#)

A token that you provide to identify the request to create a data source. Multiple calls to the `CreateDataSource` operation with the same client token will create only one data source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Required: No

[Configuration \(p. 283\)](#)

The connector configuration information that is required to access the repository.

You can't specify the `Configuration` parameter when the `Type` parameter is set to `CUSTOM`. If you do, you receive a `ValidationException` exception.

The `Configuration` parameter is required for all other data sources.

Type: [DataSourceConfiguration \(p. 457\)](#) object

Required: No

[Description \(p. 283\)](#)

A description for the data source.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1000.

Pattern: ^\P{C}*\$

Required: No

[IndexId \(p. 283\)](#)

The identifier of the index that should be associated with this data source.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

[LanguageCode \(p. 283\)](#)

The code for a language. This allows you to support a language for all documents when creating the data source. English is supported by default. For more information on supported languages, including their codes, see [Adding documents in languages other than English](#).

Type: String

Length Constraints: Minimum length of 2. Maximum length of 10.

Pattern: [a-zA-Z-]*

Required: No

[Name \(p. 283\)](#)

A unique name for the data source. A data source name can't be changed without deleting and recreating the data source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1000.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: Yes

[RoleArn \(p. 283\)](#)

The Amazon Resource Name (ARN) of a role with permission to access the data source. For more information, see [IAM Roles for Amazon Kendra](#).

You can't specify the `RoleArn` parameter when the `Type` parameter is set to `CUSTOM`. If you do, you receive a `ValidationException` exception.

The `RoleArn` parameter is required for all other data sources.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: arn:[a-zA-Z0-9-\.\.]{1,63}:[a-zA-Z0-9-\.\.]{0,63}:[a-zA-Z0-9-\.\.]{0,63}:[a-zA-Z0-9-\.\.]{0,63}:[^/].{0,1023}

Required: No

[Schedule \(p. 283\)](#)

Sets the frequency that Amazon Kendra will check the documents in your repository and update the index. If you don't set a schedule Amazon Kendra will not periodically update the index. You can call the `StartDataSourceSyncJob` operation to update the index.

You can't specify the `Schedule` parameter when the `Type` parameter is set to `CUSTOM`. If you do, you receive a `ValidationException` exception.

Type: String

Required: No

[Tags \(p. 283\)](#)

A list of key-value pairs that identify the data source. You can use the tags to identify and organize your resources and to control access to resources.

Type: Array of [Tag \(p. 548\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 200 items.

Required: No

[Type \(p. 283\)](#)

The type of repository that contains the data source.

Type: String

Valid Values: S3 | SHAREPOINT | DATABASE | SALESFORCE | ONEDRIVE | SERVICENOW | CUSTOM | CONFLUENCE | GOOGLEDRAVE | WEBCRAWLER | WORKDOCS

Required: Yes

Response Syntax

```
{  
    "Id": "string"  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[Id \(p. 290\)](#)

A unique identifier for the data source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceAlreadyExistException

HTTP Status Code: 400

ResourceNotFoundException

HTTP Status Code: 400

ServiceQuotaExceededException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

CreateFaq

Creates a new set of frequently asked question (FAQ) questions and answers.

Request Syntax

```
{  
    "ClientToken": "string",  
    "Description": "string",  
    "FileFormat": "string",  
    "IndexId": "string",  
    "LanguageCode": "string",  
    "Name": "string",  
    "RoleArn": "string",  
    "S3Path": {  
        "Bucket": "string",  
        "Key": "string"  
    },  
    "Tags": [  
        {  
            "Key": "string",  
            "Value": "string"  
        }  
    ]  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

[ClientToken \(p. 292\)](#)

A token that you provide to identify the request to create a FAQ. Multiple calls to the `CreateFaqRequest` operation with the same client token will create only one FAQ.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Required: No

[Description \(p. 292\)](#)

A description of the FAQ.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1000.

Pattern: ^\P{C}*\$

Required: No

[FileFormat \(p. 292\)](#)

The format of the input file. You can choose between a basic CSV format, a CSV format that includes customs attributes in a header, and a JSON format that includes custom attributes.

The format must match the format of the file stored in the S3 bucket identified in the `s3Path` parameter.

For more information, see [Adding questions and answers](#).

Type: String

Valid Values: CSV | CSV_WITH_HEADER | JSON

Required: No

[IndexId \(p. 292\)](#)

The identifier of the index that contains the FAQ.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

[LanguageCode \(p. 292\)](#)

The code for a language. This allows you to support a language for the FAQ document. English is supported by default. For more information on supported languages, including their codes, see [Adding documents in languages other than English](#).

Type: String

Length Constraints: Minimum length of 2. Maximum length of 10.

Pattern: [a-zA-Z-]*

Required: No

[Name \(p. 292\)](#)

The name that should be associated with the FAQ.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: Yes

[RoleArn \(p. 292\)](#)

The Amazon Resource Name (ARN) of a role with permission to access the S3 bucket that contains the FAQs. For more information, see [IAM Roles for Amazon Kendra](#).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: arn:[a-zA-Z0-9-\.]{1,63}:[a-zA-Z0-9-\.]{0,63}:[a-zA-Z0-9-\.]{0,63}:[a-zA-Z0-9-\.]{0,63}:[^/.]{0,1023}

Required: Yes

[S3Path \(p. 292\)](#)

The S3 location of the FAQ input data.

Type: [S3Path \(p. 513\)](#) object

Required: Yes

[Tags \(p. 292\)](#)

A list of key-value pairs that identify the FAQ. You can use the tags to identify and organize your resources and to control access to resources.

Type: Array of [Tag \(p. 548\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 200 items.

Required: No

Response Syntax

```
{  
  "Id": "string"  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[Id \(p. 294\)](#)

The unique identifier of the FAQ.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerException

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ServiceQuotaExceededException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

CreateIndex

Creates a new Amazon Kendra index. Index creation is an asynchronous operation. To determine if index creation has completed, check the `Status` field returned from a call to `DescribeIndex`. The `Status` field is set to `ACTIVE` when the index is ready to use.

Once the index is active you can index your documents using the `BatchPutDocument` operation or using one of the supported data sources.

Request Syntax

```
{
    "ClientToken": "string",
    "Description": "string",
    "Edition": "string",
    "Name": "string",
    "RoleArn": "string",
    "ServerSideEncryptionConfiguration": {
        "KmsKeyId": "string"
    },
    "Tags": [
        {
            "Key": "string",
            "Value": "string"
        }
    ],
    "UserContextPolicy": "string",
    "UserGroupResolutionConfiguration": {
        "UserGroupResolutionMode": "string"
    },
    "UserTokenConfigurations": [
        {
            "JsonTokenTypeConfiguration": {
                "GroupAttributeField": "string",
                "UserNameAttributeField": "string"
            },
            "JwtTokenTypeConfiguration": {
                "ClaimRegex": "string",
                "GroupAttributeField": "string",
                "Issuer": "string",
                "KeyLocation": "string",
                "SecretManagerArn": "string",
                "URL": "string",
                "UserNameAttributeField": "string"
            }
        }
    ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

ClientToken (p. 296)

A token that you provide to identify the request to create an index. Multiple calls to the `CreateIndex` operation with the same client token will create only one index.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Required: No

Description (p. 296)

A description for the index.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1000.

Pattern: ^\P{C}*\$

Required: No

Edition (p. 296)

The Amazon Kendra edition to use for the index. Choose DEVELOPER_EDITION for indexes intended for development, testing, or proof of concept. Use ENTERPRISE_EDITION for your production databases. Once you set the edition for an index, it can't be changed.

The Edition parameter is optional. If you don't supply a value, the default is ENTERPRISE_EDITION.

For more information on quota limits for enterprise and developer editions, see [Quotas](#).

Type: String

Valid Values: DEVELOPER_EDITION | ENTERPRISE_EDITION

Required: No

Name (p. 296)

The name for the new index.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1000.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: Yes

RoleArn (p. 296)

An AWS Identity and Access Management(IAM) role that gives Amazon Kendra permissions to access your Amazon CloudWatch logs and metrics. This is also the role used when you use the BatchPutDocument operation to index documents from an Amazon S3 bucket.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: arn:[a-zA-Z0-9-\.]{1,63}:[a-zA-Z0-9-\.]{0,63}:[a-zA-Z0-9-\.]{0,63}:[a-zA-Z0-9-\.]{0,63}:[^/.]{0,1023}

Required: Yes

ServerSideEncryptionConfiguration (p. 296)

The identifier of the AWS KMScustomer managed key (CMK) to use to encrypt data indexed by Amazon Kendra. Amazon Kendra doesn't support asymmetric CMKs.

Type: [ServerSideEncryptionConfiguration \(p. 529\)](#) object

Required: No

[Tags \(p. 296\)](#)

A list of key-value pairs that identify the index. You can use the tags to identify and organize your resources and to control access to resources.

Type: Array of [Tag \(p. 548\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 200 items.

Required: No

[UserContextPolicy \(p. 296\)](#)

The user context policy.

ATTRIBUTE_FILTER

All indexed content is searchable and displayable for all users. If you want to filter search results on user context, you can use the attribute filters of `_user_id` and `_group_ids` or you can provide user and group information in `UserContext`.

USER_TOKEN

Enables token-based user access control to filter search results on user context. All documents with no access control and all documents accessible to the user will be searchable and displayable.

Type: String

Valid Values: ATTRIBUTE_FILTER | USER_TOKEN

Required: No

[UserGroupResolutionConfiguration \(p. 296\)](#)

Enables fetching access levels of groups and users from an AWS Single Sign-On identity source. To configure this, see [UserGroupResolutionConfiguration](#).

Type: [UserGroupResolutionConfiguration \(p. 557\)](#) object

Required: No

[UserTokenConfigurations \(p. 296\)](#)

The user token configuration.

Type: Array of [UserTokenConfiguration \(p. 558\)](#) objects

Array Members: Maximum number of 1 item.

Required: No

Response Syntax

```
{  
    "Id": "string"  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Id ([p. 298](#))

The unique identifier of the index. Use this identifier when you query an index, set up a data source, or index a document.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerException

HTTP Status Code: 500

ResourceAlreadyExistException

HTTP Status Code: 400

ServiceQuotaExceededException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

CreateQuerySuggestionsBlockList

Creates a block list to exclude certain queries from suggestions.

Any query that contains words or phrases specified in the block list is blocked or filtered out from being shown as a suggestion.

You need to provide the file location of your block list text file in your S3 bucket. In your text file, enter each block word or phrase on a separate line.

For information on the current quota limits for block lists, see [Quotas for Amazon Kendra](#).

`CreateQuerySuggestionsBlockList` is currently not supported in the AWS GovCloud (US-West) region.

Request Syntax

```
{  
    "ClientToken": "string",  
    "Description": "string",  
    "IndexId": "string",  
    "Name": "string",  
    "RoleArn": "string",  
    "SourcesS3Path": {  
        "Bucket": "string",  
        "Key": "string"  
    },  
    "Tags": [  
        {  
            "Key": "string",  
            "Value": "string"  
        }  
    ]  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

ClientToken (p. 301)

A token that you provide to identify the request to create a query suggestions block list.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Required: No

Description (p. 301)

A user-friendly description for the block list.

For example, the description "List of all offensive words that can appear in user queries and need to be blocked from suggestions."

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1000.

Pattern: ^\P{C}*\$

Required: No

[IndexId \(p. 301\)](#)

The identifier of the index you want to create a query suggestions block list for.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

[Name \(p. 301\)](#)

A user friendly name for the block list.

For example, the block list named 'offensive-words' includes all offensive words that could appear in user queries and need to be blocked from suggestions.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z0-9](-*[a-zA-Z0-9])*

Required: Yes

[RoleArn \(p. 301\)](#)

The IAM (Identity and Access Management) role used by Amazon Kendra to access the block list text file in your S3 bucket.

You need permissions to the role ARN (Amazon Resource Name). The role needs S3 read permissions to your file in S3 and needs to give STS (Security Token Service) assume role permissions to Amazon Kendra.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: arn:[a-zA-Z0-9-\.]{1,63}:[a-zA-Z0-9-\.]{0,63}:[a-zA-Z0-9-\.]{0,63}:[a-zA-Z0-9-\.]{0,63}:[^/].{0,1023}

Required: Yes

[SourceS3Path \(p. 301\)](#)

The S3 path to your block list text file in your S3 bucket.

Each block word or phrase should be on a separate line in a text file.

For information on the current quota limits for block lists, see [Quotas for Amazon Kendra](#).

Type: [S3Path \(p. 513\)](#) object

Required: Yes

[Tags \(p. 301\)](#)

A tag that you can assign to a block list that categorizes the block list.

Type: Array of [Tag \(p. 548\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 200 items.

Required: No

Response Syntax

```
{  
    "Id": "string"  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Id (p. 303)

The unique identifier of the created block list.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ServiceQuotaExceededException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

CreateThesaurus

Creates a thesaurus for an index. The thesaurus contains a list of synonyms in Solr format.

Request Syntax

```
{  
    "ClientToken": "string",  
    "Description": "string",  
    "IndexId": "string",  
    "Name": "string",  
    "RoleArn": "string",  
    "SourcesS3Path": {  
        "Bucket": "string",  
        "Key": "string"  
    },  
    "Tags": [  
        {  
            "Key": "string",  
            "Value": "string"  
        }  
    ]  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

ClientToken (p. 305)

A token that you provide to identify the request to create a thesaurus. Multiple calls to the CreateThesaurus operation with the same client token will create only one thesaurus.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Required: No

Description (p. 305)

The description for the new thesaurus.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1000.

Pattern: ^\P{C}*\$

Required: No

IndexId (p. 305)

The unique identifier of the index for the new thesaurus.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

[Name \(p. 305\)](#)

The name for the new thesaurus.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: Yes

[RoleArn \(p. 305\)](#)

An IAM role that gives Amazon Kendra permissions to access thesaurus file specified in `SourceS3Path`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: `arn:[a-zA-Z0-9-\.]{1,63}:[a-zA-Z0-9-\.]{0,63}:[a-zA-Z0-9-\.]{0,63}:[a-zA-Z0-9-\.]{0,63}:[^/].{0,1023}`

Required: Yes

[SourceS3Path \(p. 305\)](#)

The thesaurus file Amazon S3 source path.

Type: [S3Path \(p. 513\)](#) object

Required: Yes

[Tags \(p. 305\)](#)

A list of key-value pairs that identify the thesaurus. You can use the tags to identify and organize your resources and to control access to resources.

Type: Array of [Tag \(p. 548\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 200 items.

Required: No

Response Syntax

```
{  
  "Id": "string"  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[Id \(p. 306\)](#)

The unique identifier of the thesaurus.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerException

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ServiceQuotaExceededException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)

- [AWS SDK for Ruby V3](#)

DeleteDataSource

Deletes an Amazon Kendra data source. An exception is not thrown if the data source is already being deleted. While the data source is being deleted, the `Status` field returned by a call to the `DescribeDataSource` operation is set to `DELETING`. For more information, see [Deleting Data Sources](#).

Request Syntax

```
{  
    "Id": "string",  
    "IndexId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

Id (p. 309)

The unique identifier of the data source to delete.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: Yes

IndexId (p. 309)

The unique identifier of the index associated with the data source.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerException

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DeleteFaq

Removes an FAQ from an index.

Request Syntax

```
{  
    "Id": "string",  
    "IndexId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

Id (p. 311)

The identifier of the FAQ to remove.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: Yes

IndexId (p. 311)

The index to remove the FAQ from.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DeleteIndex

Deletes an existing Amazon Kendra index. An exception is not thrown if the index is already being deleted. While the index is being deleted, the `Status` field returned by a call to the `DescribeIndex` operation is set to `DELETING`.

Request Syntax

```
{  
    "Id": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

Id (p. 313)

The identifier of the index to delete.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerException

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DeletePrincipalMapping

Deletes a group so that all users and sub groups that belong to the group can no longer access documents only available to that group.

For example, after deleting the group "Summer Interns", all interns who belonged to that group no longer see intern-only documents in their search results.

If you want to delete or replace users or sub groups of a group, you need to use the `PutPrincipalMapping` operation. For example, if a user in the group "Engineering" leaves the engineering team and another user takes their place, you provide an updated list of users or sub groups that belong to the "Engineering" group when calling `PutPrincipalMapping`. You can update your internal list of users or sub groups and input this list when calling `PutPrincipalMapping`.

`DeletePrincipalMapping` is currently not supported in the AWS GovCloud (US-West) region.

Request Syntax

```
{  
    "DataSourceId": "string",  
    "GroupId": "string",  
    "IndexId": "string",  
    "OrderingId": number  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

[DataSourceId \(p. 315\)](#)

The identifier of the data source you want to delete a group from.

This is useful if a group is tied to multiple data sources and you want to delete a group from accessing documents in a certain data source. For example, the groups "Research", "Engineering", and "Sales and Marketing" are all tied to the company's documents stored in the data sources Confluence and Salesforce. You want to delete "Research" and "Engineering" groups from Salesforce, so that these groups cannot access customer-related documents stored in Salesforce. Only "Sales and Marketing" should access documents in the Salesforce data source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: No

[GroupId \(p. 315\)](#)

The identifier of the group you want to delete.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: ^\P{C}*\$

Required: Yes

[IndexId \(p. 315\)](#)

The identifier of the index you want to delete a group from.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

[OrderingId \(p. 315\)](#)

The timestamp identifier you specify to ensure Amazon Kendra does not override the latest `DELETE` action with previous actions. The highest number ID, which is the ordering ID, is the latest action you want to process and apply on top of other actions with lower number IDs. This prevents previous actions with lower number IDs from possibly overriding the latest action.

The ordering ID can be the UNIX time of the last update you made to a group members list. You would then provide this list when calling `PutPrincipalMapping`. This ensures your `DELETE` action for that updated group with the latest members list doesn't get overwritten by earlier `DELETE` actions for the same group which are yet to be processed.

The default ordering ID is the current UNIX time in milliseconds that the action was received by Amazon Kendra.

Type: Long

Valid Range: Minimum value of 0. Maximum value of 32535158400000.

Required: No

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DeleteQuerySuggestionsBlockList

Deletes a block list used for query suggestions for an index.

A deleted block list might not take effect right away. Amazon Kendra needs to refresh the entire suggestions list to add back the queries that were previously blocked.

`DeleteQuerySuggestionsBlockList` is currently not supported in the AWS GovCloud (US-West) region.

Request Syntax

```
{  
    "Id": "string",  
    "IndexId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

Id (p. 318)

The unique identifier of the block list that needs to be deleted.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

IndexId (p. 318)

The identifier of the you want to delete a block list from.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerException

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DeleteThesaurus

Deletes an existing Amazon Kendra thesaurus.

Request Syntax

```
{  
    "Id": "string",  
    "IndexId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

Id ([p. 320](#))

The identifier of the thesaurus to delete.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: Yes

IndexId ([p. 320](#))

The identifier of the index associated with the thesaurus to delete.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DescribeDataSource

Gets information about a Amazon Kendra data source.

Request Syntax

```
{  
    "Id": "string",  
    "IndexId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

Id ([p. 322](#))

The unique identifier of the data source to describe.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: Yes

IndexId ([p. 322](#))

The identifier of the index that contains the data source.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

Response Syntax

```
{  
    "Configuration": {  
        "ConfluenceConfiguration": {  
            "AttachmentConfiguration": {  
                "AttachmentFieldMappings": [  
                    {  
                        "DataSourceFieldName": "string",  
                        "DateFormat": "string",  
                        "IndexFieldName": "string"  
                    }  
                ],  
                "CrawlAttachments": boolean  
            },  
            "BlogConfiguration": {  
                "BlogFieldMappings": [  
                    {  
                        "DataSourceFieldName": "string",  
                        "DateFormat": "string",  
                        "IndexFieldName": "string"  
                    }  
                ]  
            }  
        }  
    }  
}
```

```

        {
            "DataSourceFieldName": "string",
            "DateFormat": "string",
            "IndexFieldName": "string"
        }
    ]
},
"ExclusionPatterns": [ "string" ],
"InclusionPatterns": [ "string" ],
"PageConfiguration": {
    "PageFieldMappings": [
        {
            "DataSourceFieldName": "string",
            "DateFormat": "string",
            "IndexFieldName": "string"
        }
    ]
},
"SecretArn": "string",
"ServerUrl": "string",
"SpaceConfiguration": {
    "CrawlArchivedSpaces": boolean,
    "CrawlPersonalSpaces": boolean,
    "ExcludeSpaces": [ "string" ],
    "IncludeSpaces": [ "string" ],
    "SpaceFieldMappings": [
        {
            "DataSourceFieldName": "string",
            "DateFormat": "string",
            "IndexFieldName": "string"
        }
    ]
},
"Version": "string",
"VpcConfiguration": {
    "SecurityGroupIds": [ "string" ],
    "SubnetIds": [ "string" ]
}
},
"DatabaseConfiguration": {
    "AclConfiguration": {
        "AllowedGroupsColumnName": "string"
    },
    "ColumnConfiguration": {
        "ChangeDetectingColumns": [ "string" ],
        "DocumentDataColumnName": "string",
        "DocumentIdColumnName": "string",
        "DocumentTitleColumnName": "string",
        "FieldMappings": [
            {
                "DataSourceFieldName": "string",
                "DateFormat": "string",
                "IndexFieldName": "string"
            }
        ]
    },
    "ConnectionConfiguration": {
        "DatabaseHost": "string",
        "DatabaseName": "string",
        "DatabasePort": number,
        "SecretArn": "string",
        "TableName": "string"
    },
    "DatabaseEngineType": "string",
    "SqlConfiguration": {
        "QueryIdentifiersEnclosingOption": "string"
    }
}

```

```

},
"VpcConfiguration": {
    "SecurityGroupIds": [ "string" ],
    "SubnetIds": [ "string" ]
}
},
"GoogleDriveConfiguration": {
    "ExcludeMimeTypes": [ "string" ],
    "ExcludeSharedDrives": [ "string" ],
    "ExcludeUserAccounts": [ "string" ],
    "ExclusionPatterns": [ "string" ],
    "FieldMappings": [
        {
            "DataSourceFieldName": "string",
            "DateFormat": "string",
            "IndexFieldName": "string"
        }
    ],
    "InclusionPatterns": [ "string" ],
    "SecretArn": "string"
},
"OneDriveConfiguration": {
    "DisableLocalGroups": boolean,
    "ExclusionPatterns": [ "string" ],
    "FieldMappings": [
        {
            "DataSourceFieldName": "string",
            "DateFormat": "string",
            "IndexFieldName": "string"
        }
    ],
    "InclusionPatterns": [ "string" ],
    "OneDriveUsers": {
        "OneDriveUserList": [ "string" ],
        "OneDriveUserS3Path": {
            "Bucket": "string",
            "Key": "string"
        }
    },
    "SecretArn": "string",
    "TenantDomain": "string"
},
"S3Configuration": {
    "AccessControlListConfiguration": {
        "KeyPath": "string"
    },
    "BucketName": "string",
    "DocumentsMetadataConfiguration": {
        "S3Prefix": "string"
    },
    "ExclusionPatterns": [ "string" ],
    "InclusionPatterns": [ "string" ],
    "InclusionPrefixes": [ "string" ]
},
"SalesforceConfiguration": {
    "ChatterFeedConfiguration": {
        "DocumentDataFieldName": "string",
        "DocumentTitleFieldName": "string",
        "FieldMappings": [
            {
                "DataSourceFieldName": "string",
                "DateFormat": "string",
                "IndexFieldName": "string"
            }
        ],
        "IncludeFilterTypes": [ "string" ]
}
}

```

```
        },
        "CrawlAttachments": boolean,
        "ExcludeAttachmentFilePatterns": [ "string" ],
        "IncludeAttachmentFilePatterns": [ "string" ],
        "KnowledgeArticleConfiguration": {
            "CustomKnowledgeArticleTypeConfigurations": [
                {
                    "DocumentDataFieldName": "string",
                    "DocumentTitleFieldName": "string",
                    "FieldMappings": [
                        {
                            "DataSourceFieldName": "string",
                            "DateFormat": "string",
                            "IndexFieldName": "string"
                        }
                    ],
                    "Name": "string"
                }
            ],
            "IncludedStates": [ "string" ],
            "StandardKnowledgeArticleTypeConfiguration": {
                "DocumentDataFieldName": "string",
                "DocumentTitleFieldName": "string",
                "FieldMappings": [
                    {
                        "DataSourceFieldName": "string",
                        "DateFormat": "string",
                        "IndexFieldName": "string"
                    }
                ]
            }
        },
        "SecretArn": "string",
        "ServerUrl": "string",
        "StandardObjectAttachmentConfiguration": {
            "DocumentTitleFieldName": "string",
            "FieldMappings": [
                {
                    "DataSourceFieldName": "string",
                    "DateFormat": "string",
                    "IndexFieldName": "string"
                }
            ]
        },
        "StandardObjectConfigurations": [
            {
                "DocumentDataFieldName": "string",
                "DocumentTitleFieldName": "string",
                "FieldMappings": [
                    {
                        "DataSourceFieldName": "string",
                        "DateFormat": "string",
                        "IndexFieldName": "string"
                    }
                ],
                "Name": "string"
            }
        ]
    },
    "ServiceNowConfiguration": {
        "AuthenticationType": "string",
        "HostUrl": "string",
        "KnowledgeArticleConfiguration": {
            "CrawlAttachments": boolean,
            "DocumentDataFieldName": "string",
            "DocumentTitleFieldName": "string",
            "FieldMappings": [
                {
                    "DataSourceFieldName": "string",
                    "DateFormat": "string",
                    "IndexFieldName": "string"
                }
            ],
            "Name": "string"
        }
    }
}
```

```

    "ExcludeAttachmentFilePatterns": [ "string" ],
    "FieldMappings": [
        {
            "DataSourceFieldName": "string",
            "DateFormat": "string",
            "IndexFieldName": "string"
        }
    ],
    "FilterQuery": "string",
    "IncludeAttachmentFilePatterns": [ "string" ]
},
"SecretArn": "string",
"ServiceCatalogConfiguration": {
    "CrawlAttachments": boolean,
    "DocumentDataFieldName": "string",
    "DocumentTitleFieldName": "string",
    "ExcludeAttachmentFilePatterns": [ "string" ],
    "FieldMappings": [
        {
            "DataSourceFieldName": "string",
            "DateFormat": "string",
            "IndexFieldName": "string"
        }
    ],
    "IncludeAttachmentFilePatterns": [ "string" ]
},
"ServiceNowBuildVersion": "string"
},
"SharePointConfiguration": {
    "CrawlAttachments": boolean,
    "DisableLocalGroups": boolean,
    "DocumentTitleFieldName": "string",
    "ExclusionPatterns": [ "string" ],
    "FieldMappings": [
        {
            "DataSourceFieldName": "string",
            "DateFormat": "string",
            "IndexFieldName": "string"
        }
    ],
    "InclusionPatterns": [ "string" ],
    "SecretArn": "string",
    "SharePointVersion": "string",
    "SslCertificateS3Path": {
        "Bucket": "string",
        "Key": "string"
    },
    "Urls": [ "string" ],
    "UseChangeLog": boolean,
    "VpcConfiguration": {
        "SecurityGroupIds": [ "string" ],
        "SubnetIds": [ "string" ]
    }
},
"WebCrawlerConfiguration": {
    "AuthenticationConfiguration": {
        "BasicAuthentication": [
            {
                "Credentials": "string",
                "Host": "string",
                "Port": number
            }
        ]
    },
    "CrawlDepth": number,
    "MaxContentSizePerPageInMegabytes": number,

```

```
"MaxLinksPerPage": number,
"MaxUrlsPerMinuteCrawlRate": number,
"ProxyConfiguration": {
    "Credentials": "string",
    "Host": "string",
    "Port": number
},
"UrlExclusionPatterns": [ "string " ],
"UrlInclusionPatterns": [ "string " ],
"Urls": {
    "SeedUrlConfiguration": {
        "SeedUrls": [ "string " ],
        "WebCrawlerMode": "string"
    },
    "SiteMapsConfiguration": {
        "SiteMaps": [ "string " ]
    }
}
},
"WorkDocsConfiguration": {
    "CrawlComments": boolean,
    "ExclusionPatterns": [ "string " ],
    "FieldMappings": [
        {
            "DataSourceFieldName": "string",
            "DateFormat": "string",
            "IndexFieldName": "string"
        }
    ],
    "InclusionPatterns": [ "string " ],
    "OrganizationId": "string",
    "UseChangeLog": boolean
}
},
"CreatedAt": number,
"Description": "string",
"ErrorMessage": "string",
"Id": "string",
"IndexId": "string",
"LanguageCode": "string",
"Name": "string",
"RoleArn": "string",
"Schedule": "string",
"Status": "string",
"Type": "string",
"UpdatedAt": number
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[Configuration \(p. 322\)](#)

Information that describes where the data source is located and how the data source is configured. The specific information in the description depends on the data source provider.

Type: [DataSourceConfiguration \(p. 457\)](#) object

[CreatedAt \(p. 322\)](#)

The Unix timestamp of when the data source was created.

Type: Timestamp

Description (p. 322)

The description of the data source.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1000.

Pattern: ^\P{C}*\$

ErrorMessage (p. 322)

When the Status field value is FAILED, the ErrorMessage field contains a description of the error that caused the data source to fail.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: ^\P{C}*\$

Id (p. 322)

The identifier of the data source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

IndexId (p. 322)

The identifier of the index that contains the data source.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

LanguageCode (p. 322)

The code for a language. This shows a supported language for all documents in the data source. English is supported by default. For more information on supported languages, including their codes, see [Adding documents in languages other than English](#).

Type: String

Length Constraints: Minimum length of 2. Maximum length of 10.

Pattern: [a-zA-Z_-]*

Name (p. 322)

The name that you gave the data source when it was created.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1000.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

RoleArn (p. 322)

The Amazon Resource Name (ARN) of the role that enables the data source to access its resources.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: arn:[a-z0-9\.]{1,63}:[a-z0-9\.]{0,63}:[a-z0-9\.]{0,63}:[a-z0-9\.]{0,63}:[^/].{0,1023}

[Schedule \(p. 322\)](#)

The schedule that Amazon Kendra will update the data source.

Type: String

[Status \(p. 322\)](#)

The current status of the data source. When the status is ACTIVE the data source is ready to use. When the status is FAILED, the ErrorMessage field contains the reason that the data source failed.

Type: String

Valid Values: CREATING | DELETING | FAILED | UPDATING | ACTIVE

[Type \(p. 322\)](#)

The type of the data source.

Type: String

Valid Values: S3 | SHAREPOINT | DATABASE | SALESFORCE | ONEDRIVE | SERVICENOW | CUSTOM | CONFLUENCE | GOOGLEDRAVE | WEBcrawler | WORKDOCS

[UpdatedAt \(p. 322\)](#)

The Unix timestamp of when the data source was last updated.

Type: Timestamp

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DescribeFaq

Gets information about an FAQ list.

Request Syntax

```
{  
    "Id": "string",  
    "IndexId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

Id ([p. 331](#))

The unique identifier of the FAQ.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: Yes

IndexId ([p. 331](#))

The identifier of the index that contains the FAQ.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

Response Syntax

```
{  
    "CreatedAt": number,  
    "Description": "string",  
    "ErrorMessage": "string",  
    "FileFormat": "string",  
    "Id": "string",  
    "IndexId": "string",  
    "LanguageCode": "string",  
    "Name": "string",  
    "RoleArn": "string",  
    "S3Path": {  
        "Bucket": "string",  
        "Key": "string"  
    },  
}
```

```
    "Status": "string",
    "UpdatedAt": number
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[CreatedAt \(p. 331\)](#)

The date and time that the FAQ was created.

Type: Timestamp

[Description \(p. 331\)](#)

The description of the FAQ that you provided when it was created.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1000.

Pattern: ^\P{C}*\$

[ErrorMessage \(p. 331\)](#)

If the `Status` field is `FAILED`, the `ErrorMessage` field contains the reason why the FAQ failed.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: ^\P{C}*\$

[FileFormat \(p. 331\)](#)

The file format used by the input files for the FAQ.

Type: String

Valid Values: `CSV` | `CSV_WITH_HEADER` | `JSON`

[Id \(p. 331\)](#)

The identifier of the FAQ.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

[IndexId \(p. 331\)](#)

The identifier of the index that contains the FAQ.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

[LanguageCode \(p. 331\)](#)

The code for a language. This shows a supported language for the FAQ document. English is supported by default. For more information on supported languages, including their codes, see [Adding documents in languages other than English](#).

Type: String

Length Constraints: Minimum length of 2. Maximum length of 10.

Pattern: [a-zA-Z-]*

[Name \(p. 331\)](#)

The name that you gave the FAQ when it was created.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

[RoleArn \(p. 331\)](#)

The Amazon Resource Name (ARN) of the role that provides access to the S3 bucket containing the input files for the FAQ.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: arn:[a-z0-9-\.]{1,63}:[a-z0-9-\.]{0,63}:[a-z0-9-\.]{0,63}:[a-z0-9-\.]{0,63}:[^/].{0,1023}

[S3Path \(p. 331\)](#)

Information required to find a specific file in an Amazon S3 bucket.

Type: [S3Path \(p. 513\)](#) object

[Status \(p. 331\)](#)

The status of the FAQ. It is ready to use when the status is ACTIVE.

Type: String

Valid Values: CREATING | UPDATING | ACTIVE | DELETING | FAILED

[UpdatedAt \(p. 331\)](#)

The date and time that the FAQ was last updated.

Type: Timestamp

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

InternalServerException

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DescribeIndex

Describes an existing Amazon Kendra index

Request Syntax

```
{  
    "Id": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

Id (p. 335)

The name of the index to describe.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

Response Syntax

```
{  
    "CapacityUnits": {  
        "QueryCapacityUnits": number,  
        "StorageCapacityUnits": number  
    },  
    "CreatedAt": number,  
    "Description": string,  
    "DocumentMetadataConfigurations": [  
        {  
            "Name": string,  
            "Relevance": {  
                "Duration": string,  
                "Freshness": boolean,  
                "Importance": number,  
                "RankOrder": string,  
                "ValueImportanceMap": {  
                    string : number  
                }  
            },  
            "Search": {  
                "Displayable": boolean,  
                "Facetable": boolean,  
                "Searchable": boolean,  
                "Sortable": boolean  
            },  
            "Type": string  
        }  
    ],  
}
```

```

"Edition": "string",
"ErrorMessage": "string",
"Id": "string",
"IndexStatistics": {
    "FaqStatistics": {
        "IndexedQuestionAnswersCount": number
    },
    "TextDocumentStatistics": {
        "IndexedTextBytes": number,
        "IndexedTextDocumentsCount": number
    }
},
"Name": "string",
"RoleArn": "string",
"ServerSideEncryptionConfiguration": {
    "KmsKeyId": "string"
},
"Status": "string",
"UpdatedAt": number,
"UserContextPolicy": "string",
"UserGroupResolutionConfiguration": {
    "UserGroupResolutionMode": "string"
},
"UserTokenConfigurations": [
    {
        "JsonTokenTypeConfiguration": {
            "GroupAttributeField": "string",
            "UserNameAttributeField": "string"
        },
        "JwtTokenTypeConfiguration": {
            "ClaimRegex": "string",
            "GroupAttributeField": "string",
            "Issuer": "string",
            "KeyLocation": "string",
            "SecretManagerArn": "string",
            "URL": "string",
            "UserNameAttributeField": "string"
        }
    }
]
}

```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

CapacityUnits ([p. 335](#))

For Enterprise edition indexes, you can choose to use additional capacity to meet the needs of your application. This contains the capacity units used for the index. A 0 for the query capacity or the storage capacity indicates that the index is using the default capacity for the index.

Type: [CapacityUnitsConfiguration](#) ([p. 437](#)) object

CreatedAt ([p. 335](#))

The Unix datetime that the index was created.

Type: Timestamp

Description ([p. 335](#))

The description of the index.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1000.

Pattern: ^\P{C}*\$

[DocumentMetadataConfigurations \(p. 335\)](#)

Configuration settings for any metadata applied to the documents in the index.

Type: Array of [DocumentMetadataConfiguration \(p. 475\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 500 items.

[Edition \(p. 335\)](#)

The Amazon Kendra edition used for the index. You decide the edition when you create the index.

Type: String

Valid Values: DEVELOPER_EDITION | ENTERPRISE_EDITION

[ErrorMessage \(p. 335\)](#)

When the `status` field value is `FAILED`, the `ErrorMessage` field contains a message that explains why.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: ^\P{C}*\$

[Id \(p. 335\)](#)

The name of the index.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

[IndexStatistics \(p. 335\)](#)

Provides information about the number of FAQ questions and answers and the number of text documents indexed.

Type: [IndexStatistics \(p. 493\)](#) object

[Name \(p. 335\)](#)

The name of the index.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1000.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

[RoleArn \(p. 335\)](#)

The Amazon Resource Name (ARN) of the IAM role that gives Amazon Kendra permission to write to your Amazon Cloudwatch logs.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: arn:[a-z0-9-.]{1,63}:[a-z0-9-.]{0,63}:[a-z0-9-.]{0,63}:[a-z0-9-.]{0,63}:[^/.]{0,1023}

[ServerSideEncryptionConfiguration \(p. 335\)](#)

The identifier of the AWS KMS customer master key (CMK) used to encrypt your data. Amazon Kendra doesn't support asymmetric CMKs.

Type: [ServerSideEncryptionConfiguration \(p. 529\)](#) object

[Status \(p. 335\)](#)

The current status of the index. When the value is ACTIVE, the index is ready for use. If the Status field value is FAILED, the ErrorMessage field contains a message that explains why.

Type: String

Valid Values: CREATING | ACTIVE | DELETING | FAILED | UPDATING | SYSTEM_UPDATING

[UpdatedAt \(p. 335\)](#)

The Unix datetime that the index was last updated.

Type: Timestamp

[UserContextPolicy \(p. 335\)](#)

The user context policy for the Amazon Kendra index.

Type: String

Valid Values: ATTRIBUTE_FILTER | USER_TOKEN

[UserGroupResolutionConfiguration \(p. 335\)](#)

Shows whether you have enabled the configuration for fetching access levels of groups and users from an AWS Single Sign-On identity source.

Type: [UserGroupResolutionConfiguration \(p. 557\)](#) object

[UserTokenConfigurations \(p. 335\)](#)

The user token configuration for the Amazon Kendra index.

Type: Array of [UserTokenConfiguration \(p. 558\)](#) objects

Array Members: Maximum number of 1 item.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

InternalServerException

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DescribePrincipalMapping

Describes the processing of `PUT` and `DELETE` actions for mapping users to their groups. This includes information on the status of actions currently processing or yet to be processed, when actions were last updated, when actions were received by Amazon Kendra, the latest action that should process and apply after other actions, and useful error messages if an action could not be processed.

`DescribePrincipalMapping` is currently not supported in the AWS GovCloud (US-West) region.

Request Syntax

```
{  
    "DataSourceId": "string",  
    "GroupId": "string",  
    "IndexId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

[DataSourceId \(p. 340\)](#)

The identifier of the data source to check the processing of `PUT` and `DELETE` actions for mapping users to their groups.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: No

[GroupId \(p. 340\)](#)

The identifier of the group required to check the processing of `PUT` and `DELETE` actions for mapping users to their groups.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: ^\P{C}*\$/

Required: Yes

[IndexId \(p. 340\)](#)

The identifier of the index required to check the processing of `PUT` and `DELETE` actions for mapping users to their groups.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: Yes

Response Syntax

```
{  
    "DataSourceId": "string",  
    "GroupId": "string",  
    "GroupOrderingIdSummaries": [  
        {  
            "FailureReason": "string",  
            "LastUpdatedAt": number,  
            "OrderingId": number,  
            "ReceivedAt": number,  
            "Status": "string"  
        }  
    ],  
    "IndexId": "string"  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[DataSourceld \(p. 341\)](#)

Shows the identifier of the data source to see information on the processing of `PUT` and `DELETE` actions for mapping users to their groups.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

[GroupId \(p. 341\)](#)

Shows the identifier of the group to see information on the processing of `PUT` and `DELETE` actions for mapping users to their groups.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: ^\P{C}*\$/

[GroupOrderingIdSummaries \(p. 341\)](#)

Shows the following information on the processing of `PUT` and `DELETE` actions for mapping users to their groups:

- Status – the status can be either `PROCESSING`, `SUCCEEDED`, `DELETING`, `DELETED`, or `FAILED`.
- Last updated – the last date-time an action was updated.
- Received – the last date-time an action was received or submitted.
- Ordering ID – the latest action that should process and apply after other actions.
- Failure reason – the reason an action could not be processed.

Type: Array of [GroupOrderingIdSummary \(p. 486\)](#) objects

Array Members: Maximum number of 10 items.

IndexId (p. 341)

Shows the identifier of the index to see information on the processing of `PUT` and `DELETE` actions for mapping users to their groups.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DescribeQuerySuggestionsBlockList

Describes a block list used for query suggestions for an index.

This is used to check the current settings that are applied to a block list.

DescribeQuerySuggestionsBlockList is currently not supported in the AWS GovCloud (US-West) region.

Request Syntax

```
{  
    "Id": "string",  
    "IndexId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

Id (p. 343)

The unique identifier of the block list.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

IndexId (p. 343)

The identifier of the index for the block list.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

Response Syntax

```
{  
    "CreatedAt": number,  
    "Description": "string",  
    "ErrorMessage": "string",  
    "FileSizeBytes": number,  
    "Id": "string",  
    "IndexId": "string",  
    "ItemCount": number,  
    "Name": "string",  
}
```

```
"RoleArn": "string",
"SourcesS3Path": {
    "Bucket": "string",
    "Key": "string"
},
"Status": "string",
"UpdatedAt": number
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[CreatedAt \(p. 343\)](#)

Shows the date-time a block list for query suggestions was created.

Type: Timestamp

[Description \(p. 343\)](#)

Shows the description for the block list.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1000.

Pattern: ^\P{C}*\$

[ErrorMessage \(p. 343\)](#)

Shows the error message with details when there are issues in processing the block list.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: ^\P{C}*\$

[FileSizeBytes \(p. 343\)](#)

Shows the current size of the block list text file in S3.

Type: Long

[Id \(p. 343\)](#)

Shows the unique identifier of the block list.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

[IndexId \(p. 343\)](#)

Shows the identifier of the index for the block list.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

ItemCount (p. 343)

Shows the current number of valid, non-empty words or phrases in the block list text file.

Type: Integer

Name (p. 343)

Shows the name of the block list.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z0-9](-*[a-zA-Z0-9])*

RoleArn (p. 343)

Shows the current IAM (Identity and Access Management) role used by Amazon Kendra to access the block list text file in S3.

The role needs S3 read permissions to your file in S3 and needs to give STS (Security Token Service) assume role permissions to Amazon Kendra.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: arn:[a-zA-Z0-9-\.]{1,63}:[a-zA-Z0-9-\.]{0,63}:[a-zA-Z0-9-\.]{0,63}:[a-zA-Z0-9-\.]{0,63}:[^/].{0,1023}

SourceS3Path (p. 343)

Shows the current S3 path to your block list text file in your S3 bucket.

Each block word or phrase should be on a separate line in a text file.

For information on the current quota limits for block lists, see [Quotas for Amazon Kendra](#).

Type: [S3Path \(p. 513\)](#) object

Status (p. 343)

Shows whether the current status of the block list is ACTIVE or INACTIVE.

Type: String

Valid Values: ACTIVE | CREATING | DELETING | UPDATING |
ACTIVE_BUT_UPDATE_FAILED | FAILED

UpdatedAt (p. 343)

Shows the date-time a block list for query suggestions was last updated.

Type: Timestamp

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

InternalServerException

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DescribeQuerySuggestionsConfig

Describes the settings of query suggestions for an index.

This is used to check the current settings applied to query suggestions.

`DescribeQuerySuggestionsConfig` is currently not supported in the AWS GovCloud (US-West) region.

Request Syntax

```
{  
    "IndexId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

IndexId (p. 347)

The identifier of the index you want to describe query suggestions settings for.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

Response Syntax

```
{  
    "IncludeQueriesWithoutUserInformation": boolean,  
    "LastClearTime": number,  
    "LastSuggestionsBuildTime": number,  
    "MinimumNumberOfQueryingUsers": number,  
    "MinimumQueryCount": number,  
    "Mode": "string",  
    "QueryLogLookBackWindowInDays": number,  
    "Status": "string",  
    "TotalSuggestionsCount": number  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

IncludeQueriesWithoutUserInformation (p. 347)

Shows whether Amazon Kendra uses all queries or only uses queries that include user information to generate query suggestions.

Type: Boolean

[LastClearTime \(p. 347\)](#)

Shows the date-time query suggestions for an index was last cleared.

After you clear suggestions, Amazon Kendra learns new suggestions based on new queries added to the query log from the time you cleared suggestions. Amazon Kendra only considers re-occurrences of a query from the time you cleared suggestions.

Type: Timestamp

[LastSuggestionsBuildTime \(p. 347\)](#)

Shows the date-time query suggestions for an index was last updated.

Type: Timestamp

[MinimumNumberOfQueryingUsers \(p. 347\)](#)

Shows the minimum number of unique users who must search a query in order for the query to be eligible to suggest to your users.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 10000.

[MinimumQueryCount \(p. 347\)](#)

Shows the minimum number of times a query must be searched in order for the query to be eligible to suggest to your users.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 10000.

[Mode \(p. 347\)](#)

Shows whether query suggestions are currently in `ENABLED` mode or `LEARN_ONLY` mode.

By default, Amazon Kendra enables query suggestions.`LEARN_ONLY` turns off query suggestions for your users. You can change the mode using the [UpdateQuerySuggestionsConfig](#) operation.

Type: String

Valid Values: `ENABLED` | `LEARN_ONLY`

[QueryLogLookBackWindowInDays \(p. 347\)](#)

Shows how recent your queries are in your query log time window (in days).

Type: Integer

[Status \(p. 347\)](#)

Shows whether the status of query suggestions settings is currently Active or Updating.

Active means the current settings apply and Updating means your changed settings are in the process of applying.

Type: String

Valid Values: `ACTIVE` | `UPDATING`

[TotalSuggestionsCount \(p. 347\)](#)

Shows the current total count of query suggestions for an index.

This count can change when you update your query suggestions settings, if you filter out certain queries from suggestions using a block list, and as the query log accumulates more queries for Amazon Kendra to learn from.

Type: Integer

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DescribeThesaurus

Describes an existing Amazon Kendra thesaurus.

Request Syntax

```
{  
    "Id": "string",  
    "IndexId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

Id (p. 350)

The identifier of the thesaurus to describe.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: Yes

IndexId (p. 350)

The identifier of the index associated with the thesaurus to describe.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

Response Syntax

```
{  
    "CreatedAt": number,  
    "Description": "string",  
    "ErrorMessage": "string",  
    "FileSizeBytes": number,  
    "Id": "string",  
    "IndexId": "string",  
    "Name": "string",  
    "RoleArn": "string",  
    "SourceS3Path": {  
        "Bucket": "string",  
        "Key": "string"  
    },  
    "Status": "string",  
    "SynonymRuleCount": number,
```

```
    "TermCount": number,  
    "UpdatedAt": number  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[CreatedAt \(p. 350\)](#)

The Unix datetime that the thesaurus was created.

Type: Timestamp

[Description \(p. 350\)](#)

The thesaurus description.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1000.

Pattern: ^\P{C}*\$

[ErrorMessage \(p. 350\)](#)

When the `Status` field value is `FAILED`, the `ErrorMessage` field provides more information.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: ^\P{C}*\$

[FileSizeBytes \(p. 350\)](#)

The size of the thesaurus file in bytes.

Type: Long

[Id \(p. 350\)](#)

The identifier of the thesaurus.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

[IndexId \(p. 350\)](#)

The identifier of the index associated with the thesaurus to describe.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

[Name \(p. 350\)](#)

The thesaurus name.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

[RoleArn \(p. 350\)](#)

An IAM role that gives Amazon Kendra permissions to access thesaurus file specified in `SourceS3Path`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: arn:[a-zA-Z0-9-.]{1,63}:[a-zA-Z0-9-.]{0,63}:[a-zA-Z0-9-.]{0,63}:[a-zA-Z0-9-.]{0,63}:[^/.]{0,1023}

[SourceS3Path \(p. 350\)](#)

Information required to find a specific file in an Amazon S3 bucket.

Type: [S3Path \(p. 513\)](#) object

[Status \(p. 350\)](#)

The current status of the thesaurus. When the value is `ACTIVE`, queries are able to use the thesaurus. If the `Status` field value is `FAILED`, the `ErrorMessage` field provides more information.

If the status is `ACTIVE_BUT_UPDATE_FAILED`, it means that Amazon Kendra could not ingest the new thesaurus file. The old thesaurus file is still active.

Type: String

Valid Values: `CREATING` | `ACTIVE` | `DELETING` | `UPDATING` |
`ACTIVE_BUT_UPDATE_FAILED` | `FAILED`

[SynonymRuleCount \(p. 350\)](#)

The number of synonym rules in the thesaurus file.

Type: Long

[TermCount \(p. 350\)](#)

The number of unique terms in the thesaurus file. For example, the synonyms `a`, `b`, `c` and `a=>d`, the term count would be 4.

Type: Long

[UpdatedAt \(p. 350\)](#)

The Unix datetime that the thesaurus was last updated.

Type: Timestamp

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

GetQuerySuggestions

Fetches the queries that are suggested to your users.

GetQuerySuggestions is currently not supported in the AWS GovCloud (US-West) region.

Request Syntax

```
{  
    "IndexId": "string",  
    "MaxSuggestionsCount": number,  
    "QueryText": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

IndexId (p. 354)

The identifier of the index you want to get query suggestions from.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

MaxSuggestionsCount (p. 354)

The maximum number of query suggestions you want to show to your users.

Type: Integer

Required: No

QueryText (p. 354)

The text of a user's query to generate query suggestions.

A query is suggested if the query prefix matches what a user starts to type as their query.

Amazon Kendra does not show any suggestions if a user types fewer than two characters or more than 60 characters. A query must also have at least one search result and contain at least one word of more than four characters.

Type: String

Pattern: ^\P{C}*\$

Required: Yes

Response Syntax

```
{
```

```
"QuerySuggestionsId": "string",
"Suggestions": [
  {
    "Id": "string",
    "Value": {
      "Text": {
        "Highlights": [
          {
            "BeginOffset": number,
            "EndOffset": number
          }
        ],
        "Text": "string"
      }
    }
  }
]
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[QuerySuggestionsId \(p. 354\)](#)

The unique identifier for a list of query suggestions for an index.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 36.

[Suggestions \(p. 354\)](#)

A list of query suggestions for an index.

Type: Array of [Suggestion \(p. 544\)](#) objects

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ServiceQuotaExceededException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListDataSources

Lists the data sources that you have created.

Request Syntax

```
{  
    "IndexId": "string",  
    "MaxResults": number,  
    "NextToken": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

IndexId (p. 357)

The identifier of the index that contains the data source.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

MaxResults (p. 357)

The maximum number of data sources to return.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

NextToken (p. 357)

If the previous response was incomplete (because there is more data to retrieve), Amazon Kendra returns a pagination token in the response. You can use this pagination token to retrieve the next set of data sources (`DataSourceSummaryItems`).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 800.

Required: No

Response Syntax

```
{  
    "NextToken": "string",  
    "SummaryItems": [  
        {  
            "CreatedAt": number,  
            "LastUpdated": number,  
            "Name": "string",  
            "Type": "string"  
        }  
    ]  
}
```

```
        "Id": "string",
        "LanguageCode": "string",
        "Name": "string",
        "Status": "string",
        "Type": "string",
        "UpdatedAt": number
    }
]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[NextToken \(p. 357\)](#)

If the response is truncated, Amazon Kendra returns this token that you can use in the subsequent request to retrieve the next set of data sources.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 800.

[SummaryItems \(p. 357\)](#)

An array of summary information for one or more data sources.

Type: Array of [DataSourceSummary \(p. 460\)](#) objects

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

InternalServerException

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListDataSourceSyncJobs

Gets statistics about synchronizing Amazon Kendra with a data source.

Request Syntax

```
{  
    "Id": "string",  
    "IndexId": "string",  
    "MaxResults": number,  
    "NextToken": "string",  
    "StartTimeFilter": {  
        "EndTime": number,  
        "StartTime": number  
    },  
    "StatusFilter": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

Id (p. 360)

The identifier of the data source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: Yes

IndexId (p. 360)

The identifier of the index that contains the data source.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

MaxResults (p. 360)

The maximum number of synchronization jobs to return in the response. If there are fewer results in the list, this response contains only the actual results.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 10.

Required: No

[NextToken](#) (p. 360)

If the previous response was incomplete (because there is more data to retrieve), Amazon Kendra returns a pagination token in the response. You can use this pagination token to retrieve the next set of jobs.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 800.

Required: No

[StartTimeFilter](#) (p. 360)

When specified, the synchronization jobs returned in the list are limited to jobs between the specified dates.

Type: [TimeRange](#) (p. 553) object

Required: No

[StatusFilter](#) (p. 360)

When specified, only returns synchronization jobs with the `Status` field equal to the specified status.

Type: String

Valid Values: FAILED | SUCCEEDED | SYNCING | INCOMPLETE | STOPPING | ABORTED | SYNCING_INDEXING

Required: No

Response Syntax

```
{  
    "History": [  
        {  
            "DataSourceErrorCode": "string",  
            "EndTime": number,  
            "ErrorCode": "string",  
            "ErrorMessage": "string",  
            "ExecutionId": "string",  
            "Metrics": {  
                "DocumentsAdded": "string",  
                "DocumentsDeleted": "string",  
                "DocumentsFailed": "string",  
                "DocumentsModified": "string",  
                "DocumentsScanned": "string"  
            },  
            "StartTime": number,  
            "Status": "string"  
        }  
    ],  
    "NextToken": "string"  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[History \(p. 361\)](#)

A history of synchronization jobs for the data source.

Type: Array of [DataSourceSyncJob \(p. 462\)](#) objects

[NextToken \(p. 361\)](#)

If the response is truncated, Amazon Kendra returns this token that you can use in the subsequent request to retrieve the next set of jobs.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 800.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)

- [AWS SDK for Ruby V3](#)

ListFaqs

Gets a list of FAQ lists associated with an index.

Request Syntax

```
{  
    "IndexId": "string",  
    "MaxResults": number,  
    "NextToken": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

IndexId (p. 364)

The index that contains the FAQ lists.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

MaxResults (p. 364)

The maximum number of FAQs to return in the response. If there are fewer results in the list, this response contains only the actual results.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

NextToken (p. 364)

If the previous response was incomplete (because there is more data to retrieve), Amazon Kendra returns a pagination token in the response. You can use this pagination token to retrieve the next set of FAQs.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 800.

Required: No

Response Syntax

```
{
```

```
"FaqSummaryItems": [
  {
    "CreatedAt": number,
    "FileFormat": "string",
    "Id": "string",
    "LanguageCode": "string",
    "Name": "string",
    "Status": "string",
    "UpdatedAt": number
  }
],
"NextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

FaqSummaryItems (p. 364)

information about the FAQs associated with the specified index.

Type: Array of [FaqSummary \(p. 481\)](#) objects

NextToken (p. 364)

If the response is truncated, Amazon Kendra returns this token that you can use in the subsequent request to retrieve the next set of FAQs.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 800.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListGroupsOlderThanOrderingId

Provides a list of groups that are mapped to users before a given ordering or timestamp identifier.

`ListGroupsOlderThanOrderingId` is currently not supported in the AWS GovCloud (US-West) region.

Request Syntax

```
{  
    "DataSourceId": "string",  
    "IndexId": "string",  
    "MaxResults": number,  
    "NextToken": "string",  
    "OrderingId": number  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

DataSourceId ([p. 367](#))

The identifier of the data source for getting a list of groups mapped to users before a given ordering timestamp identifier.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: No

IndexId ([p. 367](#))

The identifier of the index for getting a list of groups mapped to users before a given ordering or timestamp identifier.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

MaxResults ([p. 367](#))

The maximum number of returned groups that are mapped to users before a given ordering or timestamp identifier.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 10.

Required: No

[NextToken \(p. 367\)](#)

If the previous response was incomplete (because there is more data to retrieve), Amazon Kendra returns a pagination token in the response. You can use this pagination token to retrieve the next set of groups that are mapped to users before a given ordering or timestamp identifier.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 800.

Required: No

[OrderingId \(p. 367\)](#)

The timestamp identifier used for the latest `PUT` or `DELETE` action for mapping users to their groups.

Type: Long

Valid Range: Minimum value of 0. Maximum value of 32535158400000.

Required: Yes

Response Syntax

```
{  
    "GroupsSummaries": [  
        {  
            "GroupId": "string",  
            "OrderingId": number  
        }  
    ],  
    "NextToken": "string"  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[GroupsSummaries \(p. 368\)](#)

Summary information for list of groups that are mapped to users before a given ordering or timestamp identifier.

Type: Array of [GroupSummary \(p. 488\)](#) objects

[NextToken \(p. 368\)](#)

If the response is truncated, Amazon Kendra returns this token that you can use in the subsequent request to retrieve the next set of groups that are mapped to users before a given ordering or timestamp identifier.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 800.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListIndices

Lists the Amazon Kendra indexes that you have created.

Request Syntax

```
{  
    "MaxResults": number,  
    "NextToken": string  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

MaxResults (p. 370)

The maximum number of data sources to return.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

NextToken (p. 370)

If the previous response was incomplete (because there is more data to retrieve), Amazon Kendra returns a pagination token in the response. You can use this pagination token to retrieve the next set of indexes (`DataSourceSummaryItems`).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 800.

Required: No

Response Syntax

```
{  
    "IndexConfigurationSummaryItems": [  
        {  
            "CreatedAt": number,  
            "Edition": string,  
            "Id": string,  
            "Name": string,  
            "Status": string,  
            "UpdatedAt": number  
        }  
    ],  
    "NextToken": string  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[IndexConfigurationSummaryItems \(p. 370\)](#)

An array of summary information for one or more indexes.

Type: Array of [IndexConfigurationSummary \(p. 491\)](#) objects

[NextToken \(p. 370\)](#)

If the response is truncated, Amazon Kendra returns this token that you can use in the subsequent request to retrieve the next set of indexes.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 800.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

InternalServerException

HTTP Status Code: 500

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListQuerySuggestionsBlockLists

Lists the block lists used for query suggestions for an index.

For information on the current quota limits for block lists, see [Quotas for Amazon Kendra](#).

`ListQuerySuggestionsBlockLists` is currently not supported in the AWS GovCloud (US-West) region.

Request Syntax

```
{  
    "IndexId": "string",  
    "MaxResults": number,  
    "NextToken": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

IndexId (p. 372)

The identifier of the index for a list of all block lists that exist for that index.

For information on the current quota limits for block lists, see [Quotas for Amazon Kendra](#).

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

MaxResults (p. 372)

The maximum number of block lists to return.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

NextToken (p. 372)

If the previous response was incomplete (because there is more data to retrieve), Amazon Kendra returns a pagination token in the response. You can use this pagination token to retrieve the next set of block lists (`BlockListSummaryItems`).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 800.

Required: No

Response Syntax

```
{  
    "BlockListSummaryItems": [  
        {  
            "CreatedAt": number,  
            "Id": "string",  
            "ItemCount": number,  
            "Name": "string",  
            "Status": "string",  
            "UpdatedAt": number  
        }  
    ],  
    "NextToken": "string"  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[BlockListSummaryItems \(p. 373\)](#)

Summary items for a block list.

This includes summary items on the block list ID, block list name, when the block list was created, when the block list was last updated, and the count of block words/phrases in the block list.

For information on the current quota limits for block lists, see [Quotas for Amazon Kendra](#).

Type: Array of [QuerySuggestionsBlockListSummary \(p. 506\)](#) objects

[NextToken \(p. 373\)](#)

If the response is truncated, Amazon Kendra returns this token that you can use in the subsequent request to retrieve the next set of block lists.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 800.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListTagsForResource

Gets a list of tags associated with a specified resource. Indexes, FAQs, and data sources can have tags associated with them.

Request Syntax

```
{  
    "ResourceARN": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

[ResourceARN \(p. 375\)](#)

The Amazon Resource Name (ARN) of the index, FAQ, or data source to get a list of tags for.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1011.

Required: Yes

Response Syntax

```
{  
    "Tags": [  
        {  
            "Key": "string",  
            "Value": "string"  
        }  
    ]  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[Tags \(p. 375\)](#)

A list of tags associated with the index, FAQ, or data source.

Type: Array of [Tag \(p. 548\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 200 items.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

InternalServerException

HTTP Status Code: 500

ResourceUnavailableException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListThesauri

Lists the Amazon Kendra thesauri associated with an index.

Request Syntax

```
{  
    "IndexId": "string",  
    "MaxResults": number,  
    "NextToken": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

IndexId ([p. 377](#))

The identifier of the index associated with the thesaurus to list.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

MaxResults ([p. 377](#))

The maximum number of thesauri to return.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

NextToken ([p. 377](#))

If the previous response was incomplete (because there is more data to retrieve), Amazon Kendra returns a pagination token in the response. You can use this pagination token to retrieve the next set of thesauri (`ThesaurusSummaryItems`).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 800.

Required: No

Response Syntax

```
{  
    "NextToken": "string",  
    "ThesaurusSummaryItems": [  
        {
```

```
        "CreatedAt": number,
        "Id": "string",
        "Name": "string",
        "Status": "string",
        "UpdatedAt": number
    }
]
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[NextToken](#) (p. 377)

If the response is truncated, Amazon Kendra returns this token that you can use in the subsequent request to retrieve the next set of thesauri.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 800.

[ThesaurusSummaryItems](#) (p. 377)

An array of summary information for a thesaurus or multiple thesauri.

Type: Array of [ThesaurusSummary](#) (p. 551) objects

Errors

For information about the errors that are common to all actions, see [Common Errors](#) (p. 563).

AccessDeniedException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

PutPrincipalMapping

Maps users to their groups so that you only need to provide the user ID when you issue the query.

You can also map sub groups to groups. For example, the group "Company Intellectual Property Teams" includes sub groups "Research" and "Engineering". These sub groups include their own list of users or people who work in these teams. Only users who work in research and engineering, and therefore belong in the intellectual property group, can see top-secret company documents in their search results.

You map users to their groups when you want to filter search results for different users based on their group's access to documents. For more information on filtering search results for different users, see [Filtering on user context](#).

If more than five `PUT` actions for a group are currently processing, a validation exception is thrown.

`PutPrincipalMapping` is currently not supported in the AWS GovCloud (US-West) region.

Request Syntax

```
{  
    "DataSourceId": "string",  
    "GroupId": "string",  
    "GroupMembers": {  
        "MemberGroups": [  
            {  
                "DataSourceId": "string",  
                "GroupId": "string"  
            }  
        ],  
        "MemberUsers": [  
            {  
                "UserId": "string"  
            }  
        ],  
        "S3PathforGroupMembers": {  
            "Bucket": "string",  
            "Key": "string"  
        }  
    },  
    "IndexId": "string",  
    "OrderingId": number,  
    "RoleArn": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

DataSourceId (p. 380)

The identifier of the data source you want to map users to their groups.

This is useful if a group is tied to multiple data sources, but you only want the group to access documents of a certain data source. For example, the groups "Research", "Engineering", and "Sales and Marketing" are all tied to the company's documents stored in the data sources Confluence and Salesforce. However, "Sales and Marketing" team only needs access to customer-related documents stored in Salesforce.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: No

[GroupId \(p. 380\)](#)

The identifier of the group you want to map its users to.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: ^\P{C}*\$

Required: Yes

[GroupMembers \(p. 380\)](#)

The list that contains your users or sub groups that belong the same group.

For example, the group "Company" includes the user "CEO" and the sub groups "Research", "Engineering", and "Sales and Marketing".

If you have more than 1000 users and/or sub groups for a single group, you need to provide the path to the S3 file that lists your users and sub groups for a group. Your sub groups can contain more than 1000 users, but the list of sub groups that belong to a group (and/or users) must be no more than 1000.

Type: [GroupMembers \(p. 485\)](#) object

Required: Yes

[IndexId \(p. 380\)](#)

The identifier of the index you want to map users to their groups.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

[OrderId \(p. 380\)](#)

The timestamp identifier you specify to ensure Amazon Kendra does not override the latest `PUT` action with previous actions. The highest number ID, which is the ordering ID, is the latest action you want to process and apply on top of other actions with lower number IDs. This prevents previous actions with lower number IDs from possibly overriding the latest action.

The ordering ID can be the UNIX time of the last update you made to a group members list. You would then provide this list when calling `PutPrincipalMapping`. This ensures your `PUT` action for that updated group with the latest members list doesn't get overwritten by earlier `PUT` actions for the same group which are yet to be processed.

The default ordering ID is the current UNIX time in milliseconds that the action was received by Amazon Kendra.

Type: Long

Valid Range: Minimum value of 0. Maximum value of 32535158400000.

Required: No

RoleArn (p. 380)

The Amazon Resource Name (ARN) of a role that has access to the S3 file that contains your list of users or sub groups that belong to a group.

For more information, see [IAM roles for Amazon Kendra](#).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: `arn:[a-zA-Z0-9-.]{1,63}:[a-zA-Z0-9-.]{0,63}:[a-zA-Z0-9-.]{0,63}:[a-zA-Z0-9-.]{0,63}:[^/.]{0,1023}`

Required: No

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerException

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ServiceQuotaExceededException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

Query

Searches an active index. Use this API to search your documents using query. The `Query` operation enables to do faceted search and to filter results based on document attributes.

It also enables you to provide user context that Amazon Kendra uses to enforce document access control in the search results.

Amazon Kendra searches your index for text content and question and answer (FAQ) content. By default the response contains three types of results.

- Relevant passages
- Matching FAQs
- Relevant documents

You can specify that the query return only one type of result using the `QueryResultTypeConfig` parameter.

Each query returns the 100 most relevant results.

Request Syntax

```
{  
    "AttributeFilter": {  
        "AndAllFilters": [  
            "AttributeFilter"  
        ],  
        "ContainsAll": {  
            "Key": "string",  
            "Value": {  
                "DateValue": number,  
                "LongValue": number,  
                "StringListValue": [ "string" ],  
                "StringValue": "string"  
            }  
        },  
        "ContainsAny": {  
            "Key": "string",  
            "Value": {  
                "DateValue": number,  
                "LongValue": number,  
                "StringListValue": [ "string" ],  
                "StringValue": "string"  
            }  
        },  
        "EqualsTo": {  
            "Key": "string",  
            "Value": {  
                "DateValue": number,  
                "LongValue": number,  
                "StringListValue": [ "string" ],  
                "StringValue": "string"  
            }  
        },  
        "GreaterThan": {  
            "Key": "string",  
            "Value": {  
                "DateValue": number,  
                "LongValue": number,  
                "StringListValue": [ "string" ],  
                "StringValue": "string"  
            }  
        }  
    }  
}
```

```

        "StringValue": "string"
    }
},
"GreaterThanOrEquals": {
    "Key": "string",
    "Value": {
        "DateValue": number,
        "LongValue": number,
        "StringListValue": [ "string" ],
        "StringValue": "string"
    }
},
"LessThan": {
    "Key": "string",
    "Value": {
        "DateValue": number,
        "LongValue": number,
        "StringListValue": [ "string" ],
        "StringValue": "string"
    }
},
"LessThanOrEquals": {
    "Key": "string",
    "Value": {
        "DateValue": number,
        "LongValue": number,
        "StringListValue": [ "string" ],
        "StringValue": "string"
    }
},
"NotFilter": "AttributeFilter",
"OrAllFilters": [
    "AttributeFilter"
]
},
"DocumentRelevanceOverrideConfigurations": [
{
    "Name": "string",
    "Relevance": {
        "Duration": "string",
        "Freshness": boolean,
        "Importance": number,
        "RankOrder": "string",
        "ValueImportanceMap": {
            "string" : number
        }
    }
}
],
"Facets": [
{
    "DocumentAttributeKey": "string"
}
],
"IndexId": "string",
"PageNumber": number,
"PageSize": number,
"QueryResultTypeFilter": "string",
"QueryText": "string",
"RequestedDocumentAttributes": [ "string" ],
"SortingConfiguration": {
    "DocumentAttributeKey": "string",
    "SortOrder": "string"
},
"UserContext": {
    "DataSourceGroups": [

```

```
{  
    "DataSourceId": "string",  
    "GroupId": "string"  
}  
,  
"Groups": [ "string" ],  
"Token": "string",  
"UserId": "string"  
},  
"visitorId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

[AttributeFilter \(p. 384\)](#)

Enables filtered searches based on document attributes. You can only provide one attribute filter; however, the `AndAllFilters`, `NotFilter`, and `OrAllFilters` parameters contain a list of other filters.

The `AttributeFilter` parameter enables you to create a set of filtering rules that a document must satisfy to be included in the query results.

Type: [AttributeFilter \(p. 430\)](#) object

Required: No

[DocumentRelevanceOverrideConfigurations \(p. 384\)](#)

Overrides relevance tuning configurations of fields or attributes set at the index level.

If you use this API to override the relevance tuning configured at the index level, but there is no relevance tuning configured at the index level, then Amazon Kendra does not apply any relevance tuning.

If there is relevance tuning configured at the index level, but you do not use this API to override any relevance tuning in the index, then Amazon Kendra uses the relevance tuning that is configured at the index level.

If there is relevance tuning configured for fields at the index level, but you use this API to override only some of these fields, then for the fields you did not override, the importance is set to 1.

Type: Array of [DocumentRelevanceConfiguration \(p. 476\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 500 items.

Required: No

[Facets \(p. 384\)](#)

An array of documents attributes. Amazon Kendra returns a count for each attribute key specified. You can use this information to help narrow the search for your user.

Type: Array of [Facet \(p. 478\)](#) objects

Required: No

[IndexId \(p. 384\)](#)

The unique identifier of the index to search. The identifier is returned in the response from the `CreateIndex` operation.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

[PageNumber \(p. 384\)](#)

Query results are returned in pages the size of the `PageSize` parameter. By default, Amazon Kendra returns the first page of results. Use this parameter to get result pages after the first one.

Type: Integer

Required: No

[PageSize \(p. 384\)](#)

Sets the number of results that are returned in each page of results. The default page size is 10. The maximum number of results returned is 100. If you ask for more than 100 results, only 100 are returned.

Type: Integer

Required: No

[QueryResultTypeFilter \(p. 384\)](#)

Sets the type of query. Only results for the specified query type are returned.

Type: String

Valid Values: DOCUMENT | QUESTION_ANSWER | ANSWER

Required: No

[QueryText \(p. 384\)](#)

The text to search for.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1000.

Pattern: ^\P{C}*\$

Required: Yes

[RequestedDocumentAttributes \(p. 384\)](#)

An array of document attributes to include in the response. No other document attributes are included in the response. By default all document attributes are included in the response.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9_][a-zA-Z0-9_-]*

Required: No

[SortingConfiguration \(p. 384\)](#)

Provides information that determines how the results of the query are sorted. You can set the field that Amazon Kendra should sort the results on, and specify whether the results should be sorted in ascending or descending order. In the case of ties in sorting the results, the results are sorted by relevance.

If you don't provide sorting configuration, the results are sorted by the relevance that Amazon Kendra determines for the result.

Type: [SortingConfiguration \(p. 540\)](#) object

Required: No

[UserContext \(p. 384\)](#)

The user context token or user and group information.

Type: [UserContext \(p. 555\)](#) object

Required: No

[VisitorId \(p. 384\)](#)

Provides an identifier for a specific user. The VisitorId should be a unique identifier, such as a GUID. Don't use personally identifiable information, such as the user's email address, as the VisitorId.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: No

Response Syntax

```
{  
    "FacetResults": [  
        {  
            "DocumentAttributeKey": "string",  
            "DocumentAttributeValueCountPairs": [  
                {  
                    "Count": number,  
                    "DocumentAttributeValue": {  
                        "DateValue": number,  
                        "LongValue": number,  
                        "StringListValue": [ "string" ],  
                        "StringValue": "string"  
                    }  
                }  
            ],  
            "DocumentAttributeValueType": "string"  
        }  
    ],
```

```

"QueryId": "string",
"ResultItems": [
    {
        "AdditionalAttributes": [
            {
                "Key": "string",
                "Value": {
                    "TextWithHighlightsValue": {
                        "Highlights": [
                            {
                                "BeginOffset": number,
                                "EndOffset": number,
                                "TopAnswer": boolean,
                                "Type": "string"
                            }
                        ],
                        "Text": "string"
                    }
                },
                "ValueType": "string"
            }
        ],
        "DocumentAttributes": [
            {
                "Key": "string",
                "Value": {
                    "DateValue": number,
                    "LongValue": number,
                    "StringListValue": [ "string" ],
                    "StringValue": "string"
                }
            }
        ],
        "DocumentExcerpt": {
            "Highlights": [
                {
                    "BeginOffset": number,
                    "EndOffset": number,
                    "TopAnswer": boolean,
                    "Type": "string"
                }
            ],
            "Text": "string"
        },
        "DocumentId": "string",
        "DocumentTitle": {
            "Highlights": [
                {
                    "BeginOffset": number,
                    "EndOffset": number,
                    "TopAnswer": boolean,
                    "Type": "string"
                }
            ],
            "Text": "string"
        },
        "DocumentURI": "string",
        "FeedbackToken": "string",
        "Id": "string",
        "ScoreAttributes": {
            "ScoreConfidence": "string"
        },
        "Type": "string"
    }
],
"TotalNumberOfResults": number

```

}

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

FacetResults ([p. 388](#))

Contains the facet results. A `FacetResult` contains the counts for each attribute key that was specified in the `Facets` input parameter.

Type: Array of [FacetResult](#) ([p. 479](#)) objects

QueryId ([p. 388](#))

The unique identifier for the search. You use `QueryId` to identify the search when using the feedback API.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

ResultItems ([p. 388](#))

The results of the search.

Type: Array of [QueryResultItem](#) ([p. 504](#)) objects

TotalNumberOfResults ([p. 388](#))

The total number of items found by the search; however, you can only retrieve up to 100 items. For example, if the search found 192 items, you can only retrieve the first 100 of the items.

Type: Integer

Errors

For information about the errors that are common to all actions, see [Common Errors](#) ([p. 563](#)).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ServiceQuotaExceededException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

StartDataSourceSyncJob

Starts a synchronization job for a data source. If a synchronization job is already in progress, Amazon Kendra returns a `ResourceInUseException` exception.

Request Syntax

```
{  
    "Id": "string",  
    "IndexId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

Id (p. 392)

The identifier of the data source to synchronize.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: Yes

IndexId (p. 392)

The identifier of the index that contains the data source.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: Yes

Response Syntax

```
{  
    "ExecutionId": "string"  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ExecutionId (p. 392)

Identifies a particular synchronization job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerException

HTTP Status Code: 500

ResourceInUseException

HTTP Status Code: 400

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

StopDataSourceSyncJob

Stops a running synchronization job. You can't stop a scheduled synchronization job.

Request Syntax

```
{  
    "Id": "string",  
    "IndexId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

[Id \(p. 394\)](#)

The identifier of the data source for which to stop the synchronization jobs.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: Yes

[IndexId \(p. 394\)](#)

The identifier of the index that contains the data source.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

SubmitFeedback

Enables you to provide feedback to Amazon Kendra to improve the performance of your index.

SubmitFeedback is currently not supported in the AWS GovCloud (US-West) region.

Request Syntax

```
{  
    "ClickFeedbackItems": [  
        {  
            "ClickTime": number,  
            "ResultId": string  
        }  
    ],  
    "IndexId": string,  
    "QueryId": string,  
    "RelevanceFeedbackItems": [  
        {  
            "RelevanceValue": string,  
            "ResultId": string  
        }  
    ]  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

[ClickFeedbackItems \(p. 396\)](#)

Tells Amazon Kendra that a particular search result link was chosen by the user.

Type: Array of [ClickFeedback \(p. 438\)](#) objects

Required: No

[IndexId \(p. 396\)](#)

The identifier of the index that was queried.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

[QueryId \(p. 396\)](#)

The identifier of the specific query for which you are submitting feedback. The query ID is returned in the response to the `Query` operation.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

[RelevanceFeedbackItems \(p. 396\)](#)

Provides Amazon Kendra with relevant or not relevant feedback for whether a particular item was relevant to the search.

Type: Array of [RelevanceFeedback \(p. 510\)](#) objects

Required: No

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

InternalServerException

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ResourceUnavailableException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)

- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

TagResource

Adds the specified tag to the specified index, FAQ, or data source resource. If the tag already exists, the existing value is replaced with the new value.

Request Syntax

```
{  
    "ResourceARN": "string",  
    "Tags": [  
        {  
            "Key": "string",  
            "Value": "string"  
        }  
    ]  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

ResourceARN ([p. 399](#))

The Amazon Resource Name (ARN) of the index, FAQ, or data source to tag.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1011.

Required: Yes

Tags ([p. 399](#))

A list of tag keys to add to the index, FAQ, or data source. If a tag already exists, the existing value is replaced with the new value.

Type: Array of [Tag \(p. 548\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 200 items.

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceUnavailableException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

UntagResource

Removes a tag from an index, FAQ, or a data source.

Request Syntax

```
{  
    "ResourceARN": "string",  
    "TagKeys": [ "string" ]  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

[ResourceARN \(p. 401\)](#)

The Amazon Resource Name (ARN) of the index, FAQ, or data source to remove the tag from.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1011.

Required: Yes

[TagKeys \(p. 401\)](#)

A list of tag keys to remove from the index, FAQ, or data source. If a tag key does not exist on the resource, it is ignored.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 200 items.

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

[AccessDeniedException](#)

HTTP Status Code: 400

[InternalServerException](#)

HTTP Status Code: 500

ResourceUnavailableException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

UpdateDataSource

Updates an existing Amazon Kendra data source.

Request Syntax

```
{
    "Configuration": {
        "ConfluenceConfiguration": {
            "AttachmentConfiguration": {
                "AttachmentFieldMappings": [
                    {
                        "DataSourceFieldName": "string",
                        "DateFormat": "string",
                        "IndexFieldName": "string"
                    }
                ],
                "CrawlAttachments": boolean
            },
            "BlogConfiguration": {
                "BlogFieldMappings": [
                    {
                        "DataSourceFieldName": "string",
                        "DateFormat": "string",
                        "IndexFieldName": "string"
                    }
                ]
            },
            "ExclusionPatterns": [ "string" ],
            "InclusionPatterns": [ "string" ],
            "PageConfiguration": {
                "PageFieldMappings": [
                    {
                        "DataSourceFieldName": "string",
                        "DateFormat": "string",
                        "IndexFieldName": "string"
                    }
                ]
            },
            "SecretArn": "string",
            "ServerUrl": "string",
            "SpaceConfiguration": {
                "CrawlArchivedSpaces": boolean,
                "CrawlPersonalSpaces": boolean,
                "ExcludeSpaces": [ "string" ],
                "IncludeSpaces": [ "string" ],
                "SpaceFieldMappings": [
                    {
                        "DataSourceFieldName": "string",
                        "DateFormat": "string",
                        "IndexFieldName": "string"
                    }
                ]
            },
            "Version": "string",
            "VpcConfiguration": {
                "SecurityGroupIds": [ "string" ],
                "SubnetIds": [ "string" ]
            }
        },
        "DatabaseConfiguration": {
            "AclConfiguration": {
                "AllowedGroupsColumnName": "string"
            }
        }
    }
}
```

```

},
"ColumnConfiguration": {
    "ChangeDetectingColumns": [ "string" ],
    "DocumentColumnName": "string",
    "DocumentIdColumnName": "string",
    "DocumentTitleColumnName": "string",
    "FieldMappings": [
        {
            "DataSourceFieldName": "string",
            "DateFormat": "string",
            "IndexFieldName": "string"
        }
    ]
},
"ConnectionConfiguration": {
    "DatabaseHost": "string",
    "DatabaseName": "string",
    "DatabasePort": number,
    "SecretArn": "string",
    "TableName": "string"
},
"DatabaseEngineType": "string",
"SqlConfiguration": {
    "QueryIdentifiersEnclosingOption": "string"
},
"VpcConfiguration": {
    "SecurityGroupIds": [ "string" ],
    "SubnetIds": [ "string" ]
},
},
"GoogleDriveConfiguration": {
    "ExcludeMimeTypes": [ "string" ],
    "ExcludeSharedDrives": [ "string" ],
    "ExcludeUserAccounts": [ "string" ],
    "ExclusionPatterns": [ "string" ],
    "FieldMappings": [
        {
            "DataSourceFieldName": "string",
            "DateFormat": "string",
            "IndexFieldName": "string"
        }
    ],
    "InclusionPatterns": [ "string" ],
    "SecretArn": "string"
},
"OneDriveConfiguration": {
    "DisableLocalGroups": boolean,
    "ExclusionPatterns": [ "string" ],
    "FieldMappings": [
        {
            "DataSourceFieldName": "string",
            "DateFormat": "string",
            "IndexFieldName": "string"
        }
    ],
    "InclusionPatterns": [ "string" ],
    "OneDriveUsers": {
        "OneDriveUserList": [ "string" ],
        "OneDriveUserS3Path": {
            "Bucket": "string",
            "Key": "string"
        }
    },
    "SecretArn": "string",
    "TenantDomain": "string"
}
},

```

```

"S3Configuration": {
    "AccessControlListConfiguration": {
        "KeyPath": "string"
    },
    "BucketName": "string",
    "DocumentsMetadataConfiguration": {
        "S3Prefix": "string"
    },
    "ExclusionPatterns": [ "string" ],
    "InclusionPatterns": [ "string" ],
    "InclusionPrefixes": [ "string" ]
},
"SalesforceConfiguration": {
    "ChatterFeedConfiguration": {
        "DocumentDataFieldName": "string",
        "DocumentTitleFieldName": "string",
        "FieldMappings": [
            {
                "DataSourceFieldName": "string",
                "DateFormat": "string",
                "IndexFieldName": "string"
            }
        ],
        "IncludeFilterTypes": [ "string" ]
},
    "CrawlAttachments": boolean,
    "ExcludeAttachmentFilePatterns": [ "string" ],
    "IncludeAttachmentFilePatterns": [ "string" ],
    "KnowledgeArticleConfiguration": {
        "CustomKnowledgeArticleTypeConfigurations": [
            {
                "DocumentDataFieldName": "string",
                "DocumentTitleFieldName": "string",
                "FieldMappings": [
                    {
                        "DataSourceFieldName": "string",
                        "DateFormat": "string",
                        "IndexFieldName": "string"
                    }
                ],
                "Name": "string"
            }
        ],
        "IncludedStates": [ "string" ],
        "StandardKnowledgeArticleTypeConfiguration": {
            "DocumentDataFieldName": "string",
            "DocumentTitleFieldName": "string",
            "FieldMappings": [
                {
                    "DataSourceFieldName": "string",
                    "DateFormat": "string",
                    "IndexFieldName": "string"
                }
            ]
        }
    },
    "SecretArn": "string",
    "ServerUrl": "string",
    "StandardObjectAttachmentConfiguration": {
        "DocumentTitleFieldName": "string",
        "FieldMappings": [
            {
                "DataSourceFieldName": "string",
                "DateFormat": "string",
                "IndexFieldName": "string"
            }
        ]
    }
},
"SecretArn": "string",
"ServerUrl": "string",
"StandardObjectAttachmentConfiguration": {
    "DocumentTitleFieldName": "string",
    "FieldMappings": [
        {
            "DataSourceFieldName": "string",
            "DateFormat": "string",
            "IndexFieldName": "string"
        }
    ]
}

```

```

        ],
    },
    "StandardObjectConfigurations": [
        {
            "DocumentDataFieldName": "string",
            "DocumentTitleFieldName": "string",
            "FieldMappings": [
                {
                    "DataSourceFieldName": "string",
                    "DateFormat": "string",
                    "IndexFieldName": "string"
                }
            ],
            "Name": "string"
        }
    ]
},
"ServiceNowConfiguration": {
    "AuthenticationType": "string",
    "HostUrl": "string",
    "KnowledgeArticleConfiguration": {
        "CrawlAttachments": boolean,
        "DocumentDataFieldName": "string",
        "DocumentTitleFieldName": "string",
        "ExcludeAttachmentFilePatterns": [ "string" ],
        "FieldMappings": [
            {
                "DataSourceFieldName": "string",
                "DateFormat": "string",
                "IndexFieldName": "string"
            }
        ],
        "FilterQuery": "string",
        "IncludeAttachmentFilePatterns": [ "string" ]
    },
    "SecretArn": "string",
    "ServiceCatalogConfiguration": {
        "CrawlAttachments": boolean,
        "DocumentDataFieldName": "string",
        "DocumentTitleFieldName": "string",
        "ExcludeAttachmentFilePatterns": [ "string" ],
        "FieldMappings": [
            {
                "DataSourceFieldName": "string",
                "DateFormat": "string",
                "IndexFieldName": "string"
            }
        ],
        "IncludeAttachmentFilePatterns": [ "string" ]
    },
    "ServiceNowBuildVersion": "string"
},
"SharePointConfiguration": {
    "CrawlAttachments": boolean,
    "DisableLocalGroups": boolean,
    "DocumentTitleFieldName": "string",
    "ExclusionPatterns": [ "string" ],
    "FieldMappings": [
        {
            "DataSourceFieldName": "string",
            "DateFormat": "string",
            "IndexFieldName": "string"
        }
    ],
    "InclusionPatterns": [ "string" ],
    "SecretArn": "string",

```

```

    "SharePointVersion": "string",
    "SslCertificateS3Path": {
        "Bucket": "string",
        "Key": "string"
    },
    "Urls": [ "string" ],
    "UseChangeLog": boolean,
    "VpcConfiguration": {
        "SecurityGroupIds": [ "string" ],
        "SubnetIds": [ "string" ]
    }
},
"WebCrawlerConfiguration": {
    "AuthenticationConfiguration": {
        "BasicAuthentication": [
            {
                "Credentials": "string",
                "Host": "string",
                "Port": number
            }
        ]
    },
    "CrawlDepth": number,
    "MaxContentSizePerPageInMegaBytes": number,
    "MaxLinksPerPage": number,
    "MaxUrlsPerMinuteCrawlRate": number,
    "ProxyConfiguration": {
        "Credentials": "string",
        "Host": "string",
        "Port": number
    },
    "UrlExclusionPatterns": [ "string" ],
    "UrlInclusionPatterns": [ "string" ],
    "Urls": {
        "SeedUrlConfiguration": {
            "SeedUrls": [ "string" ],
            "WebCrawlerMode": "string"
        },
        "SiteMapsConfiguration": {
            "SiteMaps": [ "string" ]
        }
    }
},
"WorkDocsConfiguration": {
    "CrawlComments": boolean,
    "ExclusionPatterns": [ "string" ],
    "FieldMappings": [
        {
            "DataSourceFieldName": "string",
            "DateFormat": "string",
            "IndexFieldName": "string"
        }
    ],
    "InclusionPatterns": [ "string" ],
    "OrganizationId": "string",
    "UseChangeLog": boolean
},
},
"Description": "string",
"Id": "string",
"IndexId": "string",
"LanguageCode": "string",
"Name": "string",
"RoleArn": "string",
"Schedule": "string"
}

```

}

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

Configuration (p. 403)

Configuration information for an Amazon Kendra data source.

Type: [DataSourceConfiguration \(p. 457\)](#) object

Required: No

Description (p. 403)

The new description for the data source.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1000.

Pattern: ^\P{C}*\$

Required: No

Id (p. 403)

The unique identifier of the data source to update.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: Yes

IndexId (p. 403)

The identifier of the index that contains the data source to update.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: Yes

LanguageCode (p. 403)

The code for a language. This allows you to support a language for all documents when updating the data source. English is supported by default. For more information on supported languages, including their codes, see [Adding documents in languages other than English](#).

Type: String

Length Constraints: Minimum length of 2. Maximum length of 10.

Pattern: [a-zA-Z_-]*

Required: No

Name (p. 403)

The name of the data source to update. The name of the data source can't be updated. To rename a data source you must delete the data source and re-create it.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1000.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: No

RoleArn (p. 403)

The Amazon Resource Name (ARN) of the new role to use when the data source is accessing resources on your behalf.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: arn:[a-zA-Z0-9-\.]{1,63}:[a-zA-Z0-9-\.]{0,63}:[a-zA-Z0-9-\.]{0,63}:[a-zA-Z0-9-\.]{0,63}:[^/].{0,1023}

Required: No

Schedule (p. 403)

The new update schedule for the data source.

Type: String

Required: No

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

UpdateIndex

Updates an existing Amazon Kendra index.

Request Syntax

```
{  
    "CapacityUnits": {  
        "QueryCapacityUnits": number,  
        "StorageCapacityUnits": number  
    },  
    "Description": "string",  
    "DocumentMetadataConfigurationUpdates": [  
        {  
            "Name": "string",  
            "Relevance": {  
                "Duration": "string",  
                "Freshness": boolean,  
                "Importance": number,  
                "RankOrder": "string",  
                "ValueImportanceMap": {  
                    "string": number  
                }  
            },  
            "Search": {  
                "Displayable": boolean,  
                "Facetable": boolean,  
                "Searchable": boolean,  
                "Sortable": boolean  
            },  
            "Type": "string"  
        }  
    ],  
    "Id": "string",  
    "Name": "string",  
    "RoleArn": "string",  
    "UserContextPolicy": "string",  
    "UserGroupResolutionConfiguration": {  
        "UserGroupResolutionMode": "string"  
    },  
    "UserTokenConfigurations": [  
        {  
            "JsonTokenTypeConfiguration": {  
                "GroupAttributeField": "string",  
                "UserNameAttributeField": "string"  
            },  
            "JwtTokenTypeConfiguration": {  
                "ClaimRegex": "string",  
                "GroupAttributeField": "string",  
                "Issuer": "string",  
                "KeyLocation": "string",  
                "SecretManagerArn": "string",  
                "URL": "string",  
                "UserNameAttributeField": "string"  
            }  
        }  
    ]  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

[CapacityUnits \(p. 411\)](#)

Sets the number of additional storage and query capacity units that should be used by the index. You can change the capacity of the index up to 5 times per day.

If you are using extra storage units, you can't reduce the storage capacity below that required to meet the storage needs for your index.

Type: [CapacityUnitsConfiguration \(p. 437\)](#) object

Required: No

[Description \(p. 411\)](#)

A new description for the index.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1000.

Pattern: ^\P{C}*\$

Required: No

[DocumentMetadataConfigurationUpdates \(p. 411\)](#)

The document metadata to update.

Type: Array of [DocumentMetadataConfiguration \(p. 475\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 500 items.

Required: No

[Id \(p. 411\)](#)

The identifier of the index to update.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

[Name \(p. 411\)](#)

The name of the index to update.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1000.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: No

[RoleArn \(p. 411\)](#)

A new IAM role that gives Amazon Kendra permission to access your Amazon CloudWatch logs.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: `arn:[a-z0-9-\.]{1,63}:[a-z0-9-\.]{0,63}:[a-z0-9-\.]{0,63}:[a-z0-9-\.]{0,63}:[^/].{0,1023}`

Required: No

[UserContextPolicy \(p. 411\)](#)

The user context policy.

Type: String

Valid Values: `ATTRIBUTE_FILTER` | `USER_TOKEN`

Required: No

[UserGroupResolutionConfiguration \(p. 411\)](#)

Enables fetching access levels of groups and users from an AWS Single Sign-On identity source. To configure this, see [UserGroupResolutionConfiguration](#).

Type: [UserGroupResolutionConfiguration \(p. 557\)](#) object

Required: No

[UserTokenConfigurations \(p. 411\)](#)

The user token configuration.

Type: Array of [UserTokenConfiguration \(p. 558\)](#) objects

Array Members: Maximum number of 1 item.

Required: No

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ServiceQuotaExceededException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

UpdateQuerySuggestionsBlockList

Updates a block list used for query suggestions for an index.

Updates to a block list might not take effect right away. Amazon Kendra needs to refresh the entire suggestions list to apply any updates to the block list. Other changes not related to the block list apply immediately.

If a block list is updating, then you need to wait for the first update to finish before submitting another update.

Amazon Kendra supports partial updates, so you only need to provide the fields you want to update.

`UpdateQuerySuggestionsBlockList` is currently not supported in the AWS GovCloud (US-West) region.

Request Syntax

```
{  
    "Description": "string",  
    "Id": "string",  
    "IndexId": "string",  
    "Name": "string",  
    "RoleArn": "string",  
    "SourceS3Path": {  
        "Bucket": "string",  
        "Key": "string"  
    }  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

Description (p. 415)

The description for a block list.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1000.

Pattern: ^\P{C}*\$

Required: No

Id (p. 415)

The unique identifier of a block list.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

[IndexId \(p. 415\)](#)

The identifier of the index for a block list.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

[Name \(p. 415\)](#)

The name of a block list.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z0-9](-*[a-zA-Z0-9])*

Required: No

[RoleArn \(p. 415\)](#)

The IAM (Identity and Access Management) role used to access the block list text file in S3.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: arn:[a-z0-9-\.\.]{1,63}:[a-z0-9-\.\.]{0,63}:[a-z0-9-\.\.]{0,63}:[a-z0-9-\.\.]{0,63}:[^/].{0,1023}

Required: No

[SourceS3Path \(p. 415\)](#)

The S3 path where your block list text file sits in S3.

If you update your block list and provide the same path to the block list text file in S3, then Amazon Kendra reloads the file to refresh the block list. Amazon Kendra does not automatically refresh your block list. You need to call the `UpdateQuerySuggestionsBlockList` API to refresh your block list.

If you update your block list, then Amazon Kendra asynchronously refreshes all query suggestions with the latest content in the S3 file. This means changes might not take effect immediately.

Type: [S3Path \(p. 513\)](#) object

Required: No

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerException

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

UpdateQuerySuggestionsConfig

Updates the settings of query suggestions for an index.

Amazon Kendra supports partial updates, so you only need to provide the fields you want to update.

If an update is currently processing (i.e. 'happening'), you need to wait for the update to finish before making another update.

Updates to query suggestions settings might not take effect right away. The time for your updated settings to take effect depends on the updates made and the number of search queries in your index.

You can still enable/disable query suggestions at any time.

`UpdateQuerySuggestionsConfig` is currently not supported in the AWS GovCloud (US-West) region.

Request Syntax

```
{  
    "IncludeQueriesWithoutUserInformation": boolean,  
    "IndexId": "string",  
    "MinimumNumberOfQueryingUsers": number,  
    "MinimumQueryCount": number,  
    "Mode": "string",  
    "QueryLogLookBackWindowInDays": number  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

[IncludeQueriesWithoutUserInformation \(p. 418\)](#)

`TRUE` to include queries without user information (i.e. all queries, irrespective of the user), otherwise `FALSE` to only include queries with user information.

If you pass user information to Amazon Kendra along with the queries, you can set this flag to `FALSE` and instruct Amazon Kendra to only consider queries with user information.

If you set to `FALSE`, Amazon Kendra only considers queries searched at least `MinimumQueryCount` times across `MinimumNumberOfQueryingUsers` unique users for suggestions.

If you set to `TRUE`, Amazon Kendra ignores all user information and learns from all queries.

Type: Boolean

Required: No

[IndexId \(p. 418\)](#)

The identifier of the index you want to update query suggestions settings for.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

[MinimumNumberOfQueryingUsers \(p. 418\)](#)

The minimum number of unique users who must search a query in order for the query to be eligible to suggest to your users.

Increasing this number might decrease the number of suggestions. However, this ensures a query is searched by many users and is truly popular to suggest to users.

How you tune this setting depends on your specific needs.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 10000.

Required: No

[MinimumQueryCount \(p. 418\)](#)

The the minimum number of times a query must be searched in order to be eligible to suggest to your users.

Decreasing this number increases the number of suggestions. However, this affects the quality of suggestions as it sets a low bar for a query to be considered popular to suggest to users.

How you tune this setting depends on your specific needs.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 10000.

Required: No

[Mode \(p. 418\)](#)

Set the mode to `ENABLED` or `LEARN_ONLY`.

By default, Amazon Kendra enables query suggestions. `LEARN_ONLY` mode allows you to turn off query suggestions. You can to update this at any time.

In `LEARN_ONLY` mode, Amazon Kendra continues to learn from new queries to keep suggestions up to date for when you are ready to switch to `ENABLED` mode again.

Type: String

Valid Values: `ENABLED` | `LEARN_ONLY`

Required: No

[QueryLogLookBackWindowInDays \(p. 418\)](#)

How recent your queries are in your query log time window.

The time window is the number of days from current day to past days.

By default, Amazon Kendra sets this to 180.

Type: Integer

Required: No

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerException

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

UpdateThesaurus

Updates a thesaurus file associated with an index.

Request Syntax

```
{  
    "Description": "string",  
    "Id": "string",  
    "IndexId": "string",  
    "Name": "string",  
    "RoleArn": "string",  
    "SourceS3Path": {  
        "Bucket": "string",  
        "Key": "string"  
    }  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 565\)](#).

The request accepts the following data in JSON format.

Description (p. 421)

The updated description of the thesaurus.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1000.

Pattern: ^\P{C}*\$

Required: No

Id (p. 421)

The identifier of the thesaurus to update.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: Yes

IndexId (p. 421)

The identifier of the index associated with the thesaurus to update.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: Yes

[Name \(p. 421\)](#)

The updated name of the thesaurus.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: No

[RoleArn \(p. 421\)](#)

The updated role ARN of the thesaurus.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: arn:[a-zA-Z0-9-\.]{1,63}:[a-zA-Z0-9-\.]{0,63}:[a-zA-Z0-9-\.]{0,63}:[a-zA-Z0-9-\.]{0,63}:[^/].{0,1023}

Required: No

[SourceS3Path \(p. 421\)](#)

Information required to find a specific file in an Amazon S3 bucket.

Type: [S3Path \(p. 513\)](#) object

Required: No

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 563\)](#).

AccessDeniedException

HTTP Status Code: 400

ConflictException

HTTP Status Code: 400

InternalServerError

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 400

ThrottlingException

HTTP Status Code: 400

ValidationException

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

Data Types

The following data types are supported:

- [AccessControlListConfiguration \(p. 426\)](#)
- [AclConfiguration \(p. 427\)](#)
- [AdditionalResultAttribute \(p. 428\)](#)
- [AdditionalResultAttributeValue \(p. 429\)](#)
- [AttributeFilter \(p. 430\)](#)
- [AuthenticationConfiguration \(p. 432\)](#)
- [BasicAuthenticationConfiguration \(p. 433\)](#)
- [BatchDeleteDocumentResponseFailedDocument \(p. 434\)](#)
- [BatchGetDocumentStatusResponseError \(p. 435\)](#)
- [BatchPutDocumentResponseFailedDocument \(p. 436\)](#)
- [CapacityUnitsConfiguration \(p. 437\)](#)
- [ClickFeedback \(p. 438\)](#)
- [ColumnConfiguration \(p. 439\)](#)
- [ConfluenceAttachmentConfiguration \(p. 441\)](#)
- [ConfluenceAttachmentToIndexFieldMapping \(p. 442\)](#)
- [ConfluenceBlogConfiguration \(p. 443\)](#)
- [ConfluenceBlogToIndexFieldMapping \(p. 444\)](#)
- [ConfluenceConfiguration \(p. 445\)](#)
- [ConfluencePageConfiguration \(p. 448\)](#)
- [ConfluencePageToIndexFieldMapping \(p. 449\)](#)
- [ConfluenceSpaceConfiguration \(p. 450\)](#)
- [ConfluenceSpaceToIndexFieldMapping \(p. 452\)](#)
- [ConnectionConfiguration \(p. 453\)](#)
- [DatabaseConfiguration \(p. 455\)](#)

- [DataSourceConfiguration \(p. 457\)](#)
- [DataSourceGroup \(p. 459\)](#)
- [DataSourceSummary \(p. 460\)](#)
- [DataSourceSyncJob \(p. 462\)](#)
- [DataSourceSyncJobMetrics \(p. 464\)](#)
- [DataSourceSyncJobMetricTarget \(p. 466\)](#)
- [DataSourceToIndexFieldMapping \(p. 467\)](#)
- [DataSourceVpcConfiguration \(p. 468\)](#)
- [Document \(p. 469\)](#)
- [DocumentAttribute \(p. 471\)](#)
- [DocumentAttributeValue \(p. 472\)](#)
- [DocumentAttributeValueCountPair \(p. 473\)](#)
- [DocumentInfo \(p. 474\)](#)
- [DocumentMetadataConfiguration \(p. 475\)](#)
- [DocumentRelevanceConfiguration \(p. 476\)](#)
- [DocumentsMetadataConfiguration \(p. 477\)](#)
- [Facet \(p. 478\)](#)
- [FacetResult \(p. 479\)](#)
- [FaqStatistics \(p. 480\)](#)
- [FaqSummary \(p. 481\)](#)
- [GoogleDriveConfiguration \(p. 483\)](#)
- [GroupMembers \(p. 485\)](#)
- [GroupOrderingIdSummary \(p. 486\)](#)
- [GroupSummary \(p. 488\)](#)
- [HierarchicalPrincipal \(p. 489\)](#)
- [Highlight \(p. 490\)](#)
- [IndexConfigurationSummary \(p. 491\)](#)
- [IndexStatistics \(p. 493\)](#)
- [JsonTokenTypeConfiguration \(p. 494\)](#)
- [JwtTokenTypeConfiguration \(p. 495\)](#)
- [MemberGroup \(p. 497\)](#)
- [MemberUser \(p. 498\)](#)
- [OneDriveConfiguration \(p. 499\)](#)
- [OneDriveUsers \(p. 501\)](#)
- [Principal \(p. 502\)](#)
- [ProxyConfiguration \(p. 503\)](#)
- [QueryResultItem \(p. 504\)](#)
- [QuerySuggestionsBlockListSummary \(p. 506\)](#)
- [Relevance \(p. 508\)](#)
- [RelevanceFeedback \(p. 510\)](#)
- [S3DataSourceConfiguration \(p. 511\)](#)
- [S3Path \(p. 513\)](#)
- [SalesforceChatterFeedConfiguration \(p. 514\)](#)
- [SalesforceConfiguration \(p. 516\)](#)
- [SalesforceCustomKnowledgeArticleTypeConfiguration \(p. 519\)](#)
- [SalesforceKnowledgeArticleConfiguration \(p. 521\)](#)

- [SalesforceStandardKnowledgeArticleTypeConfiguration \(p. 522\)](#)
- [SalesforceStandardObjectAttachmentConfiguration \(p. 523\)](#)
- [SalesforceStandardObjectConfiguration \(p. 524\)](#)
- [ScoreAttributes \(p. 526\)](#)
- [Search \(p. 527\)](#)
- [SeedUrlConfiguration \(p. 528\)](#)
- [ServerSideEncryptionConfiguration \(p. 529\)](#)
- [ServiceNowConfiguration \(p. 530\)](#)
- [ServiceNowKnowledgeArticleConfiguration \(p. 532\)](#)
- [ServiceNowServiceCatalogConfiguration \(p. 534\)](#)
- [SharePointConfiguration \(p. 536\)](#)
- [SiteMapsConfiguration \(p. 539\)](#)
- [SortingConfiguration \(p. 540\)](#)
- [SqlConfiguration \(p. 542\)](#)
- [Status \(p. 543\)](#)
- [Suggestion \(p. 544\)](#)
- [SuggestionHighlight \(p. 545\)](#)
- [SuggestionTextWithHighlights \(p. 546\)](#)
- [SuggestionValue \(p. 547\)](#)
- [Tag \(p. 548\)](#)
- [TextDocumentStatistics \(p. 549\)](#)
- [TextWithHighlights \(p. 550\)](#)
- [ThesaurusSummary \(p. 551\)](#)
- [TimeRange \(p. 553\)](#)
- [Urls \(p. 554\)](#)
- [UserContext \(p. 555\)](#)
- [UserGroupResolutionConfiguration \(p. 557\)](#)
- [UserTokenConfiguration \(p. 558\)](#)
- [WebCrawlerConfiguration \(p. 559\)](#)
- [WorkDocsConfiguration \(p. 562\)](#)

AccessControlListConfiguration

Access Control List files for the documents in a data source. For the format of the file, see [Access control for S3 data sources](#).

Contents

KeyPath

Path to the AWS S3 bucket that contains the ACL files.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

AclConfiguration

Provides information about the column that should be used for filtering the query response by groups.

Contents

AllowedGroupsColumnName

A list of groups, separated by semi-colons, that filters a query response based on user context. The document is only returned to users that are in one of the groups specified in the `UserContext` field of the `Query` operation.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: `^[a-zA-Z][a-zA-Z0-9_]*$`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

AdditionalResultAttribute

An attribute returned from an index query.

Contents

Key

The key that identifies the attribute.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

Value

An object that contains the attribute value.

Type: [AdditionalResultAttributeValue \(p. 429\)](#) object

Required: Yes

ValueType

The data type of the Value property.

Type: String

Valid Values: TEXT_WITH_HIGHLIGHTS_VALUE

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

AdditionalResultAttributeValue

An attribute returned with a document from a search.

Contents

TextWithHighlightsValue

The text associated with the attribute and information about the highlight to apply to the text.

Type: [TextWithHighlights \(p. 550\)](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

AttributeFilter

Provides filtering the query results based on document attributes.

When you use the `AndAllFilters` or `OrAllFilters`, filters you can use 2 layers under the first attribute filter. For example, you can use:

```
<AndAllFilters>
  1. <OrAllFilters>
  2. <EqualsTo>
```

If you use more than 2 layers, you receive a `ValidationException` exception with the message "AttributeFilter cannot have a depth of more than 2."

If you use more than 10 attribute filters in a given list for `AndAllFilters` or `OrAllFilters`, you receive a `ValidationException` with the message "AttributeFilter cannot have a length of more than 10".

Contents

AndAllFilters

Performs a logical AND operation on all supplied filters.

Type: Array of [AttributeFilter \(p. 430\)](#) objects

Required: No

ContainsAll

Returns true when a document contains all of the specified document attributes. This filter is only applicable to `StringListValue` metadata.

Type: [DocumentAttribute \(p. 471\)](#) object

Required: No

ContainsAny

Returns true when a document contains any of the specified document attributes. This filter is only applicable to `StringListValue` metadata.

Type: [DocumentAttribute \(p. 471\)](#) object

Required: No

EqualsTo

Performs an equals operation on two document attributes.

Type: [DocumentAttribute \(p. 471\)](#) object

Required: No

GreaterThan

Performs a greater than operation on two document attributes. Use with a document attribute of type `Date` or `Long`.

Type: [DocumentAttribute \(p. 471\)](#) object

Required: No

GreaterThanOrEquals

Performs a greater or equals than operation on two document attributes. Use with a document attribute of type Date or Long.

Type: [DocumentAttribute \(p. 471\)](#) object

Required: No

LessThan

Performs a less than operation on two document attributes. Use with a document attribute of type Date or Long.

Type: [DocumentAttribute \(p. 471\)](#) object

Required: No

LessThanOrEquals

Performs a less than or equals operation on two document attributes. Use with a document attribute of type Date or Long.

Type: [DocumentAttribute \(p. 471\)](#) object

Required: No

NotFilter

Performs a logical NOT operation on all supplied filters.

Type: [AttributeFilter \(p. 430\)](#) object

Required: No

OrAllFilters

Performs a logical OR operation on all supplied filters.

Type: Array of [AttributeFilter \(p. 430\)](#) objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

AuthenticationConfiguration

Provides the configuration information to connect to websites that require user authentication.

Contents

BasicAuthentication

The list of configuration information that's required to connect to and crawl a website host using basic authentication credentials.

The list includes the name and port number of the website host.

Type: Array of [BasicAuthenticationConfiguration \(p. 433\)](#) objects

Array Members: Minimum number of 0 items. Maximum number of 10 items.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

BasicAuthenticationConfiguration

Provides the configuration information to connect to websites that require basic user authentication.

Contents

Credentials

Your secret ARN, which you can create in [AWS Secrets Manager](#)

You use a secret if basic authentication credentials are required to connect to a website. The secret stores your credentials of user name and password.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: `arn:[a-z0-9-\.]{1,63}:[a-z0-9-\.]{0,63}:[a-z0-9-\.]{0,63}:[a-z0-9-\.]{0,63}:[^/].{0,1023}`

Required: Yes

Host

The name of the website host you want to connect to using authentication credentials.

For example, the host name of <https://a.example.com/page1.html> is "a.example.com".

Type: String

Length Constraints: Minimum length of 1. Maximum length of 253.

Pattern: `([^\s]*)`

Required: Yes

Port

The port number of the website host you want to connect to using authentication credentials.

For example, the port for <https://a.example.com/page1.html> is 443, the standard port for HTTPS.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 65535.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

BatchDeleteDocumentResponseFailedDocument

Provides information about documents that could not be removed from an index by the BatchDeleteDocument operation.

Contents

ErrorCode

The error code for why the document couldn't be removed from the index.

Type: String

Valid Values: InternalError | InvalidRequest

Required: No

ErrorMessage

An explanation for why the document couldn't be removed from the index.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: ^\P{C}*\$

Required: No

Id

The identifier of the document that couldn't be removed from the index.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

BatchGetDocumentStatusResponseError

Provides a response when the status of a document could not be retrieved.

Contents

DocumentId

The unique identifier of the document whose status could not be retrieved.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: No

ErrorCode

Indicates the source of the error.

Type: String

Valid Values: InternalError | InvalidRequest

Required: No

ErrorMessage

States that the API could not get the status of a document. This could be because the request is not valid or there is a system error.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: ^\P{C}*\$

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

BatchPutDocumentResponseFailedDocument

Provides information about a document that could not be indexed.

Contents

ErrorCode

The type of error that caused the document to fail to be indexed.

Type: String

Valid Values: InternalError | InvalidRequest

Required: No

ErrorMessage

A description of the reason why the document could not be indexed.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: ^\P{C}*\$

Required: No

Id

The unique identifier of the document.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

CapacityUnitsConfiguration

Specifies capacity units configured for your enterprise edition index. You can add and remove capacity units to tune an index to your requirements.

Contents

QueryCapacityUnits

The amount of extra query capacity for an index and [GetQuerySuggestions](#) capacity.

A single extra capacity unit for an index provides 0.1 queries per second or approximately 8,000 queries per day.

[GetQuerySuggestions](#) capacity is five times the provisioned query capacity for an index, or the base capacity of 2.5 calls per second, whichever is higher. For example, the base capacity for an index is 0.1 queries per second, and [GetQuerySuggestions](#) capacity has a base of 2.5 calls per second. If you add another 0.1 queries per second to total 0.2 queries per second for an index, the [GetQuerySuggestions](#) capacity is 2.5 calls per second (higher than five times 0.2 queries per second).

Type: Integer

Valid Range: Minimum value of 0.

Required: Yes

StorageCapacityUnits

The amount of extra storage capacity for an index. A single capacity unit provides 30 GB of storage space or 100,000 documents, whichever is reached first.

Type: Integer

Valid Range: Minimum value of 0.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ClickFeedback

Gathers information about when a particular result was clicked by a user. Your application uses the `SubmitFeedback` operation to provide click information.

Contents

ClickTime

The Unix timestamp of the date and time that the result was clicked.

Type: `Timestamp`

Required: Yes

ResultId

The unique identifier of the search result that was clicked.

Type: `String`

Length Constraints: Minimum length of 1. Maximum length of 73.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ColumnConfiguration

Provides information about how Amazon Kendra should use the columns of a database in an index.

Contents

ChangeDetectingColumns

One to five columns that indicate when a document in the database has changed.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 5 items.

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z][a-zA-Z0-9_]*\$

Required: Yes

DocumentDataColumnName

The column that contains the contents of the document.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z][a-zA-Z0-9_]*\$

Required: Yes

DocumentIdColumnName

The column that provides the document's unique identifier.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z][a-zA-Z0-9_]*\$

Required: Yes

DocumentTitleColumnName

The column that contains the title of the document.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z][a-zA-Z0-9_]*\$

Required: No

FieldMappings

An array of objects that map database column names to the corresponding fields in an index. You must first create the fields in the index using the `UpdateIndex` operation.

Type: Array of [DataSourceToIndexFieldMapping](#) (p. 467) objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ConfluenceAttachmentConfiguration

Specifies the attachment settings for the Confluence data source. Attachment settings are optional, if you don't specify settings attachments, Amazon Kendra won't index them.

Contents

AttachmentFieldMappings

Defines how attachment metadata fields should be mapped to index fields. Before you can map a field, you must first create an index field with a matching type using the console or the `UpdateIndex` operation.

If you specify the `AttachmentFieldMappings` parameter, you must specify at least one field mapping.

Type: Array of [ConfluenceAttachmentToIndexFieldMapping \(p. 442\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 11 items.

Required: No

CrawlAttachments

Indicates whether Amazon Kendra indexes attachments to the pages and blogs in the Confluence data source.

Type: Boolean

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ConfluenceAttachmentToIndexFieldMapping

Defines the mapping between a field in the Confluence data source to a Amazon Kendra index field.

You must first create the index field using the `UpdateIndex` operation.

Contents

DataSourceFieldName

The name of the field in the data source.

You must first create the index field using the `UpdateIndex` operation.

Type: String

Valid Values: AUTHOR | CONTENT_TYPE | CREATED_DATE | DISPLAY_URL | FILE_SIZE | ITEM_TYPE | PARENT_ID | SPACE_KEY | SPACE_NAME | URL | VERSION

Required: No

DateFieldFormat

The format for date fields in the data source. If the field specified in `DataSourceFieldName` is a date field you must specify the date format. If the field is not a date field, an exception is thrown.

Type: String

Length Constraints: Minimum length of 4. Maximum length of 40.

Pattern: `^(?!\\s).*(?<!\\s)$`

Required: No

IndexFieldName

The name of the index field to map to the Confluence data source field. The index field type must match the Confluence field type.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 30.

Pattern: `^\\P{C}*$`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ConfluenceBlogConfiguration

Specifies the blog settings for the Confluence data source. Blogs are always indexed unless filtered from the index by the `ExclusionPatterns` or `InclusionPatterns` fields in the `ConfluenceConfiguration` type.

Contents

BlogFieldMappings

Defines how blog metadata fields should be mapped to index fields. Before you can map a field, you must first create an index field with a matching type using the console or the `UpdateIndex` operation.

If you specify the `BlogFieldMappings` parameter, you must specify at least one field mapping.

Type: Array of [ConfluenceBlogToIndexFieldMapping \(p. 444\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 9 items.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ConfluenceBlogToIndexFieldMapping

Defines the mapping between a blog field in the Confluence data source to a Amazon Kendra index field.

You must first create the index field using the `UpdateIndex` operation.

Contents

DataSourceFieldName

The name of the field in the data source.

Type: String

Valid Values: AUTHOR | DISPLAY_URL | ITEM_TYPE | LABELS | PUBLISH_DATE | SPACE_KEY | SPACE_NAME | URL | VERSION

Required: No

DateFieldFormat

The format for date fields in the data source. If the field specified in `DataSourceFieldName` is a date field you must specify the date format. If the field is not a date field, an exception is thrown.

Type: String

Length Constraints: Minimum length of 4. Maximum length of 40.

Pattern: `^(?!s).*(?<!s)$`

Required: No

IndexFieldName

The name of the index field to map to the Confluence data source field. The index field type must match the Confluence field type.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 30.

Pattern: `^P{C}*$`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ConfluenceConfiguration

Provides configuration information for data sources that connect to Confluence.

Contents

AttachmentConfiguration

Specifies configuration information for indexing attachments to Confluence blogs and pages.

Type: [ConfluenceAttachmentConfiguration \(p. 441\)](#) object

Required: No

BlogConfiguration

Specifies configuration information for indexing Confluence blogs.

Type: [ConfluenceBlogConfiguration \(p. 443\)](#) object

Required: No

ExclusionPatterns

A list of regular expression patterns that apply to a URL on the Confluence server. An exclusion pattern can apply to a blog post, a page, a space, or an attachment. Items that match the pattern are excluded from the index. Items that don't match the pattern are included in the index. If an item matches both an exclusion pattern and an inclusion pattern, the item isn't included in the index.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 150.

Required: No

InclusionPatterns

A list of regular expression patterns that apply to a URL on the Confluence server. An inclusion pattern can apply to a blog post, a page, a space, or an attachment. Items that match the patterns are included in the index. Items that don't match the pattern are excluded from the index. If an item matches both an inclusion pattern and an exclusion pattern, the item isn't included in the index.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 150.

Required: No

PageConfiguration

Specifies configuration information for indexing Confluence pages.

Type: [ConfluencePageConfiguration \(p. 448\)](#) object

Required: No

SecretArn

The Amazon Resource Name (ARN) of an AWS Secrets Manager secret that contains the key/value pairs required to connect to your Confluence server. The secret must contain a JSON structure with the following keys:

- **username** - The user name or email address of a user with administrative privileges for the Confluence server.
- **password** - The password associated with the user logging in to the Confluence server.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: arn:[a-z0-9-\.]{1,63}:[a-z0-9-\.]{0,63}:[a-z0-9-\.]{0,63}:[a-z0-9-\.]{0,63}:[^/].{0,1023}

Required: Yes

ServerUrl

The URL of your Confluence instance. Use the full URL of the server. For example, <https://server.example.com:port/>. You can also use an IP address, for example, <https://192.168.1.113/>.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: ^(https?|ftp|file):\/\/([^\s]*)

Required: Yes

SpaceConfiguration

Specifies configuration information for indexing Confluence spaces.

Type: [ConfluenceSpaceConfiguration \(p. 450\)](#) object

Required: No

Version

Specifies the version of the Confluence installation that you are connecting to.

Type: String

Valid Values: CLOUD | SERVER

Required: Yes

VpcConfiguration

Specifies the information for connecting to an Amazon VPC.

Type: [DataSourceVpcConfiguration \(p. 468\)](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ConfluencePageConfiguration

Specifies the page settings for the Confluence data source.

Contents

PageFieldMappings

Defines how page metadata fields should be mapped to index fields. Before you can map a field, you must first create an index field with a matching type using the console or the `UpdateIndex` operation.

If you specify the `PageFieldMappings` parameter, you must specify at least one field mapping.

Type: Array of [ConfluencePageToIndexFieldMapping \(p. 449\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 12 items.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ConfluencePageToIndexFieldMapping

Defines the mapping between a field in the Confluence data source to a Amazon Kendra index field.

You must first create the index field using the `UpdateIndex` operation.

Contents

DataSourceFieldName

The name of the field in the data source.

Type: String

Valid Values: AUTHOR | CONTENT_STATUS | CREATED_DATE | DISPLAY_URL | ITEM_TYPE | LABELS | MODIFIED_DATE | PARENT_ID | SPACE_KEY | SPACE_NAME | URL | VERSION

Required: No

DateFieldFormat

The format for date fields in the data source. If the field specified in `DataSourceFieldName` is a date field you must specify the date format. If the field is not a date field, an exception is thrown.

Type: String

Length Constraints: Minimum length of 4. Maximum length of 40.

Pattern: `^(?!\\s).*(?<!\\s)$`

Required: No

IndexFieldName

The name of the index field to map to the Confluence data source field. The index field type must match the Confluence field type.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 30.

Pattern: `^\\P{C}*$`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ConfluenceSpaceConfiguration

Specifies the configuration for indexing Confluence spaces.

Contents

CrawlArchivedSpaces

Specifies whether Amazon Kendra should index archived spaces.

Type: Boolean

Required: No

CrawlPersonalSpaces

Specifies whether Amazon Kendra should index personal spaces. Users can add restrictions to items in personal spaces. If personal spaces are indexed, queries without user context information may return restricted items from a personal space in their results. For more information, see [Filtering on user context](#).

Type: Boolean

Required: No

ExcludeSpaces

A list of space keys of Confluence spaces. If you include a key, the blogs, documents, and attachments in the space are not indexed. If a space is in both the `ExcludeSpaces` and the `IncludeSpaces` list, the space is excluded.

Type: Array of strings

Array Members: Minimum number of 1 item.

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^\\P{C}*$`

Required: No

IncludeSpaces

A list of space keys for Confluence spaces. If you include a key, the blogs, documents, and attachments in the space are indexed. Spaces that aren't in the list aren't indexed. A space in the list must exist. Otherwise, Amazon Kendra logs an error when the data source is synchronized. If a space is in both the `IncludeSpaces` and the `ExcludeSpaces` list, the space is excluded.

Type: Array of strings

Array Members: Minimum number of 1 item.

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^\\P{C}*$`

Required: No

SpaceFieldMappings

Defines how space metadata fields should be mapped to index fields. Before you can map a field, you must first create an index field with a matching type using the console or the `UpdateIndex` operation.

If you specify the `SpaceFieldMappings` parameter, you must specify at least one field mapping.

Type: Array of [ConfluenceSpaceToIndexFieldMapping \(p. 452\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 4 items.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ConfluenceSpaceToIndexFieldMapping

Defines the mapping between a field in the Confluence data source to a Amazon Kendra index field.

You must first create the index field using the `UpdateIndex` operation.

Contents

DataSourceFieldName

The name of the field in the data source.

Type: String

Valid Values: `DISPLAY_URL` | `ITEM_TYPE` | `SPACE_KEY` | `URL`

Required: No

DateFieldFormat

The format for date fields in the data source. If the field specified in `DataSourceFieldName` is a date field you must specify the date format. If the field is not a date field, an exception is thrown.

Type: String

Length Constraints: Minimum length of 4. Maximum length of 40.

Pattern: `^(?!s).*(?<!s)$`

Required: No

IndexFieldName

The name of the index field to map to the Confluence data source field. The index field type must match the Confluence field type.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 30.

Pattern: `^P{C}*$`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ConnectionConfiguration

Provides the information necessary to connect to a database.

Contents

DatabaseHost

The name of the host for the database. Can be either a string (host.subdomain.domain.tld) or an IPv4 or IPv6 address.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 253.

Required: Yes

DatabaseName

The name of the database containing the document data.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z][a-zA-Z0-9_]*\$

Required: Yes

DatabasePort

The port that the database uses for connections.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 65535.

Required: Yes

SecretArn

The Amazon Resource Name (ARN) of credentials stored in AWS Secrets Manager. The credentials should be a user/password pair. For more information, see [Using a Database Data Source](#). For more information about AWS Secrets Manager, see [What Is AWS Secrets Manager](#) in the [AWS Secrets Manager](#) user guide.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: arn:[a-zA-Z0-9-\.]{1,63}:[a-zA-Z0-9-\.]{0,63}:[a-zA-Z0-9-\.]{0,63}:[a-zA-Z0-9-\.]{0,63}:[^/].{0,1023}

Required: Yes

TableName

The name of the table that contains the document data.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z][a-zA-Z0-9_]*\$

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

DatabaseConfiguration

Provides the information necessary to connect a database to an index.

Contents

AclConfiguration

Information about the database column that provides information for user context filtering.

Type: [AclConfiguration \(p. 427\)](#) object

Required: No

ColumnConfiguration

Information about where the index should get the document information from the database.

Type: [ColumnConfiguration \(p. 439\)](#) object

Required: Yes

ConnectionConfiguration

The information necessary to connect to a database.

Type: [ConnectionConfiguration \(p. 453\)](#) object

Required: Yes

DatabaseEngineType

The type of database engine that runs the database.

Type: String

Valid Values: RDS_AURORA_MYSQL | RDS_AURORA_POSTGRESQL | RDS MYSQL | RDS_POSTGRESQL

Required: Yes

SqlConfiguration

Provides information about how Amazon Kendra uses quote marks around SQL identifiers when querying a database data source.

Type: [SqlConfiguration \(p. 542\)](#) object

Required: No

VpcConfiguration

Provides information for connecting to an Amazon VPC.

Type: [DataSourceVpcConfiguration \(p. 468\)](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)

- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

DataSourceConfiguration

Configuration information for an Amazon Kendra data source.

Contents

ConfluenceConfiguration

Provides configuration information for connecting to a Confluence data source.

Type: [ConfluenceConfiguration \(p. 445\)](#) object

Required: No

DatabaseConfiguration

Provides information necessary to create a data source connector for a database.

Type: [DatabaseConfiguration \(p. 455\)](#) object

Required: No

GoogleDriveConfiguration

Provides configuration for data sources that connect to Google Drive.

Type: [GoogleDriveConfiguration \(p. 483\)](#) object

Required: No

OneDriveConfiguration

Provides configuration for data sources that connect to Microsoft OneDrive.

Type: [OneDriveConfiguration \(p. 499\)](#) object

Required: No

S3Configuration

Provides information to create a data source connector for a document repository in an Amazon S3 bucket.

Type: [S3DataSourceConfiguration \(p. 511\)](#) object

Required: No

SalesforceConfiguration

Provides configuration information for data sources that connect to a Salesforce site.

Type: [SalesforceConfiguration \(p. 516\)](#) object

Required: No

ServiceNowConfiguration

Provides configuration for data sources that connect to ServiceNow instances.

Type: [ServiceNowConfiguration \(p. 530\)](#) object

Required: No

SharePointConfiguration

Provides information necessary to create a data source connector for a Microsoft SharePoint site.

Type: [SharePointConfiguration \(p. 536\)](#) object

Required: No

WebCrawlerConfiguration

Provides the configuration information required for Amazon Kendra Web Crawler.

Type: [WebCrawlerConfiguration \(p. 559\)](#) object

Required: No

WorkDocsConfiguration

Provides the configuration information to connect to WorkDocs as your data source.

Type: [WorkDocsConfiguration \(p. 562\)](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

DataSourceGroup

Data source information for user context filtering.

Contents

DataSourceId

The identifier of the data source group you want to add to your list of data source groups. This is for filtering search results based on the groups' access to documents in that data source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: Yes

GroupId

The identifier of the group you want to add to your list of groups. This is for filtering search results based on the groups' access to documents.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: ^\P{C}*\$

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

DataSourceSummary

Summary information for a Amazon Kendra data source. Returned in a call to the `DescribeDataSource` operation.

Contents

CreatedAt

The UNIX datetime that the data source was created.

Type: `Timestamp`

Required: No

Id

The unique identifier for the data source.

Type: `String`

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: `[a-zA-Z0-9][a-zA-Z0-9_-]*`

Required: No

LanguageCode

The code for a language. This shows a supported language for all documents in the data source. English is supported by default. For more information on supported languages, including their codes, see [Adding documents in languages other than English](#).

Type: `String`

Length Constraints: Minimum length of 2. Maximum length of 10.

Pattern: `[a-zA-Z-]*`

Required: No

Name

The name of the data source.

Type: `String`

Length Constraints: Minimum length of 1. Maximum length of 1000.

Pattern: `[a-zA-Z0-9][a-zA-Z0-9_-]*`

Required: No

Status

The status of the data source. When the status is `ACTIVE` the data source is ready to use.

Type: `String`

Valid Values: `CREATING` | `DELETING` | `FAILED` | `UPDATING` | `ACTIVE`

Required: No

Type

The type of the data source.

Type: String

Valid Values: S3 | SHAREPOINT | DATABASE | SALESFORCE | ONEDRIVE | SERVICENOW | CUSTOM | CONFLUENCE | GOOGLEDRAVE | WEBCRAWLER | WORKDOCS

Required: No

UpdatedAt

The UNIX datetime that the data source was last updated.

Type: Timestamp

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

DataSourceSyncJob

Provides information about a synchronization job.

Contents

DataSourceErrorCode

If the reason that the synchronization failed is due to an error with the underlying data source, this field contains a code that identifies the error.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: No

EndTime

The UNIX datetime that the synchronization job was completed.

Type: Timestamp

Required: No

ErrorCode

If the Status field is set to FAILED, the ErrorCode field contains a the reason that the synchronization failed.

Type: String

Valid Values: InternalError | InvalidRequest

Required: No

ErrorMessage

If the Status field is set to ERROR, the ErrorMessage field contains a description of the error that caused the synchronization to fail.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: ^\P{C}*\$

Required: No

ExecutionId

A unique identifier for the synchronization job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: No

Metrics

Maps a batch delete document request to a specific data source sync job. This is optional and should only be supplied when documents are deleted by a data source connector.

Type: [DataSourceSyncJobMetrics \(p. 464\)](#) object

Required: No

StartTime

The UNIX datetime that the synchronization job was started.

Type: Timestamp

Required: No

Status

The execution status of the synchronization job. When the `Status` field is set to `SUCCEEDED`, the synchronization job is done. If the status code is set to `FAILED`, the `ErrorCode` and `ErrorMessage` fields give you the reason for the failure.

Type: String

Valid Values: `FAILED` | `SUCCEEDED` | `SYNCING` | `INCOMPLETE` | `STOPPING` | `ABORTED` | `SYNCING_INDEXING`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

DataSourceSyncJobMetrics

Maps a batch delete document request to a specific data source sync job. This is optional and should only be supplied when documents are deleted by a data source connector.

Contents

DocumentsAdded

The number of documents added from the data source up to now in the data source sync.

Type: String

Pattern: (([1-9][0-9]*)|0)

Required: No

DocumentsDeleted

The number of documents deleted from the data source up to now in the data source sync run.

Type: String

Pattern: (([1-9][0-9]*)|0)

Required: No

DocumentsFailed

The number of documents that failed to sync from the data source up to now in the data source sync run.

Type: String

Pattern: (([1-9][0-9]*)|0)

Required: No

DocumentsModified

The number of documents modified in the data source up to now in the data source sync run.

Type: String

Pattern: (([1-9][0-9]*)|0)

Required: No

DocumentsScanned

The current number of documents crawled by the current sync job in the data source.

Type: String

Pattern: (([1-9][0-9]*)|0)

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

DataSourceSyncJobMetricTarget

Maps a particular data source sync job to a particular data source.

Contents

DataSourceId

The ID of the data source that is running the sync job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: Yes

DataSourceSyncJobId

The ID of the sync job that is running on the data source.

If the ID of a sync job is not provided and there is a sync job running, then the ID of this sync job is used and metrics are generated for this sync job.

If the ID of a sync job is not provided and there is no sync job running, then no metrics are generated and documents are indexed/deleted at the index level without sync job metrics included.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

DataSourceToIndexFieldMapping

Maps a column or attribute in the data source to an index field. You must first create the fields in the index using the `UpdateIndex` operation.

Contents

DataSourceFieldName

The name of the column or attribute in the data source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: `^[a-zA-Z][a-zA-Z0-9_.]*$`

Required: Yes

DateFieldFormat

The type of data stored in the column or attribute.

Type: String

Length Constraints: Minimum length of 4. Maximum length of 40.

Pattern: `^(?!s).*(?<!s)$`

Required: No

IndexFieldName

The name of the field in the index.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 30.

Pattern: `^P{C}*$`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

DataSourceVpcConfiguration

Provides information for connecting to an Amazon VPC.

Contents

SecurityGroupIds

A list of identifiers of security groups within your Amazon VPC. The security groups should enable Amazon Kendra to connect to the data source.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 10 items.

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [-0-9a-zA-Z]+

Required: Yes

SubnetIds

A list of identifiers for subnets within your Amazon VPC. The subnets should be able to connect to each other in the VPC, and they should have outgoing access to the Internet through a NAT device.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 6 items.

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [\ -0-9a-zA-Z]+

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Document

A document in an index.

Contents

AccessControlList

Information on user and group access rights, which is used for user context filtering.

Type: Array of [Principal \(p. 502\)](#) objects

Required: No

Attributes

Custom attributes to apply to the document. Use the custom attributes to provide additional information for searching, to provide facets for refining searches, and to provide additional information in the query response.

Type: Array of [DocumentAttribute \(p. 471\)](#) objects

Required: No

Blob

The contents of the document.

Documents passed to the `Blob` parameter must be base64 encoded. Your code might not need to encode the document file bytes if you're using an AWS SDK to call Amazon Kendra operations. If you are calling the Amazon Kendra endpoint directly using REST, you must base64 encode the contents before sending.

Type: Base64-encoded binary data object

Required: No

ContentType

The file type of the document in the `Blob` field.

Type: String

Valid Values: `PDF` | `HTML` | `MS_WORD` | `PLAIN_TEXT` | `PPT`

Required: No

HierarchicalAccessControlList

The list of [principal](#) lists that define the hierarchy for which documents users should have access to.

Type: Array of [HierarchicalPrincipal \(p. 489\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 30 items.

Required: No

Id

A unique identifier of the document in the index.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

S3Path

Information required to find a specific file in an Amazon S3 bucket.

Type: [S3Path \(p. 513\)](#) object

Required: No

Title

The title of the document.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

DocumentAttribute

A custom attribute value assigned to a document.

Contents

Key

The identifier for the attribute.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9_][a-zA-Z0-9_-]*

Required: Yes

Value

The value of the attribute.

Type: [DocumentAttributeValue \(p. 472\)](#) object

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

DocumentAttributeValue

The value of a custom document attribute. You can only provide one value for a custom attribute.

Contents

DateValue

A date expressed as an ISO 8601 string.

It is important for the time zone to be included in the ISO 8601 date-time format. For example, 20120325T123010+01:00 is the ISO 8601 date-time format for March 25th 2012 at 12:30PM (plus 10 seconds) in Central European Time.

Type: Timestamp

Required: No

LongValue

A long integer value.

Type: Long

Required: No

StringListValue

A list of strings.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: No

StringValue

A string, such as "department".

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

DocumentAttributeValueCountPair

Provides the count of documents that match a particular attribute when doing a faceted search.

Contents

Count

The number of documents in the response that have the attribute value for the key.

Type: Integer

Required: No

DocumentAttributeValue

The value of the attribute. For example, "HR."

Type: [DocumentAttributeValue \(p. 472\)](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

DocumentInfo

Identifies a document for which to retrieve status information

Contents

Attributes

Attributes that identify a specific version of a document to check.

The only valid attributes are:

- `version`
- `datasourceId`
- `jobExecutionId`

The attributes follow these rules:

- `dataSourceId` and `jobExecutionId` must be used together.
- `version` is ignored if `dataSourceId` and `jobExecutionId` are not provided.
- If `dataSourceId` and `jobExecutionId` are provided, but `version` is not, the version defaults to "0".

Type: Array of [DocumentAttribute \(p. 471\)](#) objects

Required: No

DocumentId

The unique identifier of the document.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

DocumentMetadataConfiguration

Specifies the properties of a custom index field.

Contents

Name

The name of the index field.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 30.

Required: Yes

Relevance

Provides manual tuning parameters to determine how the field affects the search results.

Type: [Relevance \(p. 508\)](#) object

Required: No

Search

Provides information about how the field is used during a search.

Type: [Search \(p. 527\)](#) object

Required: No

Type

The data type of the index field.

Type: String

Valid Values: STRING_VALUE | STRING_LIST_VALUE | LONG_VALUE | DATE_VALUE

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

DocumentRelevanceConfiguration

Overrides the document relevance properties of a custom index field.

Contents

Name

The name of the tuning configuration to override document relevance at the index level.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 30.

Required: Yes

Relevance

Provides information for manually tuning the relevance of a field in a search. When a query includes terms that match the field, the results are given a boost in the response based on these tuning parameters.

Type: [Relevance \(p. 508\)](#) object

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

DocumentsMetadataConfiguration

Document metadata files that contain information such as the document access control information, source URI, document author, and custom attributes. Each metadata file contains metadata about a single document.

Contents

S3Prefix

A prefix used to filter metadata configuration files in the AWS S3 bucket. The S3 bucket might contain multiple metadata files. Use `S3Prefix` to include only the desired metadata files.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Facet

Information about a document attribute

Contents

DocumentAttributeKey

The unique key for the document attribute.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9_][a-zA-Z0-9_-]*

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

FacetResult

The facet values for the documents in the response.

Contents

DocumentAttributeKey

The key for the facet values. This is the same as the [DocumentAttributeKey](#) provided in the query.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: [a-zA-Z0-9_][a-zA-Z0-9_-]*

Required: No

DocumentAttributeValueCountPairs

An array of key/value pairs, where the key is the value of the attribute and the count is the number of documents that share the key value.

Type: Array of [DocumentAttributeValueCountPair](#) (p. 473) objects

Required: No

DocumentAttributeValueType

The data type of the facet value. This is the same as the type defined for the index field when it was created.

Type: String

Valid Values: STRING_VALUE | STRING_LIST_VALUE | LONG_VALUE | DATE_VALUE

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

FaqStatistics

Provides statistical information about the FAQ questions and answers contained in an index.

Contents

IndexedQuestionAnswersCount

The total number of FAQ questions and answers contained in the index.

Type: Integer

Valid Range: Minimum value of 0.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

FaqSummary

Provides information about a frequently asked questions and answer contained in an index.

Contents

CreatedAt

The UNIX datetime that the FAQ was added to the index.

Type: Timestamp

Required: No

FileFormat

The file type used to create the FAQ.

Type: String

Valid Values: CSV | CSV_WITH_HEADER | JSON

Required: No

Id

The unique identifier of the FAQ.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: No

LanguageCode

The code for a language. This shows a supported language for the FAQ document as part of the summary information for FAQs. English is supported by default. For more information on supported languages, including their codes, see [Adding documents in languages other than English](#).

Type: String

Length Constraints: Minimum length of 2. Maximum length of 10.

Pattern: [a-zA-Z-]*

Required: No

Name

The name that you assigned the FAQ when you created or updated the FAQ.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: No

Status

The current status of the FAQ. When the status is ACTIVE the FAQ is ready for use.

Type: String

Valid Values: CREATING | UPDATING | ACTIVE | DELETING | FAILED

Required: No

UpdatedAt

The UNIX datetime that the FAQ was last updated.

Type: Timestamp

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

GoogleDriveConfiguration

Provides configuration information for data sources that connect to Google Drive.

Contents

ExcludeMimeTypes

A list of MIME types to exclude from the index. All documents matching the specified MIME type are excluded.

For a list of MIME types, see [Using a Google Workspace Drive data source](#).

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 30 items.

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^\\P{C}*$`

Required: No

ExcludeSharedDrives

A list of identifiers or shared drives to exclude from the index. All files and folders stored on the shared drive are excluded.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^\\P{C}*$`

Required: No

ExcludeUserAccounts

A list of email addresses of the users. Documents owned by these users are excluded from the index. Documents shared with excluded users are indexed unless they are excluded in another way.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^\\P{C}*$`

Required: No

ExclusionPatterns

A list of regular expression patterns that apply to the path on Google Drive. Items that match the pattern are excluded from the index from both shared drives and users' My Drives. Items that don't match the pattern are included in the index. If an item matches both an exclusion pattern and an inclusion pattern, it is excluded from the index.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 150.

Required: No

FieldMappings

Defines mapping between a field in the Google Drive and a Amazon Kendra index field.

If you are using the console, you can define index fields when creating the mapping. If you are using the API, you must first create the field using the `UpdateIndex` operation.

Type: Array of [DataSourceToIndexFieldMapping \(p. 467\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Required: No

InclusionPatterns

A list of regular expression patterns that apply to path on Google Drive. Items that match the pattern are included in the index from both shared drives and users' My Drives. Items that don't match the pattern are excluded from the index. If an item matches both an inclusion pattern and an exclusion pattern, it is excluded from the index.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 150.

Required: No

SecretArn

The Amazon Resource Name (ARN) of a AWS Secrets Managersecret that contains the credentials required to connect to Google Drive. For more information, see [Using a Google Workspace Drive data source](#).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: `arn:[a-z0-9-\.]{1,63}:[a-z0-9-\.]{0,63}:[a-z0-9-\.]{0,63}:[a-z0-9-\.]{0,63}:[^/].{0,1023}`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

GroupMembers

A list of users or sub groups that belong to a group. Users and groups are useful for filtering search results to different users based on their group's access to documents.

Contents

MemberGroups

A list of sub groups that belong to a group. For example, the sub groups "Research", "Engineering", and "Sales and Marketing" all belong to the group "Company".

Type: Array of [MemberGroup \(p. 497\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 1000 items.

Required: No

MemberUsers

A list of users that belong to a group. For example, a list of interns all belong to the "Interns" group.

Type: Array of [MemberUser \(p. 498\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 1000 items.

Required: No

S3PathforGroupMembers

If you have more than 1000 users and/or sub groups for a single group, you need to provide the path to the S3 file that lists your users and sub groups for a group. Your sub groups can contain more than 1000 users, but the list of sub groups that belong to a group (and/or users) must be no more than 1000.

You can download this [example S3 file](#) that uses the correct format for listing group members. Note, `dataSourceId` is optional. The value of `type` for a group is always `GROUP` and for a user it is always `USER`.

Type: [S3Path \(p. 513\)](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

GroupOrderingIdSummary

Information on the processing of `PUT` and `DELETE` actions for mapping users to their groups.

Contents

FailureReason

The reason an action could not be processed. An action can be a `PUT` or `DELETE` action for mapping users to their groups.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: ^\P{C}*\$

Required: No

LastUpdatedAt

The last date-time an action was updated. An action can be a `PUT` or `DELETE` action for mapping users to their groups.

Type: Timestamp

Required: No

OrderingId

The order in which actions should complete processing. An action can be a `PUT` or `DELETE` action for mapping users to their groups.

Type: Long

Valid Range: Minimum value of 0. Maximum value of 32535158400000.

Required: No

ReceivedAt

The date-time an action was received by Amazon Kendra. An action can be a `PUT` or `DELETE` action for mapping users to their groups.

Type: Timestamp

Required: No

Status

The current processing status of actions for mapping users to their groups. The status can be either `PROCESSING`, `SUCCEEDED`, `DELETING`, `DELETED`, or `FAILED`.

Type: String

Valid Values: `FAILED` | `SUCCEEDED` | `PROCESSING` | `DELETING` | `DELETED`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

GroupSummary

Group summary information.

Contents

GroupId

The identifier of the group you want group summary information on.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: ^\P{C}*\$

Required: No

OrderingId

The timestamp identifier used for the latest `PUT` or `DELETE` action.

Type: Long

Valid Range: Minimum value of 0. Maximum value of 32535158400000.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

HierarchicalPrincipal

Information to define the hierarchy for which documents users should have access to.

Contents

PrincipalList

A list of [principal](#) lists that define the hierarchy for which documents users should have access to. Each hierarchical list specifies which user or group has allow or deny access for each document.

Type: Array of [Principal \(p. 502\)](#) objects

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Highlight

Provides information that you can use to highlight a search result so that your users can quickly identify terms in the response.

Contents

BeginOffset

The zero-based location in the response string where the highlight starts.

Type: Integer

Required: Yes

EndOffset

The zero-based location in the response string where the highlight ends.

Type: Integer

Required: Yes

TopAnswer

Indicates whether the response is the best response. True if this is the best response; otherwise, false.

Type: Boolean

Required: No

Type

The highlight type.

Type: String

Valid Values: STANDARD | THESAURUS_SYNONYM

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

IndexConfigurationSummary

A summary of information about an index.

Contents

CreatedAt

The Unix timestamp when the index was created.

Type: Timestamp

Required: Yes

Edition

Indicates whether the index is a enterprise edition index or a developer edition index.

Type: String

Valid Values: DEVELOPER_EDITION | ENTERPRISE_EDITION

Required: No

Id

A unique identifier for the index. Use this to identify the index when you are using operations such as `Query`, `DescribeIndex`, `UpdateIndex`, and `DeleteIndex`.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: No

Name

The name of the index.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1000.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: No

Status

The current status of the index. When the status is ACTIVE, the index is ready to search.

Type: String

Valid Values: CREATING | ACTIVE | DELETING | FAILED | UPDATING | SYSTEM_UPDATING

Required: Yes

UpdatedAt

The Unix timestamp when the index was last updated by the `UpdateIndex` operation.

Type: Timestamp

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

IndexStatistics

Provides information about the number of documents and the number of questions and answers in an index.

Contents

FaqStatistics

The number of question and answer topics in the index.

Type: [FaqStatistics \(p. 480\)](#) object

Required: Yes

TextDocumentStatistics

The number of text documents indexed.

Type: [TextDocumentStatistics \(p. 549\)](#) object

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

JsonTokenTypeConfiguration

Configuration information for the JSON token type.

Contents

GroupAttributeField

The group attribute field.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

UserNameAttributeField

The user name attribute field.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

JwtTokenTypeConfiguration

Configuration information for the JWT token type.

Contents

ClaimRegex

The regular expression that identifies the claim.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^\P{C}*\$

Required: No

GroupAttributeField

The group attribute field.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^\P{C}*\$

Required: No

Issuer

The issuer of the token.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 65.

Pattern: ^\P{C}*\$

Required: No

KeyLocation

The location of the key.

Type: String

Valid Values: URL | SECRET_MANAGER

Required: Yes

SecretManagerArn

The Amazon Resource Name (arn) of the secret.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: arn:[a-zA-Z0-9-.]{1,63}:[a-zA-Z0-9-.]{0,63}:[a-zA-Z0-9-.]{0,63}:[a-zA-Z0-9-.]{0,63}:[^/.]{0,1023}

Required: No

URL

The signing key URL.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^(https?|ftp|file):\/\/([^\s]*)`

Required: No

UserNameAttributeField

The user name attribute field.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: `^P{C}*$`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

MemberGroup

The sub groups that belong to a group.

Contents

DataSourceId

The identifier of the data source for the sub group you want to map to a group.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: No

GroupId

The identifier of the sub group you want to map to a group.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: ^\P{C}*\$

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

MemberUser

The users that belong to a group.

Contents

UserId

The identifier of the user you want to map to a group.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: ^\P{C}*\$

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

OneDriveConfiguration

Provides configuration information for data sources that connect to OneDrive.

Contents

DisableLocalGroups

A Boolean value that specifies whether local groups are disabled (`True`) or enabled (`False`).

Type: Boolean

Required: No

ExclusionPatterns

List of regular expressions applied to documents. Items that match the exclusion pattern are not indexed. If you provide both an inclusion pattern and an exclusion pattern, any item that matches the exclusion pattern isn't indexed.

The exclusion pattern is applied to the file name.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 150.

Required: No

FieldMappings

A list of `DataSourceToIndexFieldMapping` objects that map Microsoft OneDrive fields to custom fields in the Amazon Kendra index. You must first create the index fields before you map OneDrive fields.

Type: Array of [DataSourceToIndexFieldMapping \(p. 467\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Required: No

InclusionPatterns

A list of regular expression patterns. Documents that match the pattern are included in the index. Documents that don't match the pattern are excluded from the index. If a document matches both an inclusion pattern and an exclusion pattern, the document is not included in the index.

The exclusion pattern is applied to the file name.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 150.

Required: No

OneDriveUsers

A list of user accounts whose documents should be indexed.

Type: [OneDriveUsers \(p. 501\)](#) object

Required: Yes

SecretArn

The Amazon Resource Name (ARN) of an AWS Secrets Manager secret that contains the user name and password to connect to OneDrive. The user name should be the application ID for the OneDrive application, and the password is the application key for the OneDrive application.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: `arn:[a-zA-Z0-9-\.]{1,63}:[a-zA-Z0-9-\.]{0,63}:[a-zA-Z0-9-\.]{0,63}:[a-zA-Z0-9-\.]{0,63}:[^/].{0,1023}`

Required: Yes

TenantDomain

The Azure Active Directory domain of the organization.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^([a-zA-Z0-9]+(-[a-zA-Z0-9]+)*\.)+[a-zA-Z]{2,}$`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

OneDriveUsers

User accounts whose documents should be indexed.

Contents

OneDriveUserList

A list of users whose documents should be indexed. Specify the user names in email format, for example, `username@tenantdomain`. If you need to index the documents of more than 100 users, use the `OneDriveUserS3Path` field to specify the location of a file containing a list of users.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^(!\s)+@[a-zA-Z0-9_\\-\\.]+\\.[a-zA-Z]{2,5}$`

Required: No

OneDriveUserS3Path

The S3 bucket location of a file containing a list of users whose documents should be indexed.

Type: [S3Path \(p. 513\)](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Principal

Provides user and group information for document access filtering.

Contents

Access

Whether to allow or deny access to the principal.

Type: String

Valid Values: ALLOW | DENY

Required: Yes

DataSourceId

The identifier of the data source the principal should access documents from.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: No

Name

The name of the user or group.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: ^\P{C}*\$

Required: Yes

Type

The type of principal.

Type: String

Valid Values: USER | GROUP

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ProxyConfiguration

Provides the configuration information for a web proxy to connect to website hosts.

Contents

Credentials

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.
Pattern: `arn:[a-z0-9-\.]{1,63}:[a-z0-9-\.]{0,63}:[a-z0-9-\.]{0,63}:[a-z0-9-\.]{0,63}:[^/].{0,1023}`

Required: No

Host

The name of the website host you want to connect to via a web proxy server.

For example, the host name of `https://a.example.com/page1.html` is "a.example.com".

Type: String

Length Constraints: Minimum length of 1. Maximum length of 253.

Pattern: `([^\s]*)`

Required: Yes

Port

The port number of the website host you want to connect to via a web proxy server.

For example, the port for `https://a.example.com/page1.html` is 443, the standard port for HTTPS.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 65535.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

QueryResultItem

A single query result.

A query result contains information about a document returned by the query. This includes the original location of the document, a list of attributes assigned to the document, and relevant text from the document that satisfies the query.

Contents

AdditionalAttributes

One or more additional attributes associated with the query result.

Type: Array of [AdditionalResultAttribute \(p. 428\)](#) objects

Required: No

DocumentAttributes

An array of document attributes for the document that the query result maps to. For example, the document author (Author) or the source URI (SourceUri) of the document.

Type: Array of [DocumentAttribute \(p. 471\)](#) objects

Required: No

DocumentExcerpt

An extract of the text in the document. Contains information about highlighting the relevant terms in the excerpt.

Type: [TextWithHighlights \(p. 550\)](#) object

Required: No

DocumentId

The unique identifier for the document.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: No

DocumentTitle

The title of the document. Contains the text of the title and information for highlighting the relevant terms in the title.

Type: [TextWithHighlights \(p. 550\)](#) object

Required: No

DocumentURI

The URI of the original location of the document.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^(https?|ftp|file):\/\/([^\s]*)`

Required: No

FeedbackToken

A token that identifies a particular result from a particular query. Use this token to provide click-through feedback for the result. For more information, see [Submitting feedback](#).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: ^\P{C}*.\P{C}*\$

Required: No

Id

The unique identifier for the query result.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 73.

Required: No

ScoreAttributes

Indicates the confidence that Amazon Kendra has that a result matches the query that you provided. Each result is placed into a bin that indicates the confidence, `VERY_HIGH`, `HIGH`, `MEDIUM` and `LOW`. You can use the score to determine if a response meets the confidence needed for your application.

The field is only set to `LOW` when the `Type` field is set to `DOCUMENT` and Amazon Kendra is not confident that the result matches the query.

Type: [ScoreAttributes \(p. 526\)](#) object

Required: No

Type

The type of document.

Type: String

Valid Values: `DOCUMENT` | `QUESTION_ANSWER` | `ANSWER`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

QuerySuggestionsBlockListSummary

Summary information on a query suggestions block list.

This includes information on the block list ID, block list name, when the block list was created, when the block list was last updated, and the count of block words/phrases in the block list.

For information on the current quota limits for block lists, see [Quotas for Amazon Kendra](#).

Contents

CreatedAt

The date-time summary information for a query suggestions block list was last created.

Type: Timestamp

Required: No

Id

The identifier of a block list.

Type: String

Length Constraints: Fixed length of 36.

Pattern: [a-zA-Z0-9][a-zA-Z0-9-]*

Required: No

ItemCount

The number of items in the block list file.

Type: Integer

Required: No

Name

The name of the block list.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z0-9](-*[a-zA-Z0-9])*

Required: No

Status

The status of the block list.

Type: String

Valid Values: ACTIVE | CREATING | DELETING | UPDATING |
ACTIVE_BUT_UPDATE_FAILED | FAILED

Required: No

UpdatedAt

The date-time the block list was last updated.

Type: Timestamp

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Relevance

Provides information for manually tuning the relevance of a field in a search. When a query includes terms that match the field, the results are given a boost in the response based on these tuning parameters.

Contents

Duration

Specifies the time period that the boost applies to. For example, to make the boost apply to documents with the field value within the last month, you would use "2628000s". Once the field value is beyond the specified range, the effect of the boost drops off. The higher the importance, the faster the effect drops off. If you don't specify a value, the default is 3 months. The value of the field is a numeric string followed by the character "s", for example "86400s" for one day, or "604800s" for one week.

Only applies to **DATE** fields.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 10.

Pattern: [0-9]+[s]

Required: No

Freshness

Indicates that this field determines how "fresh" a document is. For example, if document 1 was created on November 5, and document 2 was created on October 31, document 1 is "fresher" than document 2. You can only set the **Freshness** field on one **DATE** type field. Only applies to **DATE** fields.

Type: Boolean

Required: No

Importance

The relative importance of the field in the search. Larger numbers provide more of a boost than smaller numbers.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 10.

Required: No

RankOrder

Determines how values should be interpreted.

When the **RankOrder** field is **ASCENDING**, higher numbers are better. For example, a document with a rating score of 10 is higher ranking than a document with a rating score of 1.

When the **RankOrder** field is **DESCENDING**, lower numbers are better. For example, in a task tracking application, a priority 1 task is more important than a priority 5 task.

Only applies to **LONG** and **DOUBLE** fields.

Type: String

Valid Values: ASCENDING | DESCENDING

Required: No

ValueImportanceMap

A list of values that should be given a different boost when they appear in the result list. For example, if you are boosting a field called "department," query terms that match the department field are boosted in the result. However, you can add entries from the department field to boost documents with those values higher.

For example, you can add entries to the map with names of departments. If you add "HR",5 and "Legal",3 those departments are given special attention when they appear in the metadata of a document. When those terms appear they are given the specified importance instead of the regular importance for the boost.

Type: String to integer map

Key Length Constraints: Minimum length of 1. Maximum length of 50.

Valid Range: Minimum value of 1. Maximum value of 10.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

RelevanceFeedback

Provides feedback on how relevant a document is to a search. Your application uses the `SubmitFeedback` operation to provide relevance information.

Contents

RelevanceValue

Whether the document was relevant or not relevant to the search.

Type: String

Valid Values: `RELEVANT` | `NOT_RELEVANT`

Required: Yes

ResultId

The unique identifier of the search result that the user provided relevance feedback for.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 73.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

S3DataSourceConfiguration

Provides configuration information for a data source to index documents in an Amazon S3 bucket.

Contents

AccessControlListConfiguration

Provides the path to the S3 bucket that contains the user context filtering files for the data source. For the format of the file, see [Access control for S3 data sources](#).

Type: [AccessControlListConfiguration \(p. 426\)](#) object

Required: No

BucketName

The name of the bucket that contains the documents.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 63.

Pattern: [a-z0-9][\.-a-z0-9]{1,61}[a-z0-9]

Required: Yes

DocumentsMetadataConfiguration

Document metadata files that contain information such as the document access control information, source URI, document author, and custom attributes. Each metadata file contains metadata about a single document.

Type: [DocumentsMetadataConfiguration \(p. 477\)](#) object

Required: No

ExclusionPatterns

A list of glob patterns for documents that should not be indexed. If a document that matches an inclusion prefix or inclusion pattern also matches an exclusion pattern, the document is not indexed.

Some [examples](#) are:

- `*.png, *.jpg` will exclude all PNG and JPEG image files in a directory (files with the extensions .png and .jpg).
- `*internal*` will exclude all files in a directory that contain 'internal' in the file name, such as 'internal', 'internal_only', 'company_internal'.
- `**/*internal*` will exclude all internal-related files in a directory and its subdirectories.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 150.

Required: No

InclusionPatterns

A list of glob patterns for documents that should be indexed. If a document that matches an inclusion pattern also matches an exclusion pattern, the document is not indexed.

Some [examples](#) are:

- `*.txt` will include all text files in a directory (files with the extension `.txt`).
- `**/*.txt` will include all text files in a directory and its subdirectories.
- `*tax*` will include all files in a directory that contain 'tax' in the file name, such as 'tax', 'taxes', 'income_tax'.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 150.

Required: No

InclusionPrefixes

A list of S3 prefixes for the documents that should be included in the index.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 150.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

S3Path

Information required to find a specific file in an Amazon S3 bucket.

Contents

Bucket

The name of the S3 bucket that contains the file.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 63.

Pattern: [a-zA-Z0-9][\.-a-zA-Z0-9]{1,61}[a-zA-Z0-9]

Required: Yes

Key

The name of the file.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

SalesforceChatterFeedConfiguration

Defines configuration for syncing a Salesforce chatter feed. The contents of the object comes from the Salesforce FeedItem table.

Contents

DocumentDataFieldName

The name of the column in the Salesforce FeedItem table that contains the content to index.
Typically this is the Body column.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z][a-zA-Z0-9_.]*\$

Required: Yes

DocumentTitleFieldName

The name of the column in the Salesforce FeedItem table that contains the title of the document.
This is typically the Title column.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z][a-zA-Z0-9_.]*\$

Required: No

FieldMappings

Maps fields from a Salesforce chatter feed into Amazon Kendra index fields.

Type: Array of [DataSourceToIndexFieldMapping \(p. 467\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Required: No

IncludeFilterTypes

Filters the documents in the feed based on status of the user. When you specify ACTIVE_USERS only documents from users who have an active account are indexed. When you specify STANDARD_USER only documents for Salesforce standard users are documented. You can specify both.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 2 items.

Valid Values: ACTIVE_USER | STANDARD_USER

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

SalesforceConfiguration

Provides configuration information for connecting to a Salesforce data source.

Contents

ChatterFeedConfiguration

Specifies configuration information for Salesforce chatter feeds.

Type: [SalesforceChatterFeedConfiguration \(p. 514\)](#) object

Required: No

CrawlAttachments

Indicates whether Amazon Kendra should index attachments to Salesforce objects.

Type: Boolean

Required: No

ExcludeAttachmentFilePatterns

A list of regular expression patterns. Documents that match the patterns are excluded from the index. Documents that don't match the patterns are included in the index. If a document matches both an exclusion pattern and an inclusion pattern, the document is not included in the index.

The regex is applied to the name of the attached file.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 150.

Required: No

IncludeAttachmentFilePatterns

A list of regular expression patterns. Documents that match the patterns are included in the index. Documents that don't match the patterns are excluded from the index. If a document matches both an inclusion pattern and an exclusion pattern, the document is not included in the index.

The regex is applied to the name of the attached file.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 150.

Required: No

KnowledgeArticleConfiguration

Specifies configuration information for the knowledge article types that Amazon Kendra indexes. Amazon Kendra indexes standard knowledge articles and the standard fields of knowledge articles, or the custom fields of custom knowledge articles, but not both.

Type: [SalesforceKnowledgeArticleConfiguration \(p. 521\)](#) object

Required: No

SecretArn

The Amazon Resource Name (ARN) of an AWS Secrets Managersecret that contains the key/value pairs required to connect to your Salesforce instance. The secret must contain a JSON structure with the following keys:

- authenticationUrl - The OAUTH endpoint that Amazon Kendra connects to get an OAUTH token.
- consumerKey - The application public key generated when you created your Salesforce application.
- consumerSecret - The application private key generated when you created your Salesforce application.
- password - The password associated with the user logging in to the Salesforce instance.
- securityToken - The token associated with the user account logging in to the Salesforce instance.
- username - The user name of the user logging in to the Salesforce instance.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: `arn:[a-z0-9-\.]{1,63}:[a-z0-9-\.]{0,63}:[a-z0-9-\.]{0,63}:[a-z0-9-\.]{0,63}:[^/].{0,1023}`

Required: Yes

ServerUrl

The instance URL for the Salesforce site that you want to index.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^(https?|ftp|file):\/\/([^\s]*)`

Required: Yes

StandardObjectAttachmentConfiguration

Provides configuration information for processing attachments to Salesforce standard objects.

Type: [SalesforceStandardObjectAttachmentConfiguration \(p. 523\)](#) object

Required: No

StandardObjectConfigurations

Specifies the Salesforce standard objects that Amazon Kendra indexes.

Type: Array of [SalesforceStandardObjectConfiguration \(p. 524\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 17 items.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)

- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

SalesforceCustomKnowledgeArticleTypeConfiguration

Provides configuration information for indexing Salesforce custom articles.

Contents

DocumentDataFieldName

The name of the field in the custom knowledge article that contains the document data to index.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z][a-zA-Z0-9_.]*\$

Required: Yes

DocumentTitleFieldName

The name of the field in the custom knowledge article that contains the document title.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z][a-zA-Z0-9_.]*\$

Required: No

FieldMappings

One or more objects that map fields in the custom knowledge article to fields in the Amazon Kendra index.

Type: Array of [DataSourceToIndexFieldMapping \(p. 467\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Required: No

Name

The name of the configuration.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z][a-zA-Z0-9_.]*\$

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)

- [AWS SDK for Ruby V3](#)

SalesforceKnowledgeArticleConfiguration

Specifies configuration information for the knowledge article types that Amazon Kendra indexes. Amazon Kendra indexes standard knowledge articles and the standard fields of knowledge articles, or the custom fields of custom knowledge articles, but not both

Contents

CustomKnowledgeArticleTypeConfigurations

Provides configuration information for custom Salesforce knowledge articles.

Type: Array of [SalesforceCustomKnowledgeArticleTypeConfiguration \(p. 519\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 10 items.

Required: No

IncludedStates

Specifies the document states that should be included when Amazon Kendra indexes knowledge articles. You must specify at least one state.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 3 items.

Valid Values: DRAFT | PUBLISHED | ARCHIVED

Required: Yes

StandardKnowledgeArticleTypeConfiguration

Provides configuration information for standard Salesforce knowledge articles.

Type: [SalesforceStandardKnowledgeArticleTypeConfiguration \(p. 522\)](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

SalesforceStandardKnowledgeArticleTypeConfiguration

Provides configuration information for standard Salesforce knowledge articles.

Contents

DocumentDataFieldName

The name of the field that contains the document data to index.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z][a-zA-Z0-9_.]*\$

Required: Yes

DocumentTitleFieldName

The name of the field that contains the document title.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z][a-zA-Z0-9_.]*\$

Required: No

FieldMappings

One or more objects that map fields in the knowledge article to Amazon Kendra index fields. The index field must exist before you can map a Salesforce field to it.

Type: Array of [DataSourceToIndexFieldMapping \(p. 467\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

SalesforceStandardObjectAttachmentConfiguration

Provides configuration information for processing attachments to Salesforce standard objects.

Contents

DocumentTitleFieldName

The name of the field used for the document title.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z][a-zA-Z0-9_.]*\$

Required: No

FieldMappings

One or more objects that map fields in attachments to Amazon Kendra index fields.

Type: Array of [DataSourceToIndexFieldMapping \(p. 467\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

SalesforceStandardObjectConfiguration

Specifies configuration information for indexing a single standard object.

Contents

DocumentDataFieldName

The name of the field in the standard object table that contains the document contents.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z][a-zA-Z0-9_.]*\$

Required: Yes

DocumentTitleFieldName

The name of the field in the standard object table that contains the document title.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z][a-zA-Z0-9_.]*\$

Required: No

FieldMappings

One or more objects that map fields in the standard object to Amazon Kendra index fields. The index field must exist before you can map a Salesforce field to it.

Type: Array of [DataSourceToIndexFieldMapping \(p. 467\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Required: No

Name

The name of the standard object.

Type: String

Valid Values: ACCOUNT | CAMPAIGN | CASE | CONTACT | CONTRACT | DOCUMENT | GROUP | IDEA | LEAD | OPPORTUNITY | PARTNER | PRICEBOOK | PRODUCT | PROFILE | SOLUTION | TASK | USER

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)

- [AWS SDK for Ruby V3](#)

ScoreAttributes

Provides a relative ranking that indicates how confident Amazon Kendra is that the response matches the query.

Contents

ScoreConfidence

A relative ranking for how well the response matches the query.

Type: String

Valid Values: `VERY_HIGH` | `HIGH` | `MEDIUM` | `LOW` | `NOT_AVAILABLE`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Search

Provides information about how a custom index field is used during a search.

Contents

Displayable

Determines whether the field is returned in the query response. The default is `true`.

Type: Boolean

Required: No

Facetable

Indicates that the field can be used to create search facets, a count of results for each value in the field. The default is `false`.

Type: Boolean

Required: No

Searchable

Determines whether the field is used in the search. If the `Searchable` field is `true`, you can use relevance tuning to manually tune how Amazon Kendra weights the field in the search. The default is `true` for string fields and `false` for number and date fields.

Type: Boolean

Required: No

Sortable

Determines whether the field can be used to sort the results of a query. If you specify sorting on a field that does not have `Sortable` set to `true`, Amazon Kendra returns an exception. The default is `false`.

Type: Boolean

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

SeedUrlConfiguration

Provides the configuration information of the seed or starting point URLs to crawl.

When selecting websites to index, you must adhere to the [Amazon Acceptable Use Policy](#) and all other Amazon terms. Remember that you must only use Amazon Kendra Web Crawler to index your own webpages, or webpages that you have authorization to index.

Contents

SeedUrls

The list of seed or starting point URLs of the websites you want to crawl.

The list can include a maximum of 100 seed URLs.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^(https?):\/\/([^\s]*)`

Required: Yes

WebCrawlerMode

You can choose one of the following modes:

- `HOST_ONLY` – crawl only the website host names. For example, if the seed URL is "abc.example.com", then only URLs with host name "abc.example.com" are crawled.
- `SUBDOMAINS` – crawl the website host names with subdomains. For example, if the seed URL is "abc.example.com", then "a.abc.example.com" and "b.abc.example.com" are also crawled.
- `EVERYTHING` – crawl the website host names with subdomains and other domains that the webpages link to.

The default mode is set to `HOST_ONLY`.

Type: String

Valid Values: `HOST_ONLY` | `SUBDOMAINS` | `EVERYTHING`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ServerSideEncryptionConfiguration

Provides the identifier of the AWS KMScustomer master key (CMK) used to encrypt data indexed by Amazon Kendra. Amazon Kendra doesn't support asymmetric CMKs.

Contents

KmsKeyId

The identifier of the AWS KMScustomer master key (CMK). Amazon Kendra doesn't support asymmetric CMKs.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ServiceNowConfiguration

Provides configuration information required to connect to a ServiceNow data source.

Contents

AuthenticationType

Determines the type of authentication used to connect to the ServiceNow instance. If you choose `HTTP_BASIC`, Amazon Kendra is authenticated using the user name and password provided in the AWS Secrets Manager secret in the `SecretArn` field. When you choose `OAUTH2`, Amazon Kendra is authenticated using the OAuth token and secret provided in the Secrets Manager secret, and the user name and password are used to determine which information Amazon Kendra has access to.

When you use `OAUTH2` authentication, you must generate a token and a client secret using the ServiceNow console. For more information, see [Using a ServiceNow data source](#).

Type: String

Valid Values: `HTTP_BASIC` | `OAUTH2`

Required: No

HostUrl

The ServiceNow instance that the data source connects to. The host endpoint should look like the following: `{instance}.service-now.com`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^(!^(https?|ftp|file):\/\/)[a-z0-9-]+(\.service-now\.com)$`

Required: Yes

KnowledgeArticleConfiguration

Provides configuration information for crawling knowledge articles in the ServiceNow site.

Type: [ServiceNowKnowledgeArticleConfiguration \(p. 532\)](#) object

Required: No

SecretArn

The Amazon Resource Name (ARN) of the AWS Secrets Manager secret that contains the user name and password required to connect to the ServiceNow instance.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: `arn:[a-z0-9-\.]{1,63}:[a-z0-9-\.]{0,63}:[a-z0-9-\.]{0,63}:[a-z0-9-\.]{0,63}:[^/].{0,1023}`

Required: Yes

ServiceCatalogConfiguration

Provides configuration information for crawling service catalogs in the ServiceNow site.

Type: [ServiceNowServiceCatalogConfiguration \(p. 534\)](#) object

Required: No

ServiceNowBuildVersion

The identifier of the release that the ServiceNow host is running. If the host is not running the LONDON release, use OTHERS.

Type: String

Valid Values: LONDON | OTHERS

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ServiceNowKnowledgeArticleConfiguration

Provides configuration information for crawling knowledge articles in the ServiceNow site.

Contents

CrawlAttachments

Indicates whether Amazon Kendra should index attachments to knowledge articles.

Type: Boolean

Required: No

DocumentDataFieldName

The name of the ServiceNow field that is mapped to the index document contents field in the Amazon Kendra index.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z][a-zA-Z0-9_.]*\$

Required: Yes

DocumentTitleFieldName

The name of the ServiceNow field that is mapped to the index document title field.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z][a-zA-Z0-9_.]*\$

Required: No

ExcludeAttachmentFilePatterns

List of regular expressions applied to knowledge articles. Items that don't match the inclusion pattern are not indexed. The regex is applied to the field specified in the `PatternTargetField`

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 150.

Required: No

FieldMappings

Mapping between ServiceNow fields and Amazon Kendra index fields. You must create the index field before you map the field.

Type: Array of [DataSourceToIndexFieldMapping \(p. 467\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Required: No

FilterQuery

A query that selects the knowledge articles to index. The query can return articles from multiple knowledge bases, and the knowledge bases can be public or private.

The query string must be one generated by the ServiceNow console. For more information, see [Specifying documents to index with a query](#).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: ^\P{C}*\$

Required: No

IncludeAttachmentFilePatterns

List of regular expressions applied to knowledge articles. Items that don't match the inclusion pattern are not indexed. The regex is applied to the field specified in the `PatternTargetField`.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 150.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ServiceNowServiceCatalogConfiguration

Provides configuration information for crawling service catalog items in the ServiceNow site

Contents

CrawlAttachments

Indicates whether Amazon Kendra should crawl attachments to the service catalog items.

Type: Boolean

Required: No

DocumentDataFieldName

The name of the ServiceNow field that is mapped to the index document contents field in the Amazon Kendra index.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z][a-zA-Z0-9_.]*\$

Required: Yes

DocumentTitleFieldName

The name of the ServiceNow field that is mapped to the index document title field.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z][a-zA-Z0-9_.]*\$

Required: No

ExcludeAttachmentFilePatterns

A list of regular expression patterns. Documents that match the patterns are excluded from the index. Documents that don't match the patterns are included in the index. If a document matches both an exclusion pattern and an inclusion pattern, the document is not included in the index.

The regex is applied to the file name of the attachment.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 150.

Required: No

FieldMappings

Mapping between ServiceNow fields and Amazon Kendra index fields. You must create the index field before you map the field.

Type: Array of [DataSourceToIndexFieldMapping \(p. 467\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Required: No

IncludeAttachmentFilePatterns

A list of regular expression patterns. Documents that match the patterns are included in the index. Documents that don't match the patterns are excluded from the index. If a document matches both an exclusion pattern and an inclusion pattern, the document is not included in the index.

The regex is applied to the file name of the attachment.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 150.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

SharePointConfiguration

Provides configuration information for connecting to a Microsoft SharePoint data source.

Contents

CrawlAttachments

TRUE to include attachments to documents stored in your Microsoft SharePoint site in the index; otherwise, FALSE.

Type: Boolean

Required: No

DisableLocalGroups

A Boolean value that specifies whether local groups are disabled (`True`) or enabled (`False`).

Type: Boolean

Required: No

DocumentTitleFieldName

The Microsoft SharePoint attribute field that contains the title of the document.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: ^[a-zA-Z][a-zA-Z0-9_.]*\$

Required: No

ExclusionPatterns

A list of regular expression patterns. Documents that match the patterns are excluded from the index. Documents that don't match the patterns are included in the index. If a document matches both an exclusion pattern and an inclusion pattern, the document is not included in the index.

The regex is applied to the display URL of the SharePoint document.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 150.

Required: No

FieldMappings

A list of `DataSourceToIndexFieldMapping` objects that map Microsoft SharePoint attributes to custom fields in the Amazon Kendra index. You must first create the index fields using the `UpdateIndex` operation before you map SharePoint attributes. For more information, see [Mapping Data Source Fields](#).

Type: Array of [DataSourceToIndexFieldMapping](#) (p. 467) objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Required: No

InclusionPatterns

A list of regular expression patterns. Documents that match the patterns are included in the index. Documents that don't match the patterns are excluded from the index. If a document matches both an inclusion pattern and an exclusion pattern, the document is not included in the index.

The regex is applied to the display URL of the SharePoint document.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 150.

Required: No

SecretArn

The Amazon Resource Name (ARN) of credentials stored in AWS Secrets Manager. The credentials should be a user/password pair. If you use SharePoint Server, you also need to provide the sever domain name as part of the credentials. For more information, see [Using a Microsoft SharePoint Data Source](#). For more information about AWS Secrets Manager see [What Is AWS Secrets Manager](#) in the [AWS Secrets Manager](#) user guide.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1284.

Pattern: `arn:[a-z0-9-\.]{1,63}:[a-z0-9-\.]{0,63}:[a-z0-9-\.]{0,63}:[a-z0-9-\.]{0,63}:[^/].{0,1023}`

Required: Yes

SharePointVersion

The version of Microsoft SharePoint that you are using as a data source.

Type: String

Valid Values: `SHAREPOINT_2013` | `SHAREPOINT_2016` | `SHAREPOINT_ONLINE`

Required: Yes

SslCertificateS3Path

Information required to find a specific file in an Amazon S3 bucket.

Type: [S3Path \(p. 513\)](#) object

Required: No

Urls

The URLs of the Microsoft SharePoint site that contains the documents that should be indexed.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^(https?|ftp|file):\/\/([^\s]*)`

Required: Yes

UseChangeLog

Set to TRUE to use the Microsoft SharePoint change log to determine the documents that need to be updated in the index. Depending on the size of the SharePoint change log, it may take longer for Amazon Kendra to use the change log than it takes it to determine the changed documents using the Amazon Kendra document crawler.

Type: Boolean

Required: No

VpcConfiguration

Provides information for connecting to an Amazon VPC.

Type: [DataSourceVpcConfiguration \(p. 468\)](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

SiteMapsConfiguration

Provides the configuration information of the sitemap URLs to crawl.

When selecting websites to index, you must adhere to the [Amazon Acceptable Use Policy](#) and all other Amazon terms. Remember that you must only use Amazon Kendra Web Crawler to index your own webpages, or webpages that you have authorization to index.

Contents

SiteMaps

The list of sitemap URLs of the websites you want to crawl.

The list can include a maximum of three sitemap URLs.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 3 items.

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^(https?):\/\/([^\s]*)`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

SortingConfiguration

Specifies the document attribute to use to sort the response to a Amazon Kendra query. You can specify a single attribute for sorting. The attribute must have the `Sortable` flag set to `true`, otherwise Amazon Kendra returns an exception.

You can sort attributes of the following types.

- Date value
- Long value
- String value

You can't sort attributes of the following type.

- String list value

Contents

DocumentAttributeKey

The name of the document attribute used to sort the response. You can use any field that has the `Sortable` flag set to true.

You can also sort by any of the following built-in attributes:

- `_category`
- `_created_at`
- `_last_updated_at`
- `_version`
- `_view_count`

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: `[a-zA-Z0-9_][a-zA-Z0-9_-]*`

Required: Yes

SortOrder

The order that the results should be returned in. In case of ties, the relevance assigned to the result by Amazon Kendra is used as the tie-breaker.

Type: String

Valid Values: `DESC` | `ASC`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)

- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

SqlConfiguration

Provides information that configures Amazon Kendra to use a SQL database.

Contents

QueryIdentifiersEnclosingOption

Determines whether Amazon Kendra encloses SQL identifiers for tables and column names in double quotes ("") when making a database query.

By default, Amazon Kendra passes SQL identifiers the way that they are entered into the data source configuration. It does not change the case of identifiers or enclose them in quotes.

PostgreSQL internally converts uppercase characters to lower case characters in identifiers unless they are quoted. Choosing this option encloses identifiers in quotes so that PostgreSQL does not convert the character's case.

For MySQL databases, you must enable the `ansi_quotes` option when you set this field to `DOUBLE_QUOTES`.

Type: String

Valid Values: `DOUBLE_QUOTES` | `NONE`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Status

Provides information about the status of documents submitted for indexing.

Contents

DocumentId

The unique identifier of the document.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: No

DocumentStatus

The current status of a document.

If the document was submitted for deletion, the status is NOT_FOUND after the document is deleted.

Type: String

Valid Values: NOT_FOUND | PROCESSING | INDEXED | UPDATED | FAILED | UPDATE_FAILED

Required: No

FailureCode

Indicates the source of the error.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: No

FailureReason

Provides detailed information about why the document couldn't be indexed. Use this information to correct the error before you resubmit the document for indexing.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Suggestion

A single query suggestion.

Contents

Id

The unique UUID (universally unique identifier) of a single query suggestion.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 73.

Required: No

Value

The value for the unique UUID (universally unique identifier) of a single query suggestion.

The value is the text string of a suggestion.

Type: [SuggestionValue \(p. 547\)](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

SuggestionHighlight

The text highlights for a single query suggestion.

Contents

BeginOffset

The zero-based location in the response string where the highlight starts.

Type: Integer

Required: No

EndOffset

The zero-based location in the response string where the highlight ends.

Type: Integer

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

SuggestionTextWithHighlights

Provides text and information about where to highlight the query suggestion text.

Contents

Highlights

The beginning and end of the query suggestion text that should be highlighted.

Type: Array of [SuggestionHighlight \(p. 545\)](#) objects

Required: No

Text

The query suggestion text to display to the user.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

SuggestionValue

The `SuggestionTextWithHighlights` structure information.

Contents

Text

The `SuggestionTextWithHighlights` structure that contains the query suggestion text and highlights.

Type: [SuggestionTextWithHighlights \(p. 546\)](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Tag

A list of key/value pairs that identify an index, FAQ, or data source. Tag keys and values can consist of Unicode letters, digits, white space, and any of the following symbols: _ . : / = + - @.

Contents

Key

The key for the tag. Keys are not case sensitive and must be unique for the index, FAQ, or data source.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

Value

The value associated with the tag. The value may be an empty string but it can't be null.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 256.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

TextDocumentStatistics

Provides information about text documents indexed in an index.

Contents

IndexedTextBytes

The total size, in bytes, of the indexed documents.

Type: Long

Valid Range: Minimum value of 0.

Required: Yes

IndexedTextDocumentsCount

The number of text documents indexed.

Type: Integer

Valid Range: Minimum value of 0.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

TextWithHighlights

Provides text and information about where to highlight the text.

Contents

Highlights

The beginning and end of the text that should be highlighted.

Type: Array of [Highlight \(p. 490\)](#) objects

Required: No

Text

The text to display to the user.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ThesaurusSummary

An array of summary information for a thesaurus or multiple thesauri.

Contents

CreatedAt

The Unix datetime that the thesaurus was created.

Type: Timestamp

Required: No

Id

The identifier of the thesaurus.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: No

Name

The name of the thesaurus.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_-]*

Required: No

Status

The status of the thesaurus.

Type: String

Valid Values: CREATING | ACTIVE | DELETING | UPDATING |
ACTIVE_BUT_UPDATE_FAILED | FAILED

Required: No

UpdatedAt

The Unix datetime that the thesaurus was last updated.

Type: Timestamp

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)

- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

TimeRange

Provides a range of time.

Contents

EndTime

The UNIX datetime of the end of the time range.

Type: Timestamp

Required: No

StartTime

The UNIX datetime of the beginning of the time range.

Type: Timestamp

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Urls

Provides the configuration information of the URLs to crawl.

You can only crawl websites that use the secure communication protocol, Hypertext Transfer Protocol Secure (HTTPS). If you receive an error when crawling a website, it could be that the website is blocked from crawling.

When selecting websites to index, you must adhere to the [Amazon Acceptable Use Policy](#) and all other Amazon terms. Remember that you must only use Amazon Kendra Web Crawler to index your own webpages, or webpages that you have authorization to index.

Contents

SeedUrlConfiguration

Provides the configuration of the seed or starting point URLs of the websites you want to crawl.

You can choose to crawl only the website host names, or the website host names with subdomains, or the website host names with subdomains and other domains that the webpages link to.

You can list up to 100 seed URLs.

Type: [SeedUrlConfiguration \(p. 528\)](#) object

Required: No

SiteMapsConfiguration

Provides the configuration of the sitemap URLs of the websites you want to crawl.

Only URLs belonging to the same website host names are crawled. You can list up to three sitemap URLs.

Type: [SiteMapsConfiguration \(p. 539\)](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

UserContext

Provides information about the user context for an Amazon Kendra index.

This is used for filtering search results for different users based on their access to documents.

You provide one of the following:

- User token
- User ID, the groups the user belongs to, and any data sources the groups can access.

If you provide both, an exception is thrown.

Contents

DataSourceGroups

The list of data source groups you want to filter search results based on groups' access to documents in that data source.

Type: Array of [DataSourceGroup \(p. 459\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 2048 items.

Required: No

Groups

The list of groups you want to filter search results based on the groups' access to documents.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 2048 items.

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: ^\P{C}*\$

Required: No

Token

The user context token for filtering search results for a user. It must be a JWT or a JSON token.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100000.

Pattern: ^\P{C}*\$

Required: No

UserId

The identifier of the user you want to filter search results based on their access to documents.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 200.

Pattern: ^\P{C}*\$

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

UserGroupResolutionConfiguration

Provides the configuration information to fetch access levels of groups and users from an AWS Single Sign-On identity source. This is useful for setting up user context filtering, where Amazon Kendra filters search results for different users based on their group's access to documents. You can also map your users to their groups for user context filtering using the [PutPrincipalMapping](#) operation.

To set up an AWS SSO identity source in the console to use with Amazon Kendra, see [Getting started with an AWS SSO identity source](#). You must also grant the required permissions to use AWS SSO with Amazon Kendra. For more information, see [IAM roles for AWS SSO](#).

Contents

UserGroupResolutionMode

The identity store provider (mode) you want to use to fetch access levels of groups and users. AWS Single Sign-On is currently the only available mode. Your users and groups must exist in an AWS SSO identity source in order to use this mode.

Type: String

Valid Values: AWS_SSO | NONE

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

UserTokenConfiguration

Provides configuration information for a token configuration.

Contents

JsonTokenTypeConfiguration

Information about the JSON token type configuration.

Type: [JsonTokenTypeConfiguration \(p. 494\)](#) object

Required: No

JwtTokenTypeConfiguration

Information about the JWT token type configuration.

Type: [JwtTokenTypeConfiguration \(p. 495\)](#) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

WebCrawlerConfiguration

Provides the configuration information required for Amazon Kendra Web Crawler.

Contents

AuthenticationConfiguration

Provides configuration information required to connect to websites using authentication.

You can connect to websites using basic authentication of user name and password.

You must provide the website host name and port number. For example, the host name of `https://a.example.com/page1.html` is "a.example.com" and the port is 443, the standard port for HTTPS. You use a secret in [AWS Secrets Manager](#) to store your authentication credentials.

Type: [AuthenticationConfiguration \(p. 432\)](#) object

Required: No

CrawlDepth

Specifies the number of levels in a website that you want to crawl.

The first level begins from the website seed or starting point URL. For example, if a website has 3 levels – index level (i.e. seed in this example), sections level, and subsections level – and you are only interested in crawling information up to the sections level (i.e. levels 0-1), you can set your depth to 1.

The default crawl depth is set to 2.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 10.

Required: No

MaxContentSizePerPageInMegabytes

The maximum size (in MB) of a webpage or attachment to crawl.

Files larger than this size (in MB) are skipped/not crawled.

The default maximum size of a webpage or attachment is set to 50 MB.

Type: Float

Valid Range: Minimum value of 1.0e-06. Maximum value of 50.

Required: No

MaxLinksPerPage

The maximum number of URLs on a webpage to include when crawling a website. This number is per webpage.

As a website's webpages are crawled, any URLs the webpages link to are also crawled. URLs on a webpage are crawled in order of appearance.

The default maximum links per page is 100.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000.

Required: No

MaxUrlsPerMinuteCrawlRate

The maximum number of URLs crawled per website host per minute.

A minimum of one URL is required.

The default maximum number of URLs crawled per website host per minute is 300.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 300.

Required: No

ProxyConfiguration

Provides configuration information required to connect to your internal websites via a web proxy.

You must provide the website host name and port number. For example, the host name of https://a.example.com/page1.html is "a.example.com" and the port is 443, the standard port for HTTPS.

Web proxy credentials are optional and you can use them to connect to a web proxy server that requires basic authentication. To store web proxy credentials, you use a secret in [AWS Secrets Manager](#).

Type: [ProxyConfiguration \(p. 503\)](#) object

Required: No

UrlExclusionPatterns

The regular expression pattern to exclude certain URLs to crawl.

If there is a regular expression pattern to include certain URLs that conflicts with the exclude pattern, the exclude pattern takes precedence.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 150.

Required: No

UrlInclusionPatterns

The regular expression pattern to include certain URLs to crawl.

If there is a regular expression pattern to exclude certain URLs that conflicts with the include pattern, the exclude pattern takes precedence.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 150.

Required: No

Urls

Specifies the seed or starting point URLs of the websites or the sitemap URLs of the websites you want to crawl.

You can include website subdomains. You can list up to 100 seed URLs and up to three sitemap URLs.

You can only crawl websites that use the secure communication protocol, Hypertext Transfer Protocol Secure (HTTPS). If you receive an error when crawling a website, it could be that the website is blocked from crawling.

When selecting websites to index, you must adhere to the [Amazon Acceptable Use Policy](#) and all other Amazon terms. Remember that you must only use Amazon Kendra Web Crawler to index your own webpages, or webpages that you have authorization to index.

Type: [Urls \(p. 554\)](#) object

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

WorkDocsConfiguration

Provides the configuration information to connect to Amazon WorkDocs as your data source.

Amazon WorkDocs connector is available in Oregon, North Virginia, Sydney, Singapore and Ireland regions.

Contents

CrawlComments

TRUE to include comments on documents in your index. Including comments in your index means each comment is a document that can be searched on.

The default is set to FALSE.

Type: Boolean

Required: No

ExclusionPatterns

A list of regular expression patterns to exclude certain files in your Amazon WorkDocs site repository. Files that match the patterns are excluded from the index. Files that don't match the patterns are included in the index. If a file matches both an inclusion pattern and an exclusion pattern, the exclusion pattern takes precedence and the file isn't included in the index.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 150.

Required: No

FieldMappings

A list of `DataSourceToIndexFieldMapping` objects that map Amazon WorkDocs field names to custom index field names in Amazon Kendra. You must first create the custom index fields using the `UpdateIndex` operation before you map to Amazon WorkDocs fields. For more information, see [Mapping Data Source Fields](#). The Amazon WorkDocs data source field names need to exist in your Amazon WorkDocs custom metadata.

Type: Array of [DataSourceToIndexFieldMapping \(p. 467\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Required: No

InclusionPatterns

A list of regular expression patterns to include certain files in your Amazon WorkDocs site repository. Files that match the patterns are included in the index. Files that don't match the patterns are excluded from the index. If a file matches both an inclusion pattern and an exclusion pattern, the exclusion pattern takes precedence and the file isn't included in the index.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 150.

Required: No

OrganizationId

The identifier of the directory corresponding to your Amazon WorkDocs site repository.

You can find the organization ID in the [AWS Directory Service](#) by going to **Active Directory**, then **Directories**. Your Amazon WorkDocs site directory has an ID, which is the organization ID. You can also set up a new Amazon WorkDocs directory in the AWS Directory Service console and enable a Amazon WorkDocs site for the directory in the Amazon WorkDocs console.

Type: String

Length Constraints: Fixed length of 12.

Pattern: d-[0-9a-fA-F]{10}

Required: Yes

UseChangeLog

`TRUE` to use the change logs to update documents in your index instead of scanning all documents.

If you are syncing your Amazon WorkDocs data source with your index for the first time, all documents are scanned. After your first sync, you can use the change logs to update your documents in your index for future syncs.

The default is set to `FALSE`.

Type: Boolean

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Common Errors

This section lists the errors common to the API actions of all AWS services. For errors specific to an API action for this service, see the topic for that API action.

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

IncompleteSignature

The request signature does not conform to AWS standards.

HTTP Status Code: 400

InternalFailure

The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

InvalidAction

The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 400

InvalidClientTokenId

The X.509 certificate or AWS access key ID provided does not exist in our records.

HTTP Status Code: 403

InvalidParameterCombination

Parameters that must not be used together were used together.

HTTP Status Code: 400

InvalidParameterValue

An invalid or out-of-range value was supplied for the input parameter.

HTTP Status Code: 400

InvalidQueryParameter

The AWS query string is malformed or does not adhere to AWS standards.

HTTP Status Code: 400

MalformedQueryString

The query string contains a syntax error.

HTTP Status Code: 404

MissingAction

The request is missing an action or a required parameter.

HTTP Status Code: 400

MissingAuthenticationToken

The request must contain either a valid (registered) AWS access key ID or X.509 certificate.

HTTP Status Code: 403

MissingParameter

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

NotAuthorized

You do not have permission to perform this action.

HTTP Status Code: 400

OptInRequired

The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

RequestExpired

The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

ServiceUnavailable

The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 400

ValidationException

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400

Common Parameters

The following list contains the parameters that all actions use for signing Signature Version 4 requests with a query string. Any action-specific parameters are listed in the topic for that action. For more information about Signature Version 4, see [Signature Version 4 Signing Process](#) in the *Amazon Web Services General Reference*.

Action

The action to be performed.

Type: string

Required: Yes

Version

The API version that the request is written for, expressed in the format YYYY-MM-DD.

Type: string

Required: Yes

X-Amz-Algorithm

The hash algorithm that you used to create the request signature.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Valid Values: AWS4-HMAC-SHA256

Required: Conditional

X-Amz-Credential

The credential scope value, which is a string that includes your access key, the date, the region you are targeting, the service you are requesting, and a termination string ("aws4_request"). The value is expressed in the following format: `access_key/YYYYMMDD/region/service/aws4_request`.

For more information, see [Task 2: Create a String to Sign for Signature Version 4](#) in the *Amazon Web Services General Reference*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-Date

The date that is used to create the signature. The format must be ISO 8601 basic format (YYYYMMDD'T'HHMMSS'Z'). For example, the following date time is a valid X-Amz-Date value: `20120325T120000Z`.

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see [Handling Dates in Signature Version 4](#) in the *Amazon Web Services General Reference*.

Type: string

Required: Conditional

X-Amz-Security-Token

The temporary security token that was obtained through a call to AWS Security Token Service (AWS STS). For a list of services that support temporary security credentials from AWS Security Token Service, go to [AWS Services That Work with IAM](#) in the *IAM User Guide*.

Condition: If you're using temporary security credentials from the AWS Security Token Service, you must include the security token.

Type: string

Required: Conditional

X-Amz-Signature

Specifies the hex-encoded signature that was calculated from the string to sign and the derived signing key.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-SignedHeaders

Specifies all the HTTP headers that were included as part of the canonical request. For more information about specifying signed headers, see [Task 1: Create a Canonical Request For Signature Version 4](#) in the *Amazon Web Services General Reference*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

AWS glossary

For the latest AWS terminology, see the [AWS glossary](#) in the *AWS General Reference*.