Salesforce Knowledge Developer Guide

Version 66.0, Spring '26

Spring '26

CONTENTS

Contents

CHAPTER 1 Developing with Salesforce Knowledge

Salesforce Knowledge gives your website visitors, clients, partners, and service agents the ultimate in support. You can create and manage your company information and securely share it when and where it is needed. Most features are declarative and let you point-and-click your way to a successful implementation. However, the API allows you to code your way to case deflection, happier customers, and more productive support agents. This guide discusses both Lightning Knowledge and Knowledge in Salesforce Classic.

Knowledge Object Model

The Knowledge object model is slightly different from other feature areas because it exposes a set of abstract Salesforce objects that aren't directly used when you create articles. These abstract objects then contain concrete derivations that you do use when creating articles.
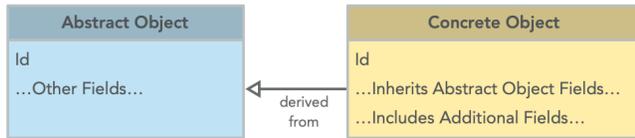
This architecture gives more flexibility to search behavior in support of Lightning Knowledge and Salesforce Classic.

Core Knowledge Objects

The core abstract Knowledge objects are for a Knowledge article (KnowledgeArticle), and a set of versions of that article (KnowledgeArticleVersion). These versions can include different translations and historical versions of an article.
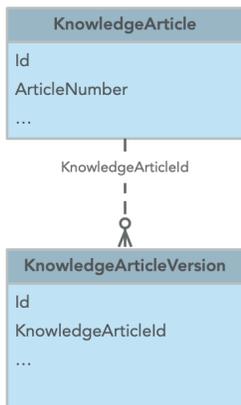
- KnowledgeArticle—The abstract Knowledge article object. This object contains metadata about a knowledge article irrespective of the article's version, translation, or status.
- KnowledgeArticleVersion—The abstract Knowledge article version object. Whenever you create a draft of an article, it gets a new version number and a unique ID. Each translation of an article also has a version and ID. Each version is a KnowledgeArticleVersion record with the KnowledgeArticle record as the parent.

When you create knowledge articles, you create records with concrete Salesforce objects. These objects inherit from their abstract counterparts. By default, these concrete objects are named "Knowledge__ka" for knowledge articles and "Knowledge__kav" for knowledge article versions in Lightning Knowledge.
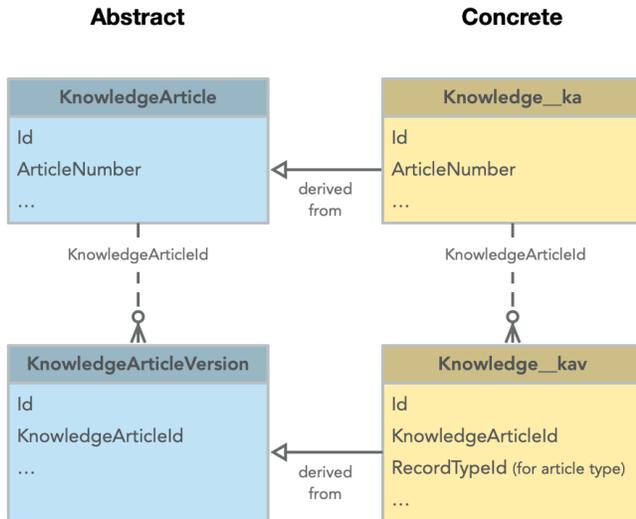
This diagram shows the default names in Lightning Knowledge for the concrete implementations, but you can change these names. The names always have the suffixes "__ka" for knowledge articles and "__kav" for knowledge article versions. You can change the "Knowledge" prefix by changing the Object Name for the Knowledge__kav object in the Object Manager.

The concrete implementation contains any of your custom fields for a knowledge article. The Knowledge__kav object also contains a "RecordTypeId" field, which is a record type used to describe the article structure, such as FAQ or Tutorial. Each record type can contain its own layouts. For example, an FAQ record type can use a special layout that displays custom fields such as "Question__c" and "Answer__c".

In Salesforce Classic, the article structure is determined by the article type. Each knowledge article type has a different concrete implementation, for example FAQ__kav, Tutorial__kav. In Lightning Knowledge, one concrete object is used, for example, Knowledge__kav, and instead the record type is used to associate an article with a unique article structure.

Complete Knowledge Object Diagram

To see the relationship of Knowledge articles to other objects, let's highlight some of the other important object relationships. In this diagram, the default value "Knowledge" is the article type name to illustrate the object mode. If you change this value in Setup, the prefix for each object with "__" in the name is changed accordingly.

These objects appear in this Knowledge object model. Each object links to its reference documentation for more information.

- **Case**—A case, which is a customer issue or problem. Every article can be associated with a case using the **CaseArticle** object.

- **CaseArticle**—The association between a **Case** and a **KnowledgeArticle** on page 11.

- **FeedComment**—A comment added to a feed for a knowledge article.

- **Knowledge__DataCategorySelection** on page 25—A data category that classifies an article.

- **Knowledge__Feed** on page 26—The feed for a knowledge article.

- **Knowledge__ka** on page 40—The concrete object of a **KnowledgeArticle** on page 11.

- **Knowledge__kav** on page 31—The concrete object of a **KnowledgeArticleVersion** on page 14.

- Knowledge__VersionHistory—The concrete object of a **KnowledgeArticleVersionHistory**.

- Knowledge__ViewStat—The concrete object of a **KnowledgeArticleViewStat**.

- Knowledge__VoteStat—The concrete object of a **KnowledgeArticleVoteStat**.

- **KnowledgeArticle** on page 11—The abstract Knowledge article object. This object contains metadata about a knowledge article irrespective of the article's version, translation, or status.

- KnowledgeArticleVersion on page 14—The abstract Knowledge article version object. Whenever you create a draft of an article, it gets a new version number and a unique ID. Each translation of an article also has a version and ID. Each version is a KnowledgeArticleVersion on page 14 record with the KnowledgeArticle on page 11 record as the parent.

- KnowledgeArticleVersionHistory—Enables read-only access to the full history of an article.

- KnowledgeArticleViewStat—Provides statistics on the number of views for the specified article.

- KnowledgeArticleVoteStat—Provides the weighted rating for the specified article.

- LinkedArticle—A knowledge article that is attached to a work order, work order line item, or work type.

- LinkedArticleFeed—The comment feed on a linked article.

- LinkedArticleHistory—The history of changes made to tracked fields on a linked article.

- WorkOrder—Field service work to be performed for a customer.

- WorkOrderLineItem—A subtask on a work order in field service.

- Vote—A vote that a user has made on a knowledge article.

  - Using Vote is allowed for Knowledge Article objects but not for Knowledge Article Version objects.

  - Only one vote record can exist per knowledge article per user.Voting isn't allowed by Site Guest Users.

  - Vote allows multiple types of values for the "Type" field, but only "Up" and "Down" are allowed for Knowledge Articles.

Knowledge Articles and Data Categories in the API

Before working with the Knowledge API, learn the basics about knowledge articles and data categories.

Articles Overview

🛑 **Important:** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Articles capture information about your company's products and services that you want to make available in your knowledge base. Articles in the knowledge base can be classified using data categories to make it easy for users to find the articles they need. Administrators can use data categories to control access to articles.

Knowledge Articles vs. Knowledge Article Versions

When working with articles, keep in mind that the KnowledgeArticle represents the parent record of all article versions. KnowledgeArticleVersion records represent each version of a given article.

Record Types vs. Article Types

The article structure is represented differently between Lightning Experience and Salesforce Classic. In Lightning Knowledge, you use the same record types available in other custom objects (see the `RecordTypeId` field on Knowledge__kav) to structure different types of articles. For example, you can use different layouts for different record types. In Salesforce Classic, you get this functionality through article types (see the `ArticleType` field on KnowledgeArticleVersion). Each unique type of article has a unique object in Salesforce Classic (for example, FAQ__kav for FAQ article types). Lightning Knowledge does not have a unique object for each type because it is handled using the record type.

Audience Channel

An audience, sometimes called a channel, refers to the types of users who can access an article. Salesforce Knowledge offers four channels where you can make articles available.

- Internal App: Salesforce users can access articles depending on their role visibility.

- Customer: Customers can access articles in a community, site, or customer portal. Customer users inherit the role visibility of the manager on the account. In a community, the article is only available to users with Customer Community or Customer Community Plus licenses.

- Partner: Partners can access articles in a community, site, or partner portal. Partner users inherit the role visibility of the manager on the account. In a community, the article is only available to users with Partner Community licenses.

- Public Knowledge Base: Articles can be made available to anonymous users by creating a public knowledge base. With Lightning Knowledge, most Salesforce orgs use Communities to create a knowledge base. Creating a public knowledge base for Salesforce Knowledge in Salesforce Classic requires Sites and Visualforce.

Publishing Cycle

Salesforce Knowledge Articles move through a publishing cycle from their creation to their deletion. The publishing cycle includes three different statuses: `Draft` is the stage when a new article is being created or an existing one is being updated. Articles with the `Online` status are draft articles that have been published and are now available to their different channels. Eventually, when a published article is at the end of its life, it can be moved to the `Archived` status or sent back to `Draft` to be updated in a subsequent version.

Working with Articles in the API

Articles are available through the KnowledgeArticleVersion and KnowledgeArticle objects in the API. They both represent an article but provide different capabilities.

KnowledgeArticleVersion

Every new draft article in Salesforce Knowledge has a version number. When an article is published and you want to update it, you can create a new `Draft` with a distinct version number. Each version has its own ID. Once the updated version is ready to be published, it replaces the former one and updates the version number. You can access the content of an article version using the KnowledgeArticleVersion object and filter on its Draft or Online status. For example, the following query returns the title of the Draft version of all the articles across all article types in United States English:

```
SELECT Title
FROM KnowledgeArticleVersion
WHERE PublishStatus='Draft'
AND language ='en_US'
```

Articles are also auto-assigned an Article Number, which is not a unique identifier to an individual article, but an identifier to a primary article and all of its available translations.

Note: Both the primary version (the knowledge article with IsMasterLanguage = 1) and the translations are KnowledgeArticleVersion objects.

KnowledgeArticle

Unlike KnowledgeArticleVersion, the ID of a KnowledgeArticle record is identical irrespective of the article's version (status). Where the KnowledgeArticleVersion object provides API access to an article's custom field values, the KnowledgeArticle object provides API access to an article's metadata fields.

The article record is the parent container of all versions of an article, whatever the publishing status (draft, published, archived) and the language. While KnowledgeArticle and KnowledgeArticleVersion represent any article in the knowledge base, use the concrete representation of these objects for the specific articles. In Lightning Knowledge, these concrete representations default to Knowledge__ka (for the Knowledge article) and Knowledge__kav (for the Knowledge article version). In Salesforce Classic, use <Article Type>__ka and <Article Type>__kav.

The following Lightning Knowledge query returns the title for all the published FAQ articles in United States English:

```
SELECT Title
FROM Knowledge__kav
```

```
WHERE PublishStatus='online'
AND Language = 'en_US'
AND RecordTypeId = '<specify RecordTypeId for FAQ here>'
```

The following Lightning Knowledge query returns the ID, title, URL, and content of articles stored in their rich text area fields:

```
SELECT ID, Title, UrlName, RTA2__c
FROM Knowledge_kav
```

The following Salesforce Classic query returns the title for all the published offers in United States English:

```
SELECT Title
FROM FAQ__kav
WHERE PublishStatus='online'
AND language ='en_US'
```

Data Categories Overview

Data categories are organized by category group and let:

- Users classify and find records.
- Administrators control access to records.

Salesforce Knowledge uses data categories to classify articles and make them easier to find. For example, to classify articles by sales regions and products, create two category groups: Sales Regions, Products. The Sales Regions category group could consist of a geographical hierarchy, such as All Sales Regions as the top level and North America, Europe, and Asia at the second level. The Products group could have All Products as the top level and Phones, Computers, and Printers at the second.

Working with Data Categories in the API

The following table lists API resources for working with data categories.

Name	Type	Description
Knowledge__DataCategorySelection	Object	Gives access to article categorization in Lightning Knowledge.
Article Type__DataCategorySelection	Object	Gives access to article categorization in Knowledge for Salesforce Classic.
QuestionDataCategorySelection	Object	Gives access to question categorization.
WITH DATA CATEGORY *filteringExpression*	SOQL clause	Filters articles depending on their status in the publishing cycle and their data categories. For more information, see the Salesforce SOQL and SOSL Reference Guide.
WITH DATA CATEGORY *DataCategorySpec*	SOSL clause	Finds articles based on their categorization. For more information, see the Salesforce SOQL and SOSL Reference Guide.
describeDataCategoryGroups()	Call	Retrieves available category groups for objects specified in the request.
describeDataCategoryGroupStructures()	Call	Retrieves available category groups along with their data category structure for objects specified in the request.
describeDataCategoryGroups	Apex method	Returns a list of the category groups associated with the specified objects. See the Apex Developer Guide.

Name	Type	Description
`describeDataCategoryGroupStructures`	Apex method	Returns available category groups along with their data category structure for objects specified in the request. See the Apex Developer Guide.

API End-of-Life Policy

Salesforce is committed to supporting each API version for a minimum of 3 years from the date of first release. To improve the quality and performance of the API, versions that are over 3 years old sometimes are no longer supported.

Salesforce notifies customers who use an API version scheduled for deprecation at least 1 year before support for the version ends.

> **Note:** Version 20.0 of REST API and SOAP API have now been deprecated and are no longer supported. You can continue to access this legacy API version until Summer '22 is released, at which point this legacy version will be retired and will become unavailable. For more information, see this Knowledge Article: Salesforce Platform API Versions 7.0 through 20.0 Retirement.

> **Note:** Versions 21.0 through 30.0 of REST API and SOAP API will be deprecated in the Summer '22 release. For more information, see this Knowledge Article: Salesforce Platform API Versions 21.0 through 30.0 Retirement.

CHAPTER 2 Salesforce Knowledge SOAP API

You can manage articles and get support information with Salesforce Objects using the SOAP API.

Knowledge Objects

Salesforce platform objects for working with Salesforce Knowledge.

IN THIS SECTION:

KnowledgeArticle

Provides read-only access to an article and the ability to delete the primary article. This object is available in API version 19.0 and later.

KnowledgeArticleVersion

Provides a global view of standard article fields across all types of articles depending on their version. This object is available in API version 18.0 and later.

Knowledge__DataCategorySelection

Represents a data category that classifies an article. This object is available in API version 39.0 and later.

Knowledge__Feed

Represents the feed for a knowledge article. This object is available in API version 39.0 and later.

Knowledge__kav

Provides access to the concrete object that represents a Knowledge article version. This object is available in API version 39.0 and later.

Knowledge__ka

Provides access to the concrete object that represents a Knowledge article, the parent object for article versions. This object is available in API version 39.0 and later.

KnowledgeArticleVersionHistory

Enables read-only access to the full history of an article. This object is available in API version 25.0 and later.

KnowledgeArticleViewStat

Provides certain statistics related to the number of views for the specified article across all article types. The view count statistics are for published and archived articles only. View counts for draft articles aren't tracked. This object is read-only and available in API version 20.0 and later.

KnowledgeArticleVoteStat

Provides the weighted rating for the specified article on a scale of 1 to 5 across all article types. This object is read-only and available in API version 20.0 and later.

CaseArticle

Represents the association between a Case and a KnowledgeArticle. This object is available in API version 20.0 and later.

LinkedArticle

Represents a knowledge article that is attached to a work order, work order line item, or work type. This object is available in API version 37.0 and later.

RecentlyViewed

Represents records or list views that the current user has recently viewed or referenced (by viewing a related record). List views are available in API version 29.0 and later.

SearchPromotionRule

Represents a promoted search term, which is one or more keywords that you associate with a Salesforce Knowledge article. When a user's search query includes these keywords, the associated article is returned first in search results. This object is available in API version 31.0 and later.

TopicAssignment

Represents the assignment of a topic to a specific feed item, record, or file. This object is available in API version 28.0 and later.

KnowledgeArticle

Provides read-only access to an article and the ability to delete the primary article. This object is available in API version 19.0 and later.

⚠ **Important:** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Unlike KnowledgeArticleVersion, the ID of a KnowledgeArticle record is identical irrespective of the article's version (status).

Knowledge__ka on page 40 is derived from this object.

Supported Calls

describeSObjects(), query(), retrieve()

Special Access Rules

Knowledge must be enabled in your org. A user must have the View Articles permission enabled. Salesforce Knowledge users, unlike customer and partner users, must also be granted the `Knowledge User` feature license.

Fields

Field Name	Details
ArchivedById	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** The ID of the user who archived the article.
ArchivedDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The date when the article was archived.
ArticleNumber	**Type** string **Properties** Autonumber, Defaulted on create, Filter, idLookup, Sort

Field Name	Details
	Description The unique number automatically assigned to the article when it's created. You can't change the format or value for this field.
CaseAssociationCount	**Type** int **Properties** Filter, Group, Sort **Description** The number of cases attached to the article.
FirstPublishedDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The date when the article was first published.
IsGeneratedByLlm	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** True if the first version of an article was created with an LLM. This object is available in API version 59.0 and later.
LastPublishedDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The date when the article was last published.
LastReferencedDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The timestamp when the current user last accessed this record, a record related to this record, or a list view.

Field Name	Details
LastViewedDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The timestamp when the current user last viewed this record or list view. If this value is null, the user has not viewed this record or list view, though they might have accessed it (`LastReferencedDate`)
MasterLanguage	**Type** picklist **Properties** Filter, Group, Restricted picklist, Sort **Description** The article's original language. Only accessible if your knowledge base supports multiple languages.
MigratedToFromArticle	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The ID for the corresponding pre- or post-migration article. Contains values only in orgs that migrate from Knowledge in Salesforce Classic to Lightning Knowledge. This field is available in API version 45.0 and later.
TotalViewCount	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** Total number of views for this article. This field is available in API version 39.0 and later.

Usage

Use this object to query or retrieve articles. KnowledgeArticle can be used in a SOQL clause, but doesn't provide access to the fields from the article. Provides read-only access to an article and the ability to delete the primary article.

Usage for SOQL with KnowledgeArticle

To expose the `migrated_to_from_id` column on KnowledgeArticle and KnowledgeArticleVersion to the sObject API: expose `MigratedToFromArticle` in KnowledgeArticle.

For SOQL:

- To filter by `MigratedToFromArticle`, remove any other filters.
- When filtering by `MigratedToFromArticle`, use the '=' or 'IN' operator.
- When filtering by `MigratedToFromArticle`, the value can't be null or empty.

KnowledgeArticleVersion

Provides a global view of standard article fields across all types of articles depending on their version. This object is available in API version 18.0 and later.

⊘ **Important:** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Use this object to:

- Query or search generically across multiple types of articles.
- Filter on a specific version.
- Update standard fields in draft versions.

When you query on the archived article, the results include both the article and the article's archived versions.

Knowledge__kav on page 31 is derived from this object.

Supported Calls

`describeLayout(), describeSObjects(), query(), retrieve(), search()`

📝 Note:

- You can only update draft versions.
- You can't update draft translations with the `knowledgeManagement` REST API.
- For Lightning Knowledge, to create, update, or delete a Knowledge article version, use the call on Knowledge__kav. For example, to delete, use `Knowledge__kav.delete()`.
- For Knowledge in Salesforce Classic, to create, update, or delete a Knowledge article version, use the call on *ArticleType*__kav, where *ArticleType* is the name of the article's type. For example, to delete, use `ArticleType__kav.delete()`.

Special Access Rules

Knowledge must be enabled in your org. A user must have the View Articles permission enabled. Salesforce Knowledge users, unlike customer and partner users, must also be granted the `Knowledge User` feature license.

Fields

Field Name	Details
ArchivedById	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** The ID of the user who archived the article.
ArchivedDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The date the article version was archived.
ArticleArchivedById	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** The ID of the user who archived the article.
ArticleArchivedDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The date the article was archived.
ArticleCaseAttachCount	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** The number of cases where this article is attached.
ArticleCreatedById	**Type** reference **Properties** Filter, Group, Nillable, Sort

Field Name	Details
	Description The ID of the user who created the article.
ArticleCreatedDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The date the article was created.
ArticleMasterLanguage	**Type** picklist **Properties** Filter, Group, Nillable, Restricted picklist, Sort **Description** The article's original language. Only accessible if your knowledge base supports multiple languages.
ArticleNumber	**Type** string **Properties** Autonumber, Defaulted on create, Filter, Sort **Description** The unique number automatically assigned to the article when it's created. You can't change the format or value for this field.
ArticleTotalViewCount	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** Total number of views for the article.
ArticleType	**Type** string **Properties** Defaulted on createFilter **Description** Indicates the API Name of the article type. The `ArticleType` is assigned to the article when it's created. You can't change the value of this field. This field is available in orgs using Knowledge in Salesforce Classic in API version 26.0 and later.

Field Name	Details
AssignedById	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** The ID of the user who assigned the article.
AssignedToId	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** The ID of the user assigned to the article.
AssignmentDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The date the article was assigned to a user.
AssignmentDueDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The due date when an article is assigned.
AssignmentNote	**Type** textarea **Properties** Filter, Nillable, Sort **Description** Notes to the assignee from the user who assigned the article.
FirstPublishedDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The date when the article was first published.

Field Name	Details
IsLatestVersion	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether the article is the most current version. (`true`) or not (`false`). This field can be `true` on the online or published version, a draft version in the primary language, a draft version in a translation, and the latest archived version. However, you can't filter by (PublishState='Online') and (IsLatestVersion=false) because the online version is also the latest version. This field is available in API version 24.0 and later.
IsMasterLanguage	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether the article has one or more translations associated with it (`true`) or not (`false`). Only accessible if your knowledge base supports multiple languages.
IsOutOfDate	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether the source article has been updated since this translated version was created (`true`) or not (`false`). Only accessible if your knowledge base supports multiple languages.
IsVisibleInApp	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Required. Indicates whether the article is visible in the Articles tab (`true`) or not (`false`).
IsVisibleInCsp	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort

Field Name	Details
	Description Required. Indicates whether the article is visible in the Customer Portal (`true`) or not (`false`).
IsVisibleInPkb	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Required. Indicates whether the article is visible in the public knowledge base (`true`) or not (`false`).
IsVisibleInPrm	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Required. Indicates whether the article is visible in the partner portal (`true`) or not (`false`).
KnowledgeArticleId	**Type** reference **Properties** Filter, Group, Sort **Description** The ID of the article independent from its version. The value for this field is retrieved from the `Id` field of the KnowledgeArticle object.
Language	**Type** picklist **Properties** Filter, Group, Restricted picklist, Sort **Description** The language that the article is written in, such as `French` or `Chinese (Traditional)`. Querying or searching articles in SOSL require that you specify the `Language` field in the WHERE clause. The language must be the same for all article types. Before API version 47.0, you must include the `Language` field to filter queries on Knowledge article versions. In API version 47.0 and later, you can filter queries on Knowledge article versions with or without `Language` depending on what you are querying.

Field Name	Details
LargeLanguageModel	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Shows the LLM used to create an article version. This object is available in API version 59.0 and later.
LastPublishedDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The date when the article was last published.
MasterVersionId	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** ID of the source article, if the article is the translation of a source article. Only accessible if your knowledge base supports multiple languages.
MigratedToFromArticleVersion	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The ID for the corresponding pre- or post-migration article version. Contains values only in orgs that migrate from Classic to Lightning Knowledge. Available in API version 43.0 and later.
NextReviewDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The date when the article must next be reviewed for accuracy. Available in API version 58.0 and later.
OwnerId	**Type** reference

Field Name	Details
	Properties Filter, Group, Sort **Description** The ID of the article's owner.
PublishStatus	**Type** picklist **Properties** Defaulted on create, Filter, Group, Restricted picklist, Sort **Description** The publication status for the article: • `Draft`: any draft articles. • `Online`: articles published in Salesforce Knowledge. • `Archived`: archived articles. A user must have the "Manage Articles" permission enabled to use `Online`. Article queries and searches in SOQL or SOSL require that you specify either the `PublishStatus` or the `Id` field in the WHERE clause. You can search for only one publication status per article type in a single SOSL query. When searching for articles with a `PublishStatus` of `Archived`, also check that `IsLatestVersion` equals `false` in your WHERE clause.
SourceId	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** ID of the source from which the article was created (Case or Reply).
Summary	**Type** textarea **Properties** Filter, Nillable, Sort **Description** Summary of the article. Maximum size is 1000 characters.
Title	**Type** string **Properties** Filter, Group, idLookup, Sort **Description** Required. Article's title. Maximum size is 255 characters.

Field Name	Details
TranslationCompletedDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** Date and time when the article was last translated. Only accessible if your knowledge base supports multiple languages.
TranslationExportedDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** Date and time when the article was last exported for translation. Only accessible if your knowledge base supports multiple languages.
TranslationImportedDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** Date and time when the article was last imported for translation. Only accessible if your knowledge base supports multiple languages.
UrlName	**Type** string **Properties** Filter, Group, idLookup, Sort **Description** Required. Represents the article's URL. Can contain alphanumeric characters and hyphens but can't begin or end with a hyphen. This value must be unique regardless of context. (For example, a unique value allows you to get expected results when running an Apex test with `SeeAllData` set to `false`.) `UrlName` is case-sensitive and its maximum size is 255 characters.
ValidationStatus	**Type** picklist **Properties** Defaulted on create, Filter, Group

Field Name	Details
	Description
	Shows whether the content of the article has been validated. Possible values are `Validated` and `Not Validated`. The default value is `Not Validated`. This field is available in API version 24.0 or later.
VersionNumber	**Type**
	int
	Properties
	Group, Sort
	Description
	The number assigned to a version of an article. This field is available in API version 24.0 and later.

Usage

Use this object to query, retrieve, or search for articles across all types of articles depending on their version. You can update draft primary articles. Also, you can delete articles that aren't drafts. Client applications can use KnowledgeArticleVersion with `describeDataCategoryGroups()` and `describeDataCategoryGroupStructures()` to return the category groups and the category structure associated with Salesforce Knowledge.

To access an article independent of its version, use the KnowledgeArticle object.

In Lightning Knowledge, the type of article is determined by the `RecordType` field on the concrete derived object (for example, Knowledge__kav on page 31). For Knowledge in Salesforce Classic, the type of article is determined by the `ArticleType` field and the concrete derived object uses the prefix of the article type name (for example, FAQ__kav for the FAQ article type).

SOQL Samples

The following SOQL clause uses KnowledgeArticleVersion to query all published articles from all articles complying with the classification specified in the WITH DATA CATEGORY clause:

```
SELECT Title, Summary
FROM KnowledgeArticleVersion
WHERE PublishStatus='Online'
AND Language = 'en_US'
WITH DATA CATEGORY Geography__c ABOVE_OR_BELOW europe__c AND Product__c BELOW All__c
```

The following SOQL clause for Lightning Knowledge uses the `Offer` record type to limit the query to all draft articles:

```
SELECT Id, Title
FROM Knowledge__kav
WHERE PublishStatus='Draft'
AND Language = 'en_US'
AND RecordTypeId = '<specify RecordTypeId for Offer here>'
WITH DATA CATEGORY Geography__c AT (france__c,usa__c) AND Product__c ABOVE dsl__c
```

The following SOQL clause for Salesforce Classic uses the `Offer` article type to limit the query to all draft articles:

```
SELECT Id, Title
FROM Offer__kav
WHERE PublishStatus='Draft'
AND Language = 'en_US'
WITH DATA CATEGORY Geography__c AT (france__c,usa__c) AND Product__c ABOVE dsl__c
```

The following SOQL clause uses KnowledgeArticleVersion to query the IDs of all archived versions of a particular article:

```
SELECT Id
FROM KnowledgeArticleVersion
WHERE PublishStatus='Archived'
AND IsLatestVersion=false
AND KnowledgeArticleId='kA1D00000001PQ6KAM'
```

SOQL and SOSL with KnowledgeArticleVersion

- Filter on a single value of `PublishStatus` for best results. To find all versions of each article, omit the `PublishStatus` filter, but do filter on one or more master key IDs. To retrieve all archived versions for a given article, specify a SOQL filter where `IsLatestVersion` is `false`.

- In API version 46.0 and earlier, queries without a filter on `PublishStatus` return published articles by default. In API version 47.0 and later, draft, published, and archived articles are returned when Lightning Knowledge is enabled.

- To support security, only users with the "View Draft Articles" permission see articles whose `PublishStatus` value is `Draft`. Similarly, only users with the "View Archived Articles" permission see articles whose `PublishStatus` value is `Archived`

- Archived article versions are stored in the **_Knowledge_**`__kav` object. To query archived article versions, specify the article `Id` and set `IsLatestVersion='0'`.

- You can't use binding variables in Apex SOQL statements with KnowledgeArticleVersion objects. For example, the following SOQL statement causes a compilation error.

```
final String PUBLISH_STATUS_ONLINE = 'Online';
List<Knowledge__kav> articles = [
SELECT Id FROM Knowledge__kav
WHERE PublishStatus = :PUBLISH_STATUS_ONLINE
];
```

Instead, use dynamic SOQL as follows. See Dynamic SOQL in *Apex Developer Guide*.

```
final String PUBLISH_STATUS_ONLINE = 'Online';
final String q = 'SELECT Id, PublishStatus FROM Knowledge__kav
WHERE PublishStatus = :PUBLISH_STATUS_ONLINE';
List<Knowledge__kav> articles = Database.query(q);
```

Other Usage for SOQL and SOSL with KnowledgeArticleVersion

To expose the *migrated_to_from_id* on **KnowledgeArticle** and **KnowledgeArticleVersion** to the sObject API: expose **MigratedToFromArticleVersion** in **KnowledgeArticleVersion**.

- For SOQL:
  - To filter by **MigratedToFromArticleVersion**, remove any other filters.
  - When filtering by **MigratedToFromArticleVersion**, use the '=' or 'IN' operator.

– When filtering by **MigratedToFromArticleVersion**, the value can't be null or empty.

- SOSL doesn't support **MigratedToFromArticleVersion**.

Associated Objects

This object has the following associated objects. Unless noted, they're available in the same API version as this object.

KnowledgeArticleVersionHistory

 History is available for tracked fields of the object.

Knowledge__DataCategorySelection

Represents a data category that classifies an article. This object is available in API version 39.0 and later.

> **Note:** By default, the prefix for this object name is `Knowledge` and that is the value shown in this reference. However, this prefix can be modified by changing the **Object Name** for the Knowledge__kav object in Object Manager.

Supported Calls

`create(), delete(), describeSObjects(), getDeleted(), getUpdated(), query(), retrieve()`

Special Access Rules

Lightning Knowledge must be enabled in your org.

Fields

Field	Details
DataCategoryGroupName	**Type** datacategorygrouppreference **Properties** Create **Description** Unique name of the data category group which has categories associated with the article.
DataCategoryName	**Type** datacategorygrouppreference **Properties** Create **Description** Unique name of the data category associated with the article.
ParentId	**Type** reference

Field	Details
	Properties
	Create, Filter, Group, Sort
	Description
	ID of the article associated with the data category selection.

Usage

Every article in Salesforce Knowledge can be categorized. A data category selection represents a category that has been selected to classify an article. You can use this object to query and manage article categorization in your organization. Client applications can create a categorization for an article with a Draft status. They can also delete and query article categorizations.

> **Note:** When using this object to classify an article, you can't select both a category (for example USA) and one of its descendants (California) or ascendant categories (North America). In this case, only the first category is selected.

Knowledge__Feed

Represents the feed for a knowledge article. This object is available in API version 39.0 and later.

For additional information about feeds, see FeedItem.

> **Note:** By default, the prefix for this object name is `Knowledge` and that is the value shown in this reference. However, this prefix can be modified by changing the **Object Name** for the Knowledge__kav object in Object Manager.

Supported Calls

`delete()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`

Special Access Rules

Lightning Knowledge must be enabled in your org.

Fields

Field	Details
BestCommentId	**Type**
	reference
	Properties
	Filter, Group, Nillable, Sort
	Description
	The ID of the comment marked as best answer on a question post.
Body	**Type**
	textarea

Field	Details
	Properties Nillable, Sort **Description** The body of the feed item. Required when `Type` is `TextPost` or `AdvancedTextPost`. Optional when `Type` is `ContentPost` or `LinkPost`. Although a value for `Body` is not required for the `ContentPost` type, an attachment is required. If an attachment isn't present, the type changes to `TextPost` or `AdvancedTextPost`, depending on the API version. `TextPost` and `AdvancedTextPost` do require a value for `Body`. 💡 Tip: See the `IsRichText` field for a list of HTML tags supported in the body of rich text posts.
CommentCount	**Type** int **Properties** Filter, Group, Sort **Description** The number of comments associated with this feed item. 💡 Tip: In a feed that supports pre-moderation, `CommentCount` isn't updated until a comment is published. For example, say that you comment on a post that already has one published comment and your comment triggers moderation. Now there are two comments on the post, but the count says there's only one. In a moderated feed, comments aren't counted until approved by an admin or someone with Can Approve Feed Post and Comment or Modify All Data. Feed moderation has implications on how you retrieve feed comments. In a moderated feed, rather than retrieving comments by looping through `CommentCount`, go through pagination until the end of comments is returned.
InsertedById	**Type** reference **Properties** Group, Nillable, Sort **Description** ID of the user who added this item to the feed. For example, if an application migrates posts and comments from another application into a feed, the `InsertedBy` value is set to the ID of the context user.
IsRichText	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort

Field	Details
	Description
	Indicates whether the feed item `Body` contains rich text. If you post a rich text feed comment using SOAP API, set `IsRichText` to `true` and escape HTML entities from the body. Otherwise, the post is rendered as plain text.
	Rich text supports the following HTML tags:
	• `<p>`
	⚲ **Tip:** Though the ` ` tag isn't supported, you can use `<p> </p>` to create lines.
	• `<a>`
	• `<b>`
	• `<code>`
	• `<i>`
	• `<u>`
	• `<s>`
	• `<ul>`
	• `<ol>`
	• `<li>`
	• `<img>`
	The `<img>` tag is accessible only through the API and must reference files in Salesforce similar to this example: `<img src="sfdc://069B0000000omjh"></img>`
	🗹 **Note:** In API version 35.0 and later, the system replaces special characters in rich text with escaped HTML. In API version 34.0 and prior, all rich text appears as a plain-text representation.
`LikeCount`	**Type**
	int
	Properties
	Filter, Group, Sort
	Description
	The number of likes associated with this feed item.
`LinkUrl`	**Type**
	url
	Properties
	Nillable, Sort
	Description
	The URL of a `LinkPost`.

Field	Details
ParentId	**Type** reference **Properties** Filter, Group, Sort **Description** ID of the Knowledge article to which the feed item is related.
RelatedRecordId	**Type** reference **Properties** Group, Nillable, Sort **Description** ID of the ContentVersion record associated with a `ContentPost`. For WDC thanks posts, it's the ID of the WorkThanks object associated with a `RypplePost`. This field is typically null for all posts except `ContentPost` and `RypplePost`. For example, set this field to an existing ContentVersion ID and post it to a feed with `Type` set to `ContentPost`.
Title	**Type** string **Properties** Group, Nillable, Sort **Description** The title of the feed item. When the `Type` is `LinkPost`, the `LinkUrl` is the URL and this field is the link name. The `Title` field can be updated on posts of `Type` `QuestionPost`.
Type	**Type** picklist **Properties** Filter, Group, Nillable, Restricted picklist, Sort **Description** The type of feed item. Except for `ContentPost`, `LinkPost`, and `TextPost`, don't create feed items of other types directly from the API. • `ActivityEvent`—indirectly generated event when a user or the API adds a Task associated with a feed-enabled parent record (excluding email tasks on cases). Also occurs when a user or the API adds or updates a Task or Event associated with a case record (excluding email and call logging). For a recurring Task with CaseFeed disabled, one event is generated for the series only. For a recurring Task with CaseFeed enabled, events are generated for the series and each occurrence.

Field	Details

- `AdvancedTextPost`—created when a user posts a group announcement and, in Lightning Experience as of API version 39.0 and later, when a user shares a post.
- `AnnouncementPost`—Not used.
- `ApprovalPost`—generated when a user submits an approval.
- `BasicTemplateFeedItem`—Not used.
- `CanvasPost`—a post made by a canvas app posted on a feed.
- `CollaborationGroupCreated`—generated when a user creates a public group.
- `CollaborationGroupUnarchived`—Not used.
- `ContentPost`—a post with an attached file.
- `CreatedRecordEvent`—generated when a user creates a record from the publisher.
- `DashboardComponentAlert`—generated when a dashboard metric or gauge exceeds a user-defined threshold.
- `DashboardComponentSnapshot`—created when a user posts a dashboard snapshot on a feed.
- `LinkPost`—a post with an attached URL.
- `PollPost`—a poll posted on a feed.
- `ProfileSkillPost`—generated when a skill is added to a user's Chatter profile.
- `QuestionPost`—generated when a user posts a question.
- `ReplyPost`—generated when Chatter Answers posts a reply.
- `RypplePost`—generated when a user creates a Thanks badge in WDC.
- `TextPost`—a direct text entry on a feed.
- `TrackedChange`—a change or group of changes to a tracked field.
- `UserStatus`—automatically generated when a user adds a post. Deprecated.

The following values appear in the `Type` picklist for all feed objects but apply only to CaseFeed:

- `AttachArticleEvent`—generated event when a user attaches an article to a case.
- `CallLogPost`—generated event when a user logs a call for a case through the user interface. CTI calls also generate this event.
- `CaseCommentPost`—generated event when a user adds a case comment for a case object.
- `ChangeStatusPost`—generated event when a user changes the status of a case.
- `ChatTranscriptPost`—generated event when Chat transcript is saved to a case.
- `EmailMessageEvent`—generated event when an email related to a case object is sent or received.
- `FacebookPost`—generated when a Facebook post is created from a case. Deprecated.
- `MilestoneEvent`—generated when a case milestone is completed or reaches violation status.
- `SocialPost`—generated when a social post is created from a case.

Field	Details
	✏️ **Note:** If you set `Type` to `ContentPost`, also specify `ContentData` and `ContentFileName`.

Knowledge__kav

Provides access to the concrete object that represents a Knowledge article version. This object is available in API version 39.0 and later.

🛑 **Important:** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

✏️ **Note:** By default, the prefix for this object name is `Knowledge` and that is the value shown in this reference. However, this prefix can be modified by changing the **Object Name** for the Knowledge__kav object in Object Manager.

This object is derived from

Supported Calls

`create()`, `describeLayout()`, `describeSObjects()`, `query()`, `retrieve()`, `search()`, `update()`, `upsert()`

This object doesn't retrieve `<ActionOverrides>`.

Special Access Rules

Lightning Knowledge must be enabled in your org. A user must have the View Articles permission enabled. Salesforce Knowledge users, unlike customer and partner users, must also be granted the `Knowledge User` feature license.

Fields

Field	Details
ArchivedById	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** The ID of the user who archived the article.
ArchivedDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The date the article version was archived.

Field	Details
ArticleArchivedById	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** The ID of the user who archived the article.
ArticleArchivedDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The date the article was archived.
ArticleCaseAttachCount	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** The number of cases where this article is attached.
ArticleCreatedById	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** The ID of the user who created the article.
ArticleCreatedDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The date the article was created.
ArticleMasterLanguage	**Type** picklist **Properties** Filter, Group, Nillable, Restricted picklist, Sort

Field	Details
	Description The article's original language. Only accessible if your knowledge base supports multiple languages.
ArticleNumber	**Type** string **Properties** Autonumber, Defaulted on create, Filter, Sort **Description** The unique number automatically assigned to the article when it's created. You can't change the format or value for this field.
ArticleTotalViewCount	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** Total number of views for the article.
AssignedById	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** The ID of the user who assigned the article.
AssignedToId	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** The ID of the user assigned to the article.
AssignmentDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The date the article was assigned to a user.
AssignmentDueDate	**Type** dateTime

Field	Details
	Properties Filter, Nillable, Sort **Description** The due date when an article is assigned.
AssignmentNote	**Type** textarea **Properties** Filter, Nillable, Sort **Description** Notes to the assignee from the user who assigned the article.
ExternalRef	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The ID of the item being referenced on the external system. For example, the ID of a document on a Google Drive or a page on Confluence.
ExternalSourceId	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** Reference to the external Knowledge data source object.
ExternalUrl	**Type** url **Properties** Filter, Nillable, Sort **Description** The URL of the knowledge content referenced in an external system. For example, the ID of a document in Google Drive or a page in Confluence.
FirstPublishedDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The date when the article was first published.

Field	Details
IsExternalData	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether the data is external to the customer's knowledge base (`true`) or not (`false`).
IsLatestVersion	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether the article is the most current version. (`true`) or not (`false`). This field can be `true` on the online or published version, a draft version in the primary language, a draft version in a translation, and the latest archived version. However, you can't filter by (PublishState='Online') and (IsLatestVersion=false) because the online version is also the latest version. This field is available in API version 24.0 and later.
IsMasterLanguage	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether the article has one or more translations associated with it (`true`) or not (`false`).
IsOutOfDate	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether the source article has been updated since this translated version was created (`true`) or not (`false`).
IsVisibleInApp	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Required. Indicates whether the article is visible in the Articles tab (`true`) or not (`false`).

Field	Details
IsVisibleInCsp	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** Required. Indicates whether the article is visible in the Customer Portal (`true`) or not (`false`).
IsVisibleInPkb	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** Required. Indicates whether the article is visible in the public knowledge base (`true`) or not (`false`).
IsVisibleInPrm	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** Required. Indicates whether the article is visible in the partner portal (`true`) or not (`false`).
KnowledgeArticleId	**Type** reference **Properties** Filter, Group, Sort **Description** The ID of the article independent from its version. The value for this field is retrieved from the `Id` field of the KnowledgeArticle object.
Language	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort **Description** The language that the article is written in, such as `French` or `Chinese` `(Traditional)`. Querying or searching articles in SOSL require that you specify the `Language` field in the WHERE clause. The language must be the same for all article types.

Field	Details
	Before API version 47.0, you must include the `Language` field to filter queries on Knowledge article versions. In API version 47.0 and later, you can filter queries on Knowledge article versions with or without `Language` depending on what you are querying.
LastPublishedDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The date when the article was last published.
MasterVersionId	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** ID of the source article, if the article is the translation of a source article. Only accessible if your knowledge base supports multiple languages.
MigratedToFromArticleVersion	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The ID for the corresponding pre- or post-migration article version. Contains values only in orgs that migrate from Classic to Lightning Knowledge. Available in API version 43.0 and later.
NextReviewDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The date when the article must next be reviewed for accuracy. Available in API version 58.0 and later.
OwnerId	**Type** reference **Properties** Filter, Group, Sort **Description** The ID of the article's owner.

Field	Details
PublishStatus	**Type** picklist **Properties** Defaulted on create, Filter, Group, Restricted picklist, Sort **Description** The publication status for the article: • `Draft`: any draft articles. • `Online`: articles published in Salesforce Knowledge. • `Archived`: archived articles. A user must have the "Manage Articles" permission enabled to use `Online`. Article queries and searches in SOQL or SOSL require that you specify either the `PublishStatus` or the `Id` field in the WHERE clause. You can search for only one publication status per article type in a single SOSL query. When searching for articles with a `PublishStatus` of `Archived`, also check that `IsLatestVersion` equals `false` in your WHERE clause.
RecordTypeId	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** Indicates the API Name that describes the type of article. Use the record type to determine the article structure and other settings for different types of content.
SourceId	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** ID of the source from which the article was created (Case or Reply). This field is only accessible from the API and isn't visible in the Salesforce UI.
Summary	**Type** textarea **Properties** Create, Filter, Nillable, Sort, Update **Description** Summary of the article. Maximum size is 1000 characters.
Title	**Type** string

Field	Details
	Properties
	Create, Defaulted on create, Filter, Group, idLookup, Sort, Update
	Description
	Required. Article's title. Maximum size is 255 characters.
TranslationCompletedDate	**Type**
	dateTime
	Properties
	Filter, Nillable, Sort
	Description
	Date and time when the article was last translated. Only accessible if your knowledge base supports multiple languages.
TranslationExportedDate	**Type**
	dateTime
	Properties
	Filter, Nillable, Sort
	Description
	Date and time when the article was last exported for translation. Only accessible if your knowledge base supports multiple languages.
TranslationImportedDate	**Type**
	dateTime
	Properties
	Filter, Nillable, Sort
	Description
	Date and time when the article was last imported for translation. Only accessible if your knowledge base supports multiple languages.
UrlName	**Type**
	string
	Properties
	Create, Filter, Group, idLookup, Sort, Update
	Description
	Required. Represents the article's URL. Can contain alphanumeric characters and hyphens but can't begin or end with a hyphen. Use a unique value regardless of context. (For example, a unique value allows you to get expected results when running an Apex test with `SeeAllData` set to `false`.) `UrlName` is case-sensitive and its maximum size is 255 characters.
ValidationStatus	**Type**
	picklist

Field	Details
	Properties Defaulted on create, Filter, Group
	Description Shows whether the content of the article has been validated. Possible values are `Validated` and `Not Validated`. The default value is `Not Validated`. This field is available in API version 24.0 or later.
VersionNumber	**Type** int
	Properties Group, Sort
	Description The number assigned to a version of an article. This field is available in API version 24.0 and later.

Knowledge__ka

Provides access to the concrete object that represents a Knowledge article, the parent object for article versions. This object is available in API version 39.0 and later.

📝 **Note:** By default, the prefix for this object name is `Knowledge` and that is the value shown in this reference. However, this prefix can be modified by changing the **Object Name** for the Knowledge__kav object in Object Manager.

This object is derived from KnowledgeArticle on page 11.

Supported Calls

`delete(), describeLayout(), describeSObjects(), query(), retrieve(), undelete()`

Special Access Rules

Lightning Knowledge must be enabled in your org. A user must have the View Articles permission enabled. Salesforce Knowledge users, unlike customer and partner users, must also be granted the `Knowledge User` feature license.

Fields

Field	Details
ArchivedById	**Type** reference
	Properties Filter, Group, Nillable, Sort

Field	Details
	Description The ID of the user who archived the article.
ArchivedDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The date when the article was archived.
ArticleNumber	**Type** string **Properties** Autonumber, Defaulted on create, Filter, idLookup, Sort **Description** The unique number automatically assigned to the article when it's created. You can't change the format or value for this field.
CaseAssociationCount	**Type** int **Properties** Filter, Group, Sort **Description** The number of cases attached to the article.
FirstPublishedDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The date when the article was first published.
LastPublishedDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The date when the article was last published.
LastReferencedDate	**Type** dateTime

Field	Details
	Properties Filter, Nillable, Sort **Description** The timestamp when the current user last accessed this record, a record related to this record, or a list view.
LastViewedDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The timestamp when the current user last viewed this record or list view. If this value is null, the user might have only accessed this record or list view (`LastReferencedDate`) but not viewed it.
MasterLanguage	**Type** picklist **Properties** Filter, Group, Restricted picklist, Sort **Description** The article's original language. Only accessible if your knowledge base supports multiple languages.
MigratedToFromArticle	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The ID for the corresponding pre- or post-migration article. Contains values only in orgs that migrate from Knowledge in Salesforce Classic to Lightning Knowledge. This field is available in API version 45.0 and later.
TotalViewCount	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** Total number of views for this article. This field is available in API version 39.0 and later.

KnowledgeArticleVersionHistory

Enables read-only access to the full history of an article. This object is available in API version 25.0 and later.

Knowledge__VersionHistory is derived from this object. To access this derived object, turn on field history tracking for Knowledge objects.

Supported Calls

`describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`

You can also enable `delete()` in API version 42.0 and later. See Enable delete of Field History and Field History Archive.

Special Access Rules

Knowledge must be enabled in your org. This object respects field, entity, and record-level security. You must have at least "Read" permission on the article type or the field to access its history. For data category security, Salesforce determines access based on the categorization of the online version of an article. If there's no online version, then security is applied based on the archived version, followed by the security of the draft version.

Fields

Field Name	Details
DataType	**Type** picklist **Properties** Filter, Group, Restricted picklist, Sort **Description** The type of data that is tracked in the history table. This field is available in API version 50.0 and later.
EventType	**Type** picklist **Properties** Filter, Group, Restricted picklist, Sort **Description** The type of event that is tracked in the history table.
FieldName	**Type** picklist **Properties** Filter, Group, Nillable, Restricted picklist, Sort **Description** Name of the tracked field.
Language	**Type** picklist **Properties** Filter, Group, Restricted picklist, Sort

Field Name	Details
	Description
	The language that the article is written in, such as `French` or `Chinese (Traditional)`. Querying or searching articles in SOSL requires that you specify the `Language` field in the WHERE clause. The language must be the same for all article types.
`NewValue`	**Type**
	anyType
	Properties
	Nillable, Sort
	Description
	The new value of the field that was changed.
`OldValue`	**Type**
	anyType
	Properties
	Nillable, Sort
	Description
	The most recent value of the field before it was changed.
`ParentId`	**Type**
	reference
	Properties
	Filter, Group, Sort
	Description
	The ID of the article.
`ParentSobjectType`	**Type**
	picklist
	Properties
	Filter, Group, Restricted picklist, Sort
	Description
	The type of object that contains the field.
`VersionId`	**Type**
	reference
	Properties
	Filter, Group, Nillable, Sort
	Description
	The ID assigned to a version of the article.

Field Name	Details
	This is a polymorphic relationship field.
VersionNumber	**Type** int **Properties** Filter, Group, Sort **Description** The number assigned to a version of an article. This field is available in API version 24.0 and later.

Usage

Use this object to query events in the history of an article. For example, you can retrieve the number of edits a particular user has made to an article, how many times the article has been published, and so on.

KnowledgeArticleViewStat

Provides certain statistics related to the number of views for the specified article across all article types. The view count statistics are for published and archived articles only. View counts for draft articles aren't tracked. This object is read-only and available in API version 20.0 and later.

Knowledge__ViewStat is derived from this object.

Supported Calls

`describeSObjects()`, `query()`, `retrieve()`

Special Access Rules

Knowledge must be enabled in your org. Users must have access to the published and archived versions of an article to retrieve its views. For more information on published and archived article versions, see the `PublishStatus` field in KnowledgeArticleVersion.

Fields

Field Name	Details
Channel	**Type** picklist **Properties** Filter, Group, Restricted picklist, Sort **Description** The channel where the article is viewed: • `AllChannels` for article views across all channels.

Field Name	Details
	• `App` for the internal Salesforce Knowledge application. • `Pkb` for article views in the public knowledge base. • `Csp` for Customer Portal. • `Prm` for article view in partner portal.
NormalizedScore	**Type** double **Properties** Filter, Nillable, Sort **Description** Article's weighted views in the selected channel. The article with most views has a score of 100. Other article views are then calculated relative to this highest view score. For example, if the best read article has 2000 views and another has 1000. The first one gets a score of 100 while the second gets 50.
ParentId	**Type** reference **Properties** Filter, Group, Sort **Description** ID of the viewed article. This corresponds to a KnowledgeArticle record.
ViewCount	**Type** int **Properties** Filter, Group, Sort **Description** The number of unique views a published or archived article has received in the selected channel. An article with a high number of views may not always have a high normalized score. The normalized score for an article is calculated based on views over time, with more recent views earning a higher score. This field is available in API version 27.0 and later.

Usage

Use this object to query or retrieve certain statistics for article views.

Alternatively, client applications can use the article type `API Name` followed by `__ViewStat` to query or retrieve most viewed articles from a specific article type.

SOQL Samples

The following SOQL clause uses KnowledgeArticleViewStat to query all the article views in Salesforce Knowledge and return the related articles:

```
SELECT Id, NormalizedScore, Parent.Id
        FROM KnowledgeArticleViewStat where Channel = 'App'
        ORDER BY NormalizedScore
```

Use the following clause to restrict your query to Offer articles for the `Offer` article type:

```
SELECT Id, NormalizedScore, Parent.Id
        FROM Offer__ViewStat where Channel = 'App'
        ORDER BY NormalizedScore
```

KnowledgeArticleVoteStat

Provides the weighted rating for the specified article on a scale of 1 to 5 across all article types. This object is read-only and available in API version 20.0 and later.

Knowledge__VoteStat is derived from this object.

Supported Calls

`describeSObjects(), query(), retrieve()`

Special Access Rules

Knowledge must be enabled in your org. Users must have access to the published version of an article to retrieve its votes. For more information on published article version, see the `PublishStatus` field in KnowledgeArticleVersion

Fields

Field Name	Details
Channel	**Type** picklist **Properties** Filter, Group, Restricted picklist, Sort **Description** The channel where the article is rated: • `AllChannels` for article views across all channels. • `App` for the internal Salesforce Knowledge application. • `Pkb` for article views in public knowledge base. • `Csp` for Customer Portal. • `Prm` for article view in partner portal.

Field Name	Details
NormalizedScore	**Type** double **Properties** Filter, Nillable, Sort **Description** Article's weighted score on a scale of 1 to 5. A higher score means more votes. Articles without recent votes trend towards an average rating of three stars.
ParentId	**Type** reference **Properties** Filter, Group, Sort **Description** The rated article. This corresponds to a KnowledgeArticle record.

Usage

Use this object to query or retrieve the rating for an article.

Alternatively, client applications can use the article type `API Name` followed by `__VoteStat` to query or retrieve the rating for an article for a specific article type.

SOQL Samples

See KnowledgeArticleViewStat.

CaseArticle

Represents the association between a Case and a KnowledgeArticle. This object is available in API version 20.0 and later.

Supported Calls

`create(), delete(), describeSObjects(), query(), retrieve()`

Special Access Rules

Access to this object is controlled by the parent Case and KnowledgeArticle. However, when querying, access is only controlled by the parent Case.

Customer Portal users can't access this object.

Fields

Field	Details
ArticleLanguage	**Type** picklist **Properties** Filter, Restricted picklist **Description** The language of the article associated with the case.
ArticleVersionNumber	**Type** int **Properties** Create, Group, Nillable **Description** The number assigned to a version of an article. This field is available in API version 24.0 and later.
CaseId	**Type** reference **Properties** Create, Filter, Group, Sort **Description** ID of the Case associated with the KnowledgeArticle.
IsSharedByEmail	**Type** int **Properties** Create, Group, Nillable **Description** Indicates that the article has been shared with the customer through an email.
KnowledgeArticleId	**Type** reference **Properties** Create, Filter, Group, Sort **Description** ID of the KnowledgeArticle associated with the Case.

Usage

This object represents the association of a knowledge article with a Case. An article is associated with a case when it's relevant to a specific issue, when it helps an agent solve the case, or when the agent sends the article to a customer.

You can use this object to include case-article associations in Apex and Visualforce.

You can't update this object via the API. If you attempt to create a record that matches an existing record, the create request simply returns the existing record.

LinkedArticle

Represents a knowledge article that is attached to a work order, work order line item, or work type. This object is available in API version 37.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `undelete()`, `update()`, `upsert()`

Special Access Rules

Knowledge must be enabled in your org. Field Service must be enabled. Only users that have access to the Knowledge article and the parent record linked to it can access this object.

In Knowledge in Salesforce Classic, only Field Service objects such as Work Order, Work Type, and Work Order Line Item are supported for linked articles. In Lightning Knowledge, other social objects such as Chat, Messaging, Voice Call, and Social Post are supported for linked articles.

To call `update()` to attach or detach articles, enable the Read user permission on the Knowledge object and the Edit user permission on the object whose article you update. Available in API version 58.0 and later.

Fields

Field Name	Details
CurrencyIsoCode	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** Available only for orgs with the multicurrency feature enabled. Contains the ISO code for any currency allowed by the organization.
KnowledgeArticleId	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort

Field Name	Details
	Description The ID of the Knowledge article attached to the record. The label in the user interface is Knowledge Article ID.
KnowledgeArticleVersionId	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort **Description** The version of the Knowledge article attached to the record. This field lists the title of the attached version and links to the version. The label in the user interface is Article Version. When you attach an article to a work order, that version of the article stays associated with the work order, even if later versions are published. If needed, you can detach and reattach an article to a work order to link the latest version.
LinkedEntityId	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort **Description** The ID of the record that the Knowledge article is attached to. The label in the user interface is Linked Record ID.
Name	**Type** string **Properties** Create, Filter, Group, idLookup, Sort, Update **Description** The title of the article. The label in the user interface is Article Title.
RecordTypeId	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The ID of the article's record type, if used. This field is only available for Lightning Knowledge.
Type	**Type** string

Field Name	Details
	Properties
	Filter, Group, Nillable, Sort
	Description
	(Read only) The type of record that the Knowledge article is attached to. For example, work order. The label in the user interface is Linked Object Type.

Usage

Admins can customize linked articles' page layouts, fields, validation rules, and more from the Linked Articles page in Setup.

Associated Objects

This object has the following associated objects. Unless noted, they are available in the same API version as this object.

LinkedArticleChangeEvent (API version 62.0)

Change events are available for the object.

LinkedArticleFeed

Feed tracking is available for the object.

LinkedArticleHistory

History is available for tracked fields of the object.

RecentlyViewed

Represents records or list views that the current user has recently viewed or referenced (by viewing a related record). List views are available in API version 29.0 and later.

Supported Calls

`describeSObjects(), query(), update()`

Special Usage Rules

The RecentlyViewed object doesn't support the Event, Task, Report, KnowledgeArticle, and Article objects.

The RecentlyViewed object supports only certain objects, and supports list views only for those supported objects. Supported objects have the fields `LastReferencedDate` and `LastViewedDate`.

> **Note:** RecentlyViewed records for users who are members of several communities can't be retrieved automatically into a map via Apex. This is because records of a user with different networks can result in duplicate IDs that maps don't support.

Fields

Field	Details
Alias	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The alias on the record.
Email	**Type** email **Properties** Filter, Group, Nillable, Sort **Description** The email address on the record.
FirstName	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The first name on the record. If the recently viewed record is a user, the value is the user's first name.
Id	**Type** ID **Properties** Defaulted on create, Filter, Group, Sort **Description** The ID of the recently viewed record or list view.
IsActive	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether the recently viewed record is an active user (`true`) or not (`false`). This field contains a value only if the recently viewed record is a user.
LastName	**Type** string

Field	Details
	Properties Filter, Group, Nillable, Sort **Description** The last name on the record.
LastReferencedDate	**Type** dateTime **Properties** Filter, Nillable, Sort, Update **Description** The timestamp when the current user last interacted with this record, directly or indirectly. Some sample scenarios are: • Viewing or opening a record. • Selecting a record in a lookup search or related list search.
LastViewedDate	**Type** dateTime **Properties** Filter, Nillable, Sort, Update **Description** The timestamp when the current user last viewed this record or list view. If this value is null, it's possible that the user only accessed this record or list view (`LastReferencedDate`), but not viewed it.
Name	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The name on the recently viewed record or list view. If the recently viewed record is a user, contact, or lead, the value is a concatenation of the `firstname` and `lastname` field values.
NetworkId	**Type** reference **Properties** Filter,Group, Nillable, Sort **Description** ID of the Experience Cloud site that this group is part of. This field is available only if digital experiences is enabled in your org. You can add a `NetworkId` only when creating a group. You can't change or add a `NetworkId` for an existing group. This field is available in API version 27.0 and later.

Field	Details
Phone	**Type** phone **Properties** Filter, Group, Nillable, Sort **Description** The phone number on the record.
ProfileId	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** If the recently viewed record is a user, this value is the user's profile ID. This field is a relationship field. **Relationship Name** Profile **Relationship Type** Lookup **Refers To** Profile
Title	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** If the recently viewed record is a user, this value is the title of the user; for example CFO or CEO.
Type	**Type** picklist **Properties** Filter, Group, Nillable, Restricted picklist, Sort **Description** The object type for this recently viewed record or list view. Valid values include any standard or custom objects that RecentlyViewed supports.
UserRoleId	**Type** reference **Properties** Filter, Group, Nillable, Sort

Field	Details
	Description
	The ID of the user role associated with this object.
	This field is a relationship field.
	Relationship Name
	UserRole
	Relationship Type
	Lookup
	Refers To
	UserRole

Usage

This object provides a heterogeneous list of different object types. The list consists of recently viewed records, records that were recently referenced (a related record was viewed), or recently viewed list views. A record is considered viewed when the user sees the record details, but not when the user sees the record in a list with other records. Use this object to programmatically construct a list of recently viewed items specific to the current user. For example, use this object on a custom user interface or for search auto-complete options. You can also retrieve a filtered list of records by object type (Type). The RecentlyViewed data is periodically truncated down to 200 records and 200 list views. RecentlyViewed data is retained for 90 days, after which it's removed on a periodic basis.

Use this query in your code to retrieve a list of all the records and list views that were recently viewed. The results are ordered from most to least recent.

```
SELECT Id, Name
FROM RecentlyViewed
WHERE LastViewedDate !=null
ORDER BY LastViewedDate DESC
```

Use this query to retrieve data that was either viewed or referenced, but only for a limited set of objects.

```
SELECT Id, Name
FROM RecentlyViewed
WHERE Type IN ('Account', 'Contact', 'Plan__c')
ORDER BY LastViewedDate DESC
```

This query retrieves a list of all recently viewed contacts with contact-specific fields, such as the contact's account name, and the custom website field. Records are ordered from most to least recent.

```
SELECT Account.Name, Title, Email, Phone, Website__c
FROM Contact
WHERE LastViewedDate != NULL
ORDER BY LastViewedDate DESC
```

SearchPromotionRule

Represents a promoted search term, which is one or more keywords that you associate with a Salesforce Knowledge article. When a user's search query includes these keywords, the associated article is returned first in search results. This object is available in API version 31.0 and later.

Supported Calls

`create(),delete(),describeLayout(),describeSObjects(),getDeleted(),getUpdated(),query(),`
`retrieve(),undelete(),update(),upsert()`

Special Access Rules

A user must have the "Manage Promoted Search Terms" permission.

Fields

Field Name	Details
`PromotedEntityId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The ID of the KnowledgeArticleVersion that the promoted search term is associated with. The article must be in published status.
`Query`	**Type** string **Properties** Create, Filter, Group, Sort, Update **Description** The text of the promoted search term. Maximum length: 100 characters. You can associate the same promoted search term with multiple articles. If the user's search matches the promoted term, all associated articles are promoted in search results, ordered by relevancy. For best results, create promoted search terms selectively and limit the number of articles that are promoted per term.

Usage

Use this object to optimize article search results in Salesforce Knowledge.

TopicAssignment

Represents the assignment of a topic to a specific feed item, record, or file. This object is available in API version 28.0 and later.

Administrators must enable topics for objects before users can add topics to records of that object type. Topics for most objects are available in API version 30.0 and later. Topics for ContentDocument are available in API version 37.0 and later.

Supported Calls

`create(),describeSObjects(),delete(),getDeleted(),getUpdate(),query(),retrieve()`

Fields

Field Name	Details
EntityId	**Type** reference **Properties** Create, Filter, Group, Sort **Description** Identifier of the feed item, record, or file. This is a polymorphic relationship field. **Relationship Name** Entity **Relationship Type** Lookup **Refers To** Account, Asset, Campaign, Case, Contact, ContentDocument, Contract, Event, FeedItem, Lead, Opportunity, Order, ProductItem, ProductItemTransaction, ProductRequest, ProductRequestLineItem, ProductRequired, ProductTransfer, ResourceAbsence, ResourcePreference, ReturnOrder, ReturnOrderLineItem, ServiceAppointment, ServiceResource, ServiceResourceSkill, ServiceTerritory, ServiceTerritoryMember, Shift, Shipment, Solution, Task, WorkOrder, WorkOrderLineItem
EntityKeyPrefix	**Type** string **Properties** Filter, Group, idLookup, Sort **Description** The first three digits of the `EntityID` field, which identify the object type (account, opportunity, etc). This read-only field is available in API version 32.0 and later. Interface label is "Record Key Prefix," which appears only in reports.
EntityType	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The standard name for the object type (account, opportunity, etc). This read-only field is available in API version 33.0 and later. 📝 **Note:** Querying topic assignments for the ManagedContentVersion entity type isn't supported. Interface label is "Object Type," which appears only in reports.

Field Name	Details
	💡 **Tip:** In most cases, you should use this field rather than `EntityKeyPrefix`, which exists primarily to support older reports.
NetworkId	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort **Description** Identifier of the community to which the TopicAssignment belongs. This field is available only if digital experiences is enabled in your org.
TopicId	**Type** reference **Properties** Create, Filter, Group, Sort **Description** Identifier of the topic. This is a relationship field. **Relationship Name** Topic **Relationship Type** Lookup **Refers To** Topic

Usage

Use this object to query the assignments of topics to feed items, records, or files. To assign or remove topics, you must have the "Assign Topics" permission.

In SOQL `SELECT` syntax, this object supports nested semi-joins, allowing queries on Knowledge articles assigned to specific topics. For example:

```
SELECT parentId FROM KnowledgeArticleViewStat
    WHERE parentId in (SELECT KnowledgeArticleId FROM KnowledgeArticleVersion
    WHERE publishStatus = 'Online' AND language = 'en_US'
    AND  Id in (select EntityId from TopicAssignment where TopicId ='0T0xx0000000xxx'))
```

There is no SOQL limit if the logged-in user has the "View All Data" permission. If they do have that permission, do one of the following:

- Specify a LIMIT clause of 1,100 records or fewer.
- Filter on `Id` or `Entity` when using a `WHERE` clause with "=".

🚫 **Important:** Deleting this object's records removes all its data. This action is irreversible.

> **Note:** When you create a report type on the TopicAssignment object, all queries are generated in SQL, which does not enforce the 1,100 record limit clause.

Knowledge SOAP API Calls

SOAP API calls for working with Salesforce Knowledge.

IN THIS SECTION:

describeKnowledge()

Retrieves the Knowledge language settings in the organization.

describeDataCategoryGroups()

Retrieves available category groups for objects specified in the request.

describeDataCategoryGroupStructures()

Retrieves available category groups along with their data category structure for objects specified in the request.

search()

Executes a text search in your organization's data.

`describeKnowledge()`

Retrieves the Knowledge language settings in the organization.

Syntax

```
KnowledgeSettings result = _connection.describeKnowledgeSettings();
```

Usage

Use this call to describe the existing Knowledge language settings, including the default Knowledge language, supported languages, and a list of Knowledge language information. You can also use KnowledgeSettings in the Metadata API to obtain similar information.

Sample Code—Java

This sample shows how to retrieve the Knowledge language settings. It returns the default Knowledge language, a list of Knowledge supported language, including the language code and whether it's an active Knowledge language.

```
public void describeKnowledgeSettingsSample() {
try {

// Make the describe call for KnowledgeSettings
KnowledgeSettings result = connection.describeKnowledgeSettings();

// Get the properties of KnowledgeSettings
System.out.println("Knowledge default language: " + result.getDefaultLanguage());
for (KnowledgeLanguageItem lang : result.getLanguages()) {
System.out.println("Language: " + lang.getName());
System.out.println("Active: " + lang.isActive());
```

```
}
} catch (ConnectionException ex) {
ex.printStackTrace();
}
}
```

Sample Code—C#

This sample shows how to retrieve the Knowledge language settings. It returns the default Knowledge language, a list of Knowledge supported language, including the language code and whether it's an active Knowledge language.

```
public void describeKnowledgeSettingsSample() {
try {

// Make the describe call for KnowledgeSettings
KnowledgeSettings result = connection.describeKnowledgeSettings();

// Get the properties of KnowledgeSettings
Console.WriteLine("Knowledge default language: " + result.getDefaultLanguage());
for (KnowledgeLanguageItem lang : result.getLanguages()) {
Console.WriteLine("Language: " + lang.getName());
Console.WriteLine("Active: " + lang.isActive());
}
} catch (SoapException ex) {
ex.printStackTrace();
}
}
```

Response

KnowledgeSettings

describeDataCategoryGroups()

Retrieves available category groups for objects specified in the request.

Syntax

```
DescribeDataCategoryGroupResult[] = connection.describeDataCategoryGroups()(string[]
sObjectTypes);
```

Usage

Use this call to describe the available category groups for the objects specified in the request. This call can be used with the `describeDataCategoryGroupStructures()` call to describe all the categories available for a specific object. For additional information about data categories, see "Work with Data Categories" in the Salesforce online help.

Sample Code—Java

This sample shows how to retrieve the data category groups associated with:

- Salesforce Knowledge articles
- Questions from the Answers feature

It returns the name, label and description of a category group and the name of the associated `sobject` (article or question). It also returns the number of data categories in the data category group.

```
public void describeDataCategoryGroupsSample() {
                try {
                // Make the describe call for data category groups
                DescribeDataCategoryGroupResult[] results =
                connection.describeDataCategoryGroups(new String[] {
                "KnowledgeArticleVersion", "Question"});

                // Get the properties of each data category group
                for (int i = 0; i < results.length; i++) {
                System.out.println("sObject: " +
                results[i].getSobject());
                System.out.println("Group name: " +
                results[i].getName());
                System.out.println("Group label: " +
                results[i].getLabel());
                System.out.println("Group description: " +
                (results[i].getDescription()==null? "" :
                results[i].getDescription()));
                System.out.println("Number of categories: " +
                results[i].getCategoryCount());
                }
                } catch (ConnectionException ce) {
                ce.printStackTrace();
                }
                }
```

Sample Code—C#

This sample shows how to retrieve the data category groups associated with:

- Salesforce Knowledge articles
- Questions from the Answers feature

It returns the name, label and description of a category group and the name of the associated `sobject` (article or question). It also returns the number of data categories in the data category group.

```
public void describeDataCategoryGroups() {
                try {
                // Make the describe call for data category groups
                DescribeDataCategoryGroupResult[] results =
                binding.describeDataCategoryGroups(new String[] {
                "KnowledgeArticleVersion", "Question"});

                // Get the properties of each data category group
                for (int i = 0; i < results.Length; i++) {
                Console.WriteLine("sObject: " +
                results[i].sobject);
                Console.WriteLine("Group name: " +
                results[i].name);
```

```
                        Console.WriteLine("Group label: " +
                        results[i].label);
                        Console.WriteLine("Group description: " +
                        (results[i].description==null? "" :
                        results[i].description));
                        Console.WriteLine("Number of categories: " +
                        results[i].categoryCount);
                        }
                        } catch (SoapException e) {
                        Console.WriteLine("An unexpected error has occurred: " +
                        e.Message + "\n" + e.StackTrace);
                        }
                        }
```

Arguments

Name	Type	Description
sObjectTypes	string[]	The specified value can be: • `KnowledgeArticleVersion`—to retrieve category groups associated with article types. • `Question`—to retrieve category groups associated with questions. For additional information about articles and questions, see "Work with Articles and Translations" in the Salesforce online help.

Response

DescribeDataCategoryGroupResult

Faults

InvalidSObjectFault

UnexpectedErrorFault

IN THIS SECTION:

DescribeDataCategoryGroupResult

The describeDataCategoryGroups() call returns a DescribeDataCategoryGroupResult object containing the list of the category groups associated with the specified objects.

DescribeDataCategoryGroupResult

The describeDataCategoryGroups() call returns a DescribeDataCategoryGroupResult object containing the list of the category groups associated with the specified objects.

Name	Type	Description
categoryCount	int	The number of visible data categories in the data category group.
description	string	The description of the data category group.
label	string	Label for the data category group in the Salesforce user interface.
name	string	The unique name used for API access to the data category group .
sobject	string	The object associated with the data category group.

describeDataCategoryGroupStructures()

Retrieves available category groups along with their data category structure for objects specified in the request.

Syntax

```
describeDataCategoryGroupStructures()[] = connection.
            describeDataCategoryGroupStructures()(DataCategoryGroupSObjectTypePair[]
 pairs, boolean topCategoriesOnly)
```

Usage

Use this call to return the visible data category structure for the given object category group pairs. First use
describeDataCategoryGroups() to find the available category groups for the objects specified. From the returned list, choose
the object category group pairs to pass as the input in describeDataCategoryGroupStructures(). This call returns all
the visible categories and data category structure as output. For additional information about data categories and data category visibility,
see "Work with Data Categories" and "Data Category Visibility" in Salesforce Help .

Sample Code—Java

This sample shows how to use sObject and data category group pairs to retrieve data categories for each pair. It calls
describeDataCategoryGroupStructures() with two pairs, KnowledgeArticleVersion/Regions and Question/Regions,
and iterates through the results of this call. It gets the top categories for each result, which is "All", and then gets the first-level child
categories. The sample requires that you set up a data category group called *Regions* with some child categories and associate it
with a knowledge article and questions. Alternatively, you can replace the data category group name in the sample if you want to use
an existing data category group in your org that has a different name.

```
public void describeDataCateogryGroupStructuresSample() {
            try {
            // Create the data category pairs
            DataCategoryGroupSobjectTypePair pair1 =
            new DataCategoryGroupSobjectTypePair();
            DataCategoryGroupSobjectTypePair pair2 =
            new DataCategoryGroupSobjectTypePair();
            pair1.setSobject("KnowledgeArticleVersion");
            pair1.setDataCategoryGroupName("Regions");
            pair2.setSobject("Question");
```

```
pair2.setDataCategoryGroupName("Regions");

DataCategoryGroupSobjectTypePair[] pairs =
new DataCategoryGroupSobjectTypePair[] {
pair1,
pair2
};

// Get the list of top level categories using the describe call
DescribeDataCategoryGroupStructureResult[] results =
connection.describeDataCategoryGroupStructures(
pairs,
false
);

// Iterate through each result and get some properties
// including top categories and child categories
for (int i = 0; i < results.length; i++) {
DescribeDataCategoryGroupStructureResult result =
results[i];
String sObject = result.getSobject();
System.out.println("sObject: " + sObject);
System.out.println("Group name: " + result.getName());
System.out.println("Group label: " + result.getLabel());
System.out.println("Group description: " +
result.getDescription());

// Get the top-level categories
DataCategory[] topCategories = result.getTopCategories();

// Iterate through the top level categories and retrieve
// some information
for (int j = 0; j < topCategories.length; j++) {
DataCategory topCategory = topCategories[j];
System.out.println("Category name: " +
topCategory.getName());
System.out.println("Category label: " +
topCategory.getLabel());
DataCategory [] childCategories =
topCategory.getChildCategories();
System.out.println("Child categories: ");
for (int k = 0; k < childCategories.length; k++) {
System.out.println("\t" + k + ". Category name: " +
childCategories[k].getName());
System.out.println("\t" + k + ". Category label: " +
childCategories[k].getLabel());
}
}
}
} catch (ConnectionException ce) {
ce.printStackTrace();
}
}
```

Sample Code—C#

This sample shows how to use sObject and data category group pairs to retrieve data categories for each pair. It calls `describeDataCategoryGroupStructures()` with two pairs, KnowledgeArticleVersion/Regions and Question/Regions, and iterates through the results of this call. It gets the top categories for each result, which is "All", and then gets the first-level child categories. The sample requires that you set up a data category group called *Regions* with some child categories and associate it with a knowledge article and questions. Alternatively, you can replace the data category group name in the sample if you want to use an existing data category group in your org that has a different name.

```csharp
public void describeDataCateogryGroupStructuresSample() {
                try {
                // Create the data category pairs
                DataCategoryGroupSobjectTypePair pair1 =
                new DataCategoryGroupSobjectTypePair();
                DataCategoryGroupSobjectTypePair pair2 =
                new DataCategoryGroupSobjectTypePair();
                pair1.sobject = "KnowledgeArticleVersion";
                //pair1.setDataCategoryGroupName("Regions");
                pair1.dataCategoryGroupName = "KBArticleCategories";
                pair2.sobject = "Question";
                //pair2.setDataCategoryGroupName("Regions");
                pair2.dataCategoryGroupName = "KBArticleCategories";

                DataCategoryGroupSobjectTypePair[] pairs =
                new DataCategoryGroupSobjectTypePair[] {
                pair1,
                pair2
                };

                // Get the list of top level categories using the describe call
                DescribeDataCategoryGroupStructureResult[] results =
                binding.describeDataCategoryGroupStructures(
                pairs,
                false
                );

                // Iterate through each result and get some properties
                // including top categories and child categories
                for (int i = 0; i < results.Length; i++) {
                DescribeDataCategoryGroupStructureResult result =
                results[i];
                String sObject = result.sobject;
                Console.WriteLine("sObject: " + sObject);
                Console.WriteLine("Group name: " + result.name);
                Console.WriteLine("Group label: " + result.label);
                Console.WriteLine("Group description: " +
                result.description);

                // Get the top-level categories
                DataCategory[] topCategories = result.topCategories;

                // Iterate through the top level categories and retrieve
                // some information
                for (int j = 0; j < topCategories.Length; j++) {
```

```
                    DataCategory topCategory = topCategories[j];
                    Console.WriteLine("Category name: " +
                    topCategory.name);
                    Console.WriteLine("Category label: " +
                    topCategory.label);
                    DataCategory [] childCategories =
                    topCategory.childCategories;
                    Console.WriteLine("Child categories: ");
                    for (int k = 0; k < childCategories.Length; k++) {
                    Console.WriteLine("\t" + k + ". Category name: " +
                    childCategories[k].name);
                    Console.WriteLine("\t" + k + ". Category label: " +
                    childCategories[k].label);
                    }
                    }
                    }
                    }
                    catch (SoapException e)
                    {
                    Console.WriteLine("An unexpected error has occurred: " +
                    e.Message + "\n" + e.StackTrace);
                    }
                    }
```

Arguments

Name	Type	Description
pairs	DataCategoryGroupSObjectTypePair[]	Specifies a category group and an object to query. Visible data categories are retrieved for that object.
topCategoriesOnly	boolean	Indicates whether the call returns only the top (true) or all the categories (false) visible depending on the user's data category group visibility settings. For more information on data category group visibility, see Data Category Visibility in Salesforce Help.

DataCategoryGroupSObjectTypePair contains the following fields:

Name	Type	Description
dataCategoryGroupName	string	The unique name used for API access to the data category group.
sobject	string	The object associated with the data category group

Response

describeDataCategoryGroupStructures()

Faults

```
InvalidSObjectFault
UnexpectedErrorFault
```

IN THIS SECTION:

[describeDataCategoryGroupStructures()](#)
The describeDataCategoryGroupStructures() call returns an array of DescribeDataCategoryGroupStructureResult objects containing the category groups and categories associated with the specified objects.

describeDataCategoryGroupStructures()

The describeDataCategoryGroupStructures() call returns an array of DescribeDataCategoryGroupStructureResult objects containing the category groups and categories associated with the specified objects.

Name	Type	Description
description	string	The description of the data category group.
label	string	The label for the data category group in the Salesforce user interface.
name	string	The unique name used for API access to the data category group.
sobject	string	The object associated with the data category group.
topCategories	DataCategory[]	A list of top level categories visible depending on the user's data category group visibility settings. For more information on data category group visibility, see "Data Category Visibility" in the Salesforce online help.

DataCategory

Name	Type	Description
childDataCategories	DataCategory[]	A recursive list of visible sub categories in the data category.
label	string	The label for the data category in the Salesforce user interface.
name	string	The unique name used for API access to the data category.

search()

Executes a text search in your organization's data.

Syntax

```
SearchResult = connection.search(String searchString);
```

Usage

Use search() to search for records based on a search string. The search call supports searching custom objects. For an extensive discussion about the syntax and rules used for text searches, see the Salesforce SOQL and SOSL Reference Guide.

Certain objects cannot be searched via the API, such as Attachment objects. To search an object via the search() call, the object must be configured as searchable (isSearchable is true). To determine whether an object can be searched, your client application can invoke the describeSObjects() call on the object and inspect its searchable property.

Sample Code—Java

This sample makes the search() call by passing it a SOSL query, which returns contacts, leads, and accounts whose phone fields contain a specified value. Next, it gets the sObject records from the results and stores the records in arrays depending on the record type. Finally, it writes the fields of the returned contacts, leads, and accounts to the console.

```java
public void searchSample() {
        try {
        // Perform the search using the SOSL query.
        SearchResult sr = connection.search(
        "FIND {4159017000} IN Phone FIELDS RETURNING "
        + "Contact(Id, Phone, FirstName, LastName), "
        + "Lead(Id, Phone, FirstName, LastName), "
        + "Account(Id, Phone, Name)");

        // Get the records from the search results.
        SearchRecord[] records = sr.getSearchRecords();

        ArrayList<Contact> contacts = new ArrayList<Contact>();
        ArrayList<Lead> leads = new ArrayList<Lead>();
        ArrayList<Account> accounts = new ArrayList<Account>();

        // For each record returned, find out if it's a
        // contact, lead, or account and add it to the
        // appropriate array, then write the records
        // to the console.
        if (records.length > 0) {
        for (int i = 0; i < records.length; i++) {
        SObject record = records[i].getRecord();
        if (record instanceof Contact) {
        contacts.add((Contact) record);
        } else if (record instanceof Lead) {
        leads.add((Lead) record);
        } else if (record instanceof Account) {
        accounts.add((Account) record);
        }
        }

        System.out.println("Found " + contacts.size() + " contacts.");
        for (Contact c : contacts) {
        System.out.println(c.getId() + ", " + c.getFirstName() + ", "
        + c.getLastName() + ", " + c.getPhone());
        }
        System.out.println("Found " + leads.size() + " leads.");
        for (Lead d : leads) {
```

```
System.out.println(d.getId() + ", " + d.getFirstName() + ", "
+ d.getLastName() + ", " + d.getPhone());
}
System.out.println("Found " + accounts.size() + " accounts.");
for (Account a : accounts) {
System.out.println(a.getId() + ", " + a.getName() + ", "
+ a.getPhone());
}
} else {
System.out.println("No records were found for the search.");
}
} catch (Exception ce) {
ce.printStackTrace();
}
}
```

Sample Code—C#

This sample makes the `search()` call by passing it a SOSL query, which returns contacts, leads, and accounts whose phone fields contain a specified value. Next, it gets the sObject records from the results and stores the records in arrays depending on the record type. Finally, it writes the fields of the returned contacts, leads, and accounts to the console.

```
public void searchSample()
{
try
{
// Perform the search using the SOSL query.
SearchResult sr = binding.search(
"FIND {4159017000} IN Phone FIELDS RETURNING "
+ "Contact(Id, Phone, FirstName, LastName), "
+ "Lead(Id, Phone, FirstName, LastName), "
+ "Account(Id, Phone, Name)");

// Get the records from the search results.
SearchRecord[] records = sr.searchRecords;

List<Contact> contacts = new List<Contact>();
List<Lead> leads = new List<Lead>();
List<Account> accounts = new List<Account>();

// For each record returned, find out if it's a
// contact, lead, or account and add it to the
// appropriate array, then write the records
// to the console.
if (records.Length > 0)
{
for (int i = 0; i < records.Length; i++)
{
sObject record = records[i].record;
if (record is Contact)
{
contacts.Add((Contact)record);
}
else if (record is Lead)
```

```
                       {
                       leads.Add((Lead)record);
                       }
                       else if (record is Account)
                       {
                       accounts.Add((Account)record);
                       }
                       }

                       Console.WriteLine("Found " + contacts.Count + " contacts.");
                       foreach (Contact c in contacts)
                       {
                       Console.WriteLine(c.Id + ", " +
                       c.FirstName + ", " +
                       c.LastName + ", " +
                       c.Phone);
                       }
                       Console.WriteLine("Found " + leads.Count + " leads.");
                       foreach (Lead d in leads)
                       {
                       Console.WriteLine(d.Id + ", " +
                       d.FirstName + ", " +
                       d.LastName + ", " +
                       d.Phone);
                       }
                       Console.WriteLine("Found " + accounts.Count + " accounts.");
                       foreach (Account a in accounts)
                       {
                       Console.WriteLine(a.Id + ", " +
                       a.Name + ", " +
                       a.Phone);
                       }
                       }
                       else
                       {
                       Console.WriteLine("No records were found for the search.");
                       }
                       }
                       catch (SoapException e)
                       {
                       Console.WriteLine("An unexpected error has occurred: " +
                       e.Message + "\n" + e.StackTrace);
                       }
                       }
```

Arguments

Name	Type	Description
search	string	Search string that specifies the text expression to search for, the scope of fields to search, the list of objects and fields to retrieve, and the maximum number of records to return. For more information, see the Salesforce SOQL and SOSL Reference Guide.

Response

SearchResult

Fault

```
InvalidFieldFault

InvalidSObjectFault

MalformedSearchFault

UnexpectedErrorFault
```

IN THIS SECTION:

SearchResult

The `search()` call returns a SearchResult object.

SearchResult

The `search()` call returns a SearchResult object.

A SearchResult object has these fields.

Name	Type	Description
queryId	string	Unique identifier for the `SOSL` search.
searchRecords	SearchRecord[]	Array of SearchRecord objects, each of which contains an sObject.
searchResultsMetadata	SearchResultsMetadata	Metadata for SearchRecords.

SearchRecord

Represents an individual record returned from a search.

Name	Type	Description
record	sObject	The individual record returned by the search.
searchRecordMetadata	SearchRecordMetadata	Metadata for `searchRecords`.
snippet	SearchSnippet	On the search results page, shows terms that match the search string, highlighted within the surrounding text.

SearchRecordMetadata

Metadata for search results at the record level.

Name	Type	Description
searchPromoted	boolean	Indicates that an article has been promoted in search results. Admins define promoted search terms by adding promoted terms to knowledge

Name	Type	Description
		articles. Users who search for these keywords see the article first in search results. Available in API version 42.0 and later.
spellCorrected	boolean	Indicates that a record matches a spell-corrected search term. Appears in the response only when true.

SearchSnippet

Excerpts shown on search results pages for article, case, feed, and idea searches.

Name	Type	Description
text	string	The excerpt that contains the match for the search term.
wholeFields	WholeFields	The list of highlighted fields.

WholeFields

Contains the complete text of each field that contains highlighting for terms that match the search query. The highlighted terms are surrounded by `<mark>` tags.

Name	Type	Description
name	string	The name of the highlighted field.
value	string	The highlighted text.

SearchResultsMetadata

Global metadata for the search result.

Name	Type	Description
entityMetadata	EntitySearchMetadata	Search results metadata at the object level.

EntitySearchMetadata

Metadata for search results at the object level.

Name	Type	Description
fieldMetadata	FieldLevelSearchMetadata	Metadata for search results at the field level.
searchPromotedMetadata	EntitySearchPromotionMetadata	Metadata for search term promotion at the object level. Available in API version 42.0 and later.
spellCorrectionMetadata	EntitySpellCorrectionMetadata	Metadata for spelling correction at the object level.
entityName	string	Identifies the object.

FieldLevelSearchMetadata

Metadata for search results at the field level.

Name	Type	Description
name	string	The field name.
label	string	The field label.
type	string	The field type.

EntitySearchPromotionMetadata

Metadata for search term promotion at the object level. Appears in the response only when at least one article for an object is a promoted result. Available in API version 42.0 and later.

Name	Type	Description
promotedResultCount	int	Count of promoted article results at the object level.

EntitySpellCorrectionMetadata

Metadata for spelling correction at the object level. Appears in the response only when at least one record for an object matches a spell-corrected search term.

Name	Type	Description
correctedQuery	string	The spell-corrected search term.
hasNonCorrectedResults	boolean	If `true`, indicates that the user has access to at least one record that matches a search term that wasn't spell-corrected. Each object sometimes returns a different value.

CHAPTER 3   Salesforce Knowledge REST APIs

You can manage articles and get support information with REST APIs.

REST API uses the same underlying data model and standard objects as those in SOAP API. For generic REST API information see the REST API Developer Guide.

Invocable Actions for Lightning Knowledge

Actions for Lightning Knowledge can be invoked from a REST endpoint. Use these actions to manage your articles and article versions.

To learn more about how to use invocable actions, see the Actions Developer Guide.

IN THIS SECTION:

Knowledge Actions

Manage your Knowledge articles using invocable actions.

Knowledge Actions

Manage your Knowledge articles using invocable actions.

🛇 **Important:** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

The Assign and Publish actions are available in API version 44.0 and later. All the other actions are available in API version 45.0 and later.

Lightning Knowledge must be set up in your org. The user must have permissions to manage articles.

You can use multiple inputs to an invocable action. This technique is useful for actions that don't take lists, such as `restoreKnowledgeArticleVersion`.

Supported REST HTTP Methods

URIs

Archive Knowledge articles:

`/services/data/v`***XX.X***`/actions/standard/archiveKnowledgeArticles`

Assign Knowledge articles:

`/services/data/v`***XX.X***`/actions/standard/assignKnowledgeArticles`

Create draft from online Knowledge articles:

`/services/data/v`***XX.X***`/actions/standard/createDraftFromOnlineKnowledgeArticle`

Delete Knowledge articles:

`/services/data/v`***XX.X***`/actions/standard/deleteKnowledgeArticles`

Publish Knowledge articles:

`/services/data/v`***XX.X***`/actions/standard/publishKnowledgeArticles`

Restore Knowledge article version:

`/services/data/v`***XX.X***`/actions/standard/restoreKnowledgeArticleVersion`

Retrieve Smart Link URL:

`/services/data/v`***XX.X***`/actions/standard/getArticleSmartLinkUrl`

Submit Knowledge article for translation:

`/services/data/v`***XX.X***`/actions/standard/submitKnowledgeArticleForTranslation`

Formats

JSON, XML

HTTP Methods

GET, HEAD, POST

Authentication

`Authorization: Bearer `***`token`***

Other Information

Error Response Types on page 86

Archive Knowledge Articles

URI: `/services/data/v`***`XX.X`***`/actions/standard/archiveKnowledgeArticles`

Table 1: Inputs

Input	Details
`articleVersionIdList`	**Type** string **Description** Required. Comma-separated article version ID list.

Sample Input

The following code sample archives two articles:

```
{
  "inputs" : [
    {
      "articleVersionIdList" : [ "ka0RM00000004VeYAI", "ka0RM00000003doYAA" ]
    }
  ]
}
```

Sample Output

The following code sample illustrates a response after a successful request.

```
[ {
  "actionName" : "archiveKnowledgeArticles",
  "errors" : null,
  "isSuccess" : true,
  "outputValues" : {
    "ka0RM00000004Ve" : "Success",
    "ka0RM00000003do" : "Success"
  }
} ]
```

The following code sample illustrates a response with one success and one failure:

```
[ {
  "actionName" : "archiveKnowledgeArticles",
```

```
  "errors" : null,
  "isSuccess" : false,
  "outputValues" : {
    "ka0RM00000004Ve" : "You can't perform this action. Be sure the action is valid for
 the current state of the article, and that you have permission to perform it.",
    "ka0RM00000003do" : "Success"
  }
} ]
```

Assign Knowledge Articles

URI: `/services/data/v`**_XX.X_**`/actions/standard/assignKnowledgeArticles`

Table 2: Inputs

Input	Details
`articleVersionIdList`	**Type** string **Description** Required. Comma-separated article version ID list.
`assigneeId`	**Type** ID **Description** Required. ID of the assigned user.
`assignAction`	**Type** string **Description** Required. Assign action. Valid actions are: • ASSIGN_DRAFT_MASTER • ASSIGN_DRAFT_TRANSLATION
`dueDate`	**Type** string **Description** Optional. Assigned due date.
`instruction`	**Type** string **Description** Optional. Instructions for the assignee.
`sendEmailNotification`	**Type** boolean

Input	Details
	Description Optional. Indicates whether to send an email notification. Defaults to `false`.

Sample Input

The following code sample assigns two articles for translation:

```
{
  "inputs" : [
    {
      "articleVersionIdList" : [ "ka0RM00000004VeYAI", "ka0RM00000003doYAA" ]
      "assigneeId" : "005RM00000AAAAAYA4",
      "assignAction" : "ASSIGN_DRAFT_TRANSLATION"
    }
  ]
}
```

Sample Output

The following code sample illustrates a response after a successful request.

```
[ {
  "actionName" : "assignKnowledgeArticles",
  "errors" : null,
  "isSuccess" : true,
  "outputValues" : {
    "ka0RM00000004Ve" : "Success",
    "ka0RM00000003do" : "Success"
  }
} ]
```

Create Draft from Online Knowledge Article

URI: `/services/data/v`***XX.X***`/actions/standard/createDraftFromOnlineKnowledgeArticle`

Table 3: Inputs

Input	Details
`action`	**Type** string **Description** Required. Edit action for primary language or translation articles. Valid actions are: • EDIT_AS_DRAFT_ARTICLE • EDIT_AS_DRAFT_TRANSLATION
`unpublish`	**Type** boolean

79

Input	Details
	Description Required. Indicates whether to keep the article published (`false`) or archive the published article (`true`). Use `false` to keep the current article version online and create a draft. Use `true` to archive the current online version, which removes it from the knowledge base, and creates a draft.
articleVersionId	**Type** string **Description** Article version ID. Required to create a draft from an online (published) translation. Optional to create a draft from the online primary article if the Article ID is provided.
articleId	**Type** string **Description** Article ID. Required when creating a draft from the online (published) primary article if the Article Version ID isn't provided.

Sample Input

The following code sample creates a draft from a primary article and archives the original article:

```
{
  "inputs" : [
    {
      "action" : "EDIT_AS_DRAFT_ARTICLE",
      "unpublish" : true,
      "articleId" : "kA0RM00000004pP0AQ"
    }
  ]
}
```

Sample Output

The following code sample illustrates a response after a successful request.

```
[ {
  "actionName" : "createDraftFromOnlineKnowledgeArticle",
  "errors" : null,
  "isSuccess" : true,
  "outputValues" : {
    "kA0RM00000004pP0AQ" : "Success"
  }
} ]
```

Delete Knowledge Articles

URI: `/services/data/v`***XX.X***`/actions/standard/deleteKnowledgeArticles`

Table 4: Inputs

Input	Details
articleVersionIdList	**Type** string **Description** Required. Comma-separated article version ID list.

Sample Input

The following code sample deletes two articles:

```
{
  "inputs" : [
    {
      "articleVersionIdList" : [ "ka0RM00000004VeYAI", "ka0RM00000003doYAA" ]
    }
  ]
}
```

Sample Output

The following code sample illustrates a response after a successful request.

```
[ {
  "actionName" : "deleteKnowledgeArticles",
  "errors" : null,
  "isSuccess" : true,
  "outputValues" : {
    "ka0RM00000004Ve" : "Success",
    "ka0RM00000003do" : "Success"
  }
} ]
```

Publish Knowledge Articles

URI: /services/data/v**XX.X**/actions/standard/publishKnowledgeArticles

Table 5: Inputs

Input	Details
articleVersionIdList	**Type** string **Description** Required. Comma-separated article version ID list.
pubAction	**Type** string **Description** Required. Publish action. Valid actions are:

Input	Details
	• PUBLISH_ARTICLE (which replaces the latest version)
	• PUBLISH_ARTICLE_NEW_VERSION (which creates a new version)
	• SCHEDULE_ARTICLE_FOR_PUBLICATION
	• PUBLISH_TRANSLATION
`pubDate`	**Type** string **Description** Optional. Scheduled publish date in ISO 8601 format `yyyy-MM-dd\'T\'HH:mm:ss.SSSZ`. For example, for February 8, 2023, 1:40 pm UTC+01:00 use `2023-02-08T13:40:00.000+0100`.

Sample Input

The following code sample publishes two articles:

```
{
  "inputs" : [
    {
      "articleVersionIdList" : [ "ka0RM00000004VeYAI", "ka0RM00000003doYAA" ],
      "pubAction" : "PUBLISH_ARTICLE"
    }
  ]
}
```

Sample Output

The following code sample illustrates a response after a successful request.

```
[ {
  "actionName" : "publishKnowledgeArticles",
  "errors" : null,
  "isSuccess" : true,
  "outputValues" : {
    "ka0RM00000004Ve" : "Success",
    "ka0RM00000003do" : "Success"
  }
} ]
```

Restore Knowledge Article Version

URI: `/services/data/v``XX.X``/actions/standard/restoreKnowledgeArticleVersion`

Table 6: Inputs

Input	Details
`action`	**Type** string

Input	Details
	Description Required. The only valid action is: RESTORE_KNOWLEDGE_ARTICLE_VERSION
articleId	**Type** string **Description** Required. Article ID.
versionNumber	**Type** integer **Description** Optional. Version number of the archived article version to restore. Default is the latest archived version.

Sample Input

The following code restores the latest archived version:

```
{
  "inputs" : [
    {
      "action" : "RESTORE_KNOWLEDGE_ARTICLE_VERSION",
      "articleId" : "kA0RM00000004pP0AQ"
    }
  ]
}
```

The following code restores a past archived version of a published article:

```
{
  "inputs" : [
    {
      "action" : "RESTORE_KNOWLEDGE_ARTICLE_VERSION",
      "versionNumber":3,
      "articleId" : "kA0RM00000004pP0AQ"
    }
  ]
}
```

The following code restores two archived articles:

```
{
  "inputs" : [
    {
      "action" : "RESTORE_KNOWLEDGE_ARTICLE_VERSION",
      "articleId" : "kA0RM00000004pP0AQ"
    },
    {
      "action" : "RESTORE_KNOWLEDGE_ARTICLE_VERSION",
```

```
        "articleId" : "kA0RM00000004pP0AB"
    }
  ]
}
```

Sample Output

The following code sample illustrates a response after a successful request.

```
[ {
  "actionName" : "restoreKnowledgeArticleVersion",
  "errors" : null,
  "isSuccess" : true,
  "outputValues" : {
    "kA0RM00000004pP0AQ" : "Success"
  }
} ]
```

Retrieve Smart Link URL

URI: /services/data/v**XX.X**/actions/standard/getArticleSmartLinkUrl

Table 7: Inputs

Input	Details
articleVersionId	**Type** string **Description** Required. The ID of the Knowledge article version.

Sample Input

The following code sample retrieves the SmartLink URL of a Knowledge article version:

```
{
    "inputs":[
        {
            "articleVersionId":"ka0xx00000000cjAAA"
        }
    ]
}
```

Sample Output

The following code sample illustrates a response after a successful request.

```
[
    {
        "actionName":"getArticleSmartLinkUrl",
        "errors":null,
        "isSuccess":true,
        "outputValues":{
```

```
"articleSmartLinkUrl":"https://example.lightning.force.com/lightning/articles/Knowledge/Test-Redirection-1"


        }
    }
]
```

Submit Knowledge Article for Translation

URI: `/services/data/v`***XX.X***`/actions/standard/submitKnowledgeArticleForTranslation`

Table 8: Inputs

Input	Details
`articleId`	**Type** string **Description** Required. Article ID.
`language`	**Type** string **Description** Required. Language code for the translation.
`assigneeId`	**Type** ID **Description** Required. ID of the assigned user.
`dueDate`	**Type** string **Description** Optional. Assigned due date.
`sendEmailNotification`	**Type** boolean **Description** Optional. Indicates whether to send an email notification. Defaults to `false`.

Table 9: Outputs

Output	Details
`articleId`	**Type** ID

Output	Details
	Description Article ID.
language	**Type** string
	Description Language code for the translation.

Sample Input

The following code sample submits one article for translation into Spanish:

```
{
  "inputs" : [
    {
      "articleId" : "kA0RM00000004pP0AQ",
      "language" : "es",
      "assigneeId" : "005RM00000AAAAAYA4"
    }
  ]
}
```

Sample Output

The following code sample illustrates a response after a successful request.

```
[ {
  "actionName" : "submitKnowledgeArticleForTranslation",
  "errors" : null,
  "isSuccess" : true,
  "outputValues" : {
    "articleId" : "kA0RM00000004pP0AQ",
    "language" : "es"
  }
} ]
```

Error Response Types

Knowledge actions can respond with two types of error responses: action-scoped errors and item-scoped errors.

Action-scoped errors describe an error about the overall action that you're trying to invoke. Action-scoped errors have a `statusCode` in addition to a `message`. This example illustrates an action-scoped error caused by sending invalid input values.

```
[ {
        "actionName" : "restoreKnowledgeArticleVersion",
        "errors" : [ {
        "statusCode" : "INVALID_API_INPUT",
        "message" : "You can't perform this action. Be sure the action is valid for the
current state of the article, and that you have permission to perform it.",
        "fields" : [ ]
        } ],
```

```
        "isSuccess" : false,
        "outputValues" : null
        } ]
```

Item-scoped errors describe a problem with a specific article or article version within the action. For example, this code illustrates an archiveKnowledgeArticles action response with one failed item and one successful item.

```
[ {
        "actionName" : "archiveKnowledgeArticles",
        "errors" : null,
        "isSuccess" : false,
        "outputValues" : {
        "ka0RM00000004Ve" : "You can't perform this action. Be sure the action is valid
for the current state of the article, and that you have permission to perform it.",
        "ka0RM00000003do" : "Success"
        }
        } ]
```

If any type of error occurs with an action, the `isSuccess` field is `false`.

Manage Knowledge with REST APIs

REST APIs provide programmatic access to many actions you can perform on your Knowledge base, an article, and its translations.

IN THIS SECTION:

Archive the Primary Version of an Article

Archives the primary version of an article. The actions are defined by the field change you request on the resource. To archive the primary version, use `"publishStatus":"Archived"`. To schedule a date for archiving, use `"archiveScheduleDate" : <date>`.

Assign a Task Related to a Primary Article

Assigns a task to a user for a primary article, including due date and instructions. The actions are defined by the field change you request on the resource.

Assign a Task Related to a Translation

Assigns a task to a user for a translated article, including due date and instructions. The actions are defined by the field change you request on the resource.

Delete a Primary Version of an Article

Deletes the primary version of an article.

Delete a Translated Version of an Article

Deletes a translated version of an article.

Edit an Online Version of a Primary Article

Creates a draft copy of the online version of a primary article. This does not unpublish the online version.

Get a List of Salesforce Site and Experience Cloud Site URLs for an Online Article

Returns a list of article URLs in Salesforce Sites, Experience Cloud sites, or both. Article URLs from Salesforce Sites are returned if the article is visible in a public knowledge base. Article URLs from Experience Cloud sites are returned if the article is visible to partners or customers. This API creates the URLs for the **Insert URL into Email** Lightning action.

Publish the Primary Version of an Article

Publishes the primary version of an article.

Restore an Archived Version of an Article

Restores an archived version of the article. If `versionNumber` isn't specified, restores the latest version of the archived article.

Retrieve Article Metadata

Retrieves the metadata of an article.

Get Knowledge Language Settings

Gets the existing Knowledge language settings, including the default knowledge language and a list of supported Knowledge language information. This resource can be used in API version 31.0 and later.

Retrieve a Version of an Article

Retrieves the version ID of an article.

Search for Metadata Elements of a Primary Version

Searches for metadata elements of the online primary version of an article.

Search for Metadata Elements of a Translated Version

Searches for metadata elements of a translated version of an article.

Set a Translated Article Version to Complete

Sets a translated article version to complete.

Set a Translated Article Version to Incomplete

Sets a translated article version to incomplete.

Submit an Article for Translation

Submits an article for translation for multiple languages and blocks translations for inactive languages.

Unpublish the Primary Version of an Article

Unpublishes the online primary version of an article when there isn't an existing draft article.

Unpublish the Online Version of a Translated Article

Unpublishes the online version of a translated article.

Parameterized Search

Executes a simple REST search using parameters instead of a SOSL clause. Indicate parameters in the URI with the GET method. Or, use the POST method to create complex searches in a request body.

Search

Executes the specified SOSL search. The search string must be URL-encoded.

Search Scope and Order

Returns an ordered list of objects in the default global search scope of a logged-in user. Global search keeps track of which objects the user interacts with and how often, and arranges the search results accordingly. Objects used most frequently appear at the top of the list.

Search Result Layouts

Returns search result layout information for the objects in the query string. For each object, this call returns the list of fields displayed on the search results page as columns, the number of rows displayed on the first page, and the label used on the search results page.

Search for Records Suggested by Autocomplete and Instant Results

Returns a list of suggested records whose names match the user's search string. The suggestions resource provides autocomplete results and instant results for users to navigate directly to likely relevant records, before performing a full search. This resource is available in REST API version 32.0 and later.

Returns a list of Salesforce Knowledge article titles that match the user's search query string. Provides a shortcut to navigate directly to likely relevant articles before the user performs a search. This resource is available in REST API version 30.0 and later.

Returns a list of suggested searches based on the user's query string text matching searches that other users have performed in Salesforce Knowledge. Provides a way to improve search effectiveness, before the user performs a search. This resource is available in REST API version 30.0 and later.

Archive the Primary Version of an Article

Archives the primary version of an article. The actions are defined by the field change you request on the resource. To archive the primary version, use `"publishStatus":"Archived"`. To schedule a date for archiving, use `"archiveScheduleDate" : <date>`.

🛑 **Important:** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Syntax

URI

```
/services/data/v25.0/knowledgeManagement/articleVersions/masterVersions/<versionID>
```

Formats

JSON, XML

HTTP Method

PATCH

Authentication

`Authorization: OAuth` ***accesstoken***

Parameters

Parameter	Description
publishStatus	Publishing status of the article. Use `archived`.
archiveScheduleDate	Date to archive the article.

Input:

Archive:

```
{
      "publishStatus":"Archived"
      }
```

Schedule for archiving (using GMT date format):

```
{
       "archiveScheduleDate" : "2012-04-19T07:00:00.000+0000"
       }
```

Output:
   HTTP status code 204 is returned when an existing record is updated.

Assign a Task Related to a Primary Article

Assigns a task to a user for a primary article, including due date and instructions. The actions are defined by the field change you request on the resource.

Syntax

URI

```
/services/data/v25.0/knowledgeManagement/articleVersions/masterVersions/<versionID>
```

Formats
   JSON, XML

HTTP Method
   PATCH

Authentication
   Authorization: OAuth **_accesstoken_**

Parameters

Parameter	Description
assigneeId	Assigns the primary article to a user ID or a queue ID.
dueDate	Date that the task is due.
instruction	Instructions for the task.

Input:

```
{
      "assigneeId":"05Dxx0000dsads",
      "dueDate":"2012-04-19T07:00:00.000+0000",
      "instruction":"Please review."
      }
```

Output:
   HTTP status code 204 is returned when an existing record is updated.

Assign a Task Related to a Translation

Assigns a task to a user for a translated article, including due date and instructions. The actions are defined by the field change you request on the resource.

Syntax

URI

```
/services/data/v25.0/knowledgeManagement/articleVersions/translations/<translationVersionId>
```

Formats
JSON, XML

HTTP Method
PATCH

Authentication
Authorization: OAuth *accesstoken*

Parameters

Parameter	Description
assigneeId	Assigns the primary article to a user ID or a queue ID.
dueDate	Date that the task is due.
instruction	Instructions for the task.

Input:

```
{
        "assigneeId":"05Dxx0000dsads",
        "dueDate":"2012-04-19T07:00:00.000+0000",
        "instruction":"Please review."
        }
```

Output:
HTTP status code 204 is returned when an existing record is updated.

Delete a Primary Version of an Article

Deletes the primary version of an article.

Syntax

URI

```
/services/data/v25.0/knowledgeManagement/articleVersions/masterVersions/<versionID>
```

Formats
JSON, XML

HTTP Method
DELETE

Authentication
Authorization: OAuth *accesstoken*

Parameters
   None

Input:
   None required

Output:
   HTTP status code 204 is returned when an existing record is deleted.

Delete a Translated Version of an Article

Deletes a translated version of an article.

Syntax

URI

```
/services/data/v25.0/knowledgeManagement/articleVersions/translations/<versionID>
```

Formats
   JSON, XML

HTTP Method
   DELETE

Authentication
   Authorization: OAuth *accesstoken*

Parameters
   None

Input
   None needed

Output
   HTTP status code 204 is returned when an existing record is deleted.

Edit an Online Version of a Primary Article

Creates a draft copy of the online version of a primary article. This does not unpublish the online version.

Syntax

URI

```
/services/data/v25.0/knowledgeManagement/articleVersions/masterVersions
```

Formats
   JSON, XML

HTTP Method
   POST

Authentication
   Authorization: OAuth *accesstoken*

Parameters

Parameter	Description
articleId	The ID of the article.

Input:

```
{
      "articleId":<articleID>
      }
```

Get a List of Salesforce Site and Experience Cloud Site URLs for an Online Article

Returns a list of article URLs in Salesforce Sites, Experience Cloud sites, or both. Article URLs from Salesforce Sites are returned if the article is visible in a public knowledge base. Article URLs from Experience Cloud sites are returned if the article is visible to partners or customers. This API creates the URLs for the **Insert URL into Email** Lightning action.

The org must have Lightning Knowledge enabled, and **Allow users to share articles via public URLs** enabled in Knowledge Settings. In addition, the API user must have read access on Knowledge.

Available in API versions 46.0 and later.

Syntax

URI

```
/services/data/v46.0/knowledgeManagement/articles/siteListForOnlineArticle?articleId=articleId&language=xx_XX
```

Formats
  JSON, XML

HTTP Method
  GET

Authentication
  Authorization: OAuth *accesstoken*

Parameters

Parameter	Description
articleId	Either the 15– or 18–digit version of the article version ID.
language	Language for the selected article.

Input
  None

Publish the Primary Version of an Article

Publishes the primary version of an article.

🛑 **Important:** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

The publishing actions are defined by the field change you request on the resource. To publish a minor version, use `"publishStatus":"Online"`. To publish a major version, use `"publishStatus":"Online"` and `"versionNumber":"NextVersion"`. To schedule a publication date, use `"publishScheduleDate" : <date>`.

Syntax

URI

```
/services/data/v25.0/knowledgeManagement/articleVersions/masterVersions/<versionId>
```

Formats
JSON, XML

HTTP Method
PATCH

Authentication
Authorization: OAuth **accesstoken**

Parameters

Parameter	Description
publishStatus	Publishing status of the article. Use `online`.
versionNumber	Version of the article.
publishScheduleDate	Date to publish the article in ISO 8601 format YYYY-MM-DDTHH:mm:ss+/-HHmm. For example, for February 8, 2023, 1:40 pm UTC+01:00 use `2023-02-08T13:40:31+0100..`

Input
Publish a minor version:

```
{
        "publishStatus":"Online"
        }
```

Publish a major version:

```
{
        "publishStatus":"Online",
        "versionNumber":"NextVersion"
        }
```

Schedule for publication:

```
{
        "publishScheduleDate" : "2012-05-19T07:00:00+0000"
        }
```

Output

HTTP status code 204 is returned when an existing record is updated.

Restore an Archived Version of an Article

Restores an archived version of the article. If `versionNumber` isn't specified, restores the latest version of the archived article.

🛑 **Important:** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Syntax

URI

```
/services/data/v25.0/knowledgeManagement/articleVersions/masterVersions/
```

Formats

JSON, XML

HTTP Method

POST

Authentication

`Authorization: OAuth ` ***accesstoken***

Parameters

Parameter	Description
articleId	The ID of the article.
versionNumber	Version of the article. If this field is not specified, the latest version of the archived article is restored.

Input:

```
{
"articleId":"<articleID>",
"versionNumber": <number>
}
```

Retrieve Article Metadata

Retrieves the metadata of an article.

Syntax

URI

```
/services/data/v25.0/knowledgeManagement/articles/<articleId>
```

Formats
JSON, XML

HTTP Method
GET

Authentication
Authorization: OAuth *accesstoken*

Parameters
None

Input
None required

Get Knowledge Language Settings

Gets the existing Knowledge language settings, including the default knowledge language and a list of supported Knowledge language information. This resource can be used in API version 31.0 and later.

Salesforce Knowledge must be enabled in your organization. It gets the Knowledge language settings, including the default knowledge language and a list of supported Knowledge language information.

Syntax

URI
```
/services/data/v*XX.X*/knowledgeManagement/settings
```

Formats
JSON, XML

HTTP methods
GET

Authentication
Authorization: Bearer *token*

Request body
None required

Request parameters
None

Example

Example Request

```
curl
https://MyDomainName.my.salesforce.com/services/data/v66.0/knowledgeManagement/settings
 -H "Authorization: Bearer token"
```

Example Response Body

```
{
    "defaultLanguage" : "en_US",
    "knowledgeEnabled" : true,
    "languages" : [ {
    "active" : true,
    "name" : "en_US"
    }, {
    "active" : true,
    "name" : "it"
    }, {
    "active" : true,
    "name" : "zh_CN"
    }, {
    "active" : true,
    "name" : "fr"
    } ]
}
```

Retrieve a Version of an Article

Retrieves the version ID of an article.

🚫 Important: Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

Syntax

URI

```
/services/data/v25.0/knowledgeManagement/articleVersions/masterVersions/<articleVersionId>
```

Formats
   JSON, XML

HTTP Method
   GET

Authentication
   Authorization: OAuth *accesstoken*

Parameters
   None

Input
   None required

Search for Metadata Elements of a Primary Version

Searches for metadata elements of the online primary version of an article.

Syntax

URI

```
/services/data/v25.0/knowledgeManagement/articleVersions/masterVersions?filterArticleId=value1&FilterPublishStatus=value2
```

Formats
JSON, XML

HTTP Method
GET

Authentication
Authorization: OAuth *accesstoken*

Example
This example searches for the online version of the primary article 'kA0x50000000jsh':

```
/services/data/v25.0/knowledgeManagement/articleVersions/masterVersions?filterArticleId=kA0x50000000jsh&filterPublishStatus=online"
```

Search for Metadata Elements of a Translated Version

Searches for metadata elements of a translated version of an article.

Syntax

URI

```
/services/data/v25.0/knowledgeManagement/articleVersions/translations?filterArticleId=value1&
  filterLanguage=value2&FilterPublishStatus=value3";
```

Formats
JSON, XML

HTTP Method
GET

Authentication
Authorization: OAuth *accesstoken*

Example
This example searches for the German online translation of the article 'kA0x50000000jsh':

```
/services/data/v25.0/knowledgeManagement/articleVersions/translations?filterArticleId=kA0x50000000jsh&
  filterLanguage=de&filterPublishStatus=online"
```

Set a Translated Article Version to Complete

Sets a translated article version to complete.

Syntax

URI

```
/services/data/v25.0/knowledgeManagement/articleVersions/translations/<translationVersionID>
```

Formats
    JSON, XML

HTTP Method
    PATCH

Authentication
    `Authorization: OAuth` ***accesstoken***

Parameters

Parameter	Description
`complete`	Set this value to `true` for translations that are complete.

Input:

```
{
        "complete":"true"
        }
```

Output:
    HTTP status code 204 is returned when an existing record is updated.

Set a Translated Article Version to Incomplete

Sets a translated article version to incomplete.

Syntax

URI

```
/services/data/v25.0/knowledgeManagement/articleVersions/translations/<translationVersionID>
```

Formats
    JSON, XML>

HTTP Method
    PATCH

Authentication
    `Authorization: OAuth` ***accesstoken***

Parameters

Parameter	Description
`complete`	Set this value to `false` to set a translation to incomplete.

Input:

```
{
        "complete":"false"
        }
```

Output:

HTTP status code 204 is returned when an existing record is updated.

Submit an Article for Translation

Submits an article for translation for multiple languages and blocks translations for inactive languages.

Syntax

URI

```
/knowledgeManagement/articleVersions/translations
```

Available since release

25.0

Some parameters are supported in API versions 43.0 and higher.

Formats

JSON, XML

HTTP Method

POST

Authentication

Authorization: OAuth **accesstoken**

Parameters

Parameter	Description	Available
articleID	String. Required. The ID of the article.	25.0
assignments	JSON Array. An array of assignment details. Each assignment is a list with these properties: • language: Required. A language code. • assigneeId: A user Id or queue Id. • dueDate: The date that the task is due. If you use this parameter, don't use the language parameter.	43.0
language	String. A language code. If you use this parameter, don't use the assignments parameter.	25.0-42.0
sendEmailNotification	Boolean. Specifies whether to send an email to the assignee (true) or not (false).	43.0

Example Request Body

```
{
   "articleId":"kA0xx00000000BO",
   "assignments":[
       {
```

```
            "language":"sq",
            "assigneeId":"005xx000001T7MF",
            "dueDate":""
        },
        {
            "language":"zh_CN",
            "assigneeId":"005xx000001T7MF",
            "dueDate":""
        }
    ],
    "sendEmailNotification":true
}
```

Unpublish the Primary Version of an Article

Unpublishes the online primary version of an article when there isn't an existing draft article.

The publishing actions are defined by the field change you request on the resource. To unpublish the primary version, use
`"publishStatus":"draft"`.

Syntax

URI

```
/services/data/v25.0/knowledgeManagement/articleVersions/masterVersions/<versionId>
```

Formats
   JSON, XML

HTTP Method
   PATCH

Authentication
   `Authorization: OAuth accesstoken`

Parameters

Parameter	Description
publishStatus	Publishing status of the article. Use draft.

Input

```
{
        "publishStatus":"draft"
        }
```

Output:
   HTTP status code 204 is returned when an existing record is updated.

Unpublish the Online Version of a Translated Article

Unpublishes the online version of a translated article.

The publishing actions are defined by the field change you request on the resource. To edit and remove a translation from online state, use `"publishStatus":"draft"`.

Syntax

URI

```
/services/data/v25.0/knowledgeManagement/articleVersions/translations/<translationVersionID>
```

Formats
JSON, XML

HTTP Method
PATCH

Authentication
`Authorization: OAuth ` ***accesstoken***

Parameters

Parameter	Description
publishStatus	Publishing status of the article. Use `draft`.

Input:

```
{
        "publishStatus":"draft"
        }
```

Parameterized Search

Executes a simple REST search using parameters instead of a SOSL clause. Indicate parameters in the URI with the GET method. Or, use the POST method to create complex searches in a request body.

Search

Executes the specified SOSL search. The search string must be URL-encoded.

For more information on SOSL see the *SOQL and SOSL Reference*.

Syntax

URI
`/services/data/v`***XX.X***`/search/?q=`***SOSL_searchString***

Formats
JSON, XML

HTTP Method
GET

Authentication
`Authorization: Bearer ` ***token***

Parameters

Parameter	Description
q	A SOSL statement that is properly URL-encoded.

Example

See Search for a String.

Search Scope and Order

Returns an ordered list of objects in the default global search scope of a logged-in user. Global search keeps track of which objects the user interacts with and how often, and arranges the search results accordingly. Objects used most frequently appear at the top of the list.

The returned list reflects the object order in the user's default search scope, including any pinned objects on the user's search results page. This call is useful if you want to implement a custom search results page using the optimized global search scope. The search string must be URL-encoded.

Syntax

URI
```
/services/data/vXX.X/search/scopeOrder
```

Formats
    JSON, XML

HTTP Method
    GET

Authentication
```
Authorization: Bearer token
```

Example

See Get the Default Search Scope and Order.

Search Result Layouts

Returns search result layout information for the objects in the query string. For each object, this call returns the list of fields displayed on the search results page as columns, the number of rows displayed on the first page, and the label used on the search results page.

This call supports bulk fetch for up to 100 objects in a query.

Syntax

URI
```
/services/data/vXX.X/search/layout/?q=commaDelimitedObjectList
```

Formats
    JSON, XML

HTTP Method
    GET

Authentication
    `Authorization: Bearer` ***token***

Response format

Property	Type	Description
`field`	String	Object and field name formatted with a period separating. For example: `Account.Name`.
`format`	String	The type of date field, such as the date only or date and time. Only date related types are specified; otherwise, `null`.
`label`	String	Name as it appears to users
`name`	String	API name

Example

See Get Search Result Layouts for Objects.

Search for Records Suggested by Autocomplete and Instant Results

Returns a list of suggested records whose names match the user's search string. The suggestions resource provides autocomplete results and instant results for users to navigate directly to likely relevant records, before performing a full search. This resource is available in REST API version 32.0 and later.

The suggestions resource returns records when the record's name field includes the exact text in the search string. The last term in the search string can match the beginning of a word. Records that contain the search string within a word aren't considered a match.

> **Note:** If the user's search query contains quotation marks or wildcards, those symbols are automatically removed from the query string in the URI.

For example, the text string `national u` is treated as `national u*` and returns "National Utility", "National Urban Company", and "First National University".

The suggestions resource returns display-ready data about likely relevant records that the user can access. A relevance algorithm determines the order of results. Each suggested record in the results contains these elements:

Element	Description
`Attributes`	The record's object type and the URL for accessing the record.
	Also includes the requested lookup fields' values. For example, if you requested `fields=Id,Name`, the result would include the ID and name.

Element	Description
Name (or Title)	The record's Name field. In the absence of a standard Name field, the Title field is used for these objects: • Dashboard • Idea • IdeaTheme • Note • Question In the absence of a standard Name or Title field, the main identifying field is used. For example, in cases, the Case Number is used.
Id	The record's unique identifier.

The suggestions resource supports all searchable objects except the following.

- ContentNote
- Event
- External objects
- FeedComment
- FeedPost
- IdeaComment
- Pricebook2
- Reply
- TagDefinition
- Task

Syntax

URI

/services/data/v**XX.X**/search/suggestions?q=**searchString**&sobject=**objectTypes**

Formats

JSON, XML

HTTP methods

GET

Authentication

Authorization: Bearer **token**

Request body

None required

Request parameters

Parameter	Description
`fields`	Optional. Used for creating lookup queries. Specify multiple fields using a comma-separated list. Specifies which lookup fields to be returned in the response.
`dynamicFields`	Optional. Available in API version 48.0 and later. Used to return additional dynamic fields. Specify multiple options using a comma-separated list. For example, if `dynamicFields=secondaryField` then each suggested record in the results contains an additional field besides `Id` and `Name` (or `Title`) based on the next eligible field in the search layout.
`groupId`	Optional. Specifies one or more unique identifiers of one or more groups that the question to return was posted to. Specify multiple groups using a comma-separated list. This parameter is only applicable when the parameter type equals `question`. Don't use with the `userId`.
`ignoreUnsupportedSObjects`	Optional. If an unsupported object is included in a request, this parameter indicates what action to take. If it's set to `false`, an error is returned. If it's set to `true`, the object is ignored and no error is returned. See the Unsupported Objects section for reference. The default is `false`.
`limit`	Optional. Specifies the maximum number of suggested records to return. If a limit isn't specified, 5 records are returned by default. If there are more suggested records than the limit specified, the response body's `hasMoreResults` property is `true`.
`networkId`	Optional. Specifies one or more unique identifiers for the Experience Cloud sites to return the question to. Specify multiple sites using a comma-separated list. This parameter is only applicable when the parameter `type` equals `question` or parameter `sobject` equals `user`.
`q`	Required. The user's search query string, properly URL-encoded. Suggestions are returned only if the user's query string meets the minimum length requirements: one character for queries in Chinese, Japanese, Korean, and Thai; three characters for all other languages. Query strings that exceed the maximum length of 255 characters (or 200 consecutive characters without a space break) return an error.
`sobject`	Required. The objects that the search is scoped to, such as Account or offer__c.
	If the `sobject` value is `feedItem`, the `type` parameter is required and it must have a value of `question`.
	Specify up to 10 objects with a comma-separated list. For example: `sobject=Account,Contact,Lead`. To take advantage of the feature, activate the **CrossObjectTypeahead** permission.
	To specify the specific fields to return by object, use the following syntax with multiple fields in a comma-separated list. The *sobject* is lowercase.
	`sobject=`***sobject***`.fields=`***fields***

Parameter	Description
	For example: ```\n&sobject=Account,Contact,Lead&account.fields=Website,Phone\n&contact.fields=Phone\n```
`topicId`	Optional. Specifies the unique identifier of the single topic that the question to return was tagged as. This parameter is only applicable when the parameter `type` equals `question`.
`type`	Required when the `sobject` value is `feedItem`. Including this parameter for all other `sobject` values doesn't affect the query. Specifies that the type of Feed is questions. Valid value: `question`.
`userId`	Optional. Specifies one or more unique identifiers of one or more users who authored the question to return. Specify multiple users using a comma-separated list. This parameter is only applicable when the parameter type equals `question`. Don't use with the `groupId`.

`useSearchScope`	Optional. Available in API version 40.0 and later. The default value is `false`. If `false`, the objects specified in the request are used to suggest records. If `true`, in addition to the objects specified in the request, the user's search scope is used to suggest records. The search scope is the list of objects a user uses most frequently.
	
	• If the request doesn't specify an object, use `useSearchScope=true`.
	• If `useSearchScope=true` and the user's search scope is empty, the default search scope is used to suggest records.
	• Typically, only the first 10 objects are used to suggest records. However, an admin can assign objects that are always considered when returning results. If configured, up to 15 objects are used to suggest records. For more information about assigning objects, see Assign Search Results Objects to Users (Beta).
	• Objects specified in the `sobject` parameter are prioritized over objects in the user's search scope.
	• Values for the `ignoreUnsupportedSObjects` parameter aren't applied to the objects in the search scope.
	
	This example uses only the search scope.
	
	```\n.../search/suggestions?q=Acme&useSearchScope=true\n```
	
	This example uses the search scope and the Account object.
	
	```\n.../search/suggestions?q=Acme&sobject=Account&useSearchScope=true\n```

`where`	Optional. A filter that follows the same syntax as the SOQL `WHERE` clause. URLs encode the expression. Use the clause for an object, or globally for all compatible objects. An example of an object-specific clause is: `account.where=name%20LIKE%20%27Smith%25%27`. An example of a global clause is: `where=name%20LIKE%20%27Smith%25%27`. The parameter

Parameter	Description
	must be lower case. Any object-specific `where` clauses override the global `where` clause. You can't use this parameter for the Question object.

To specify multiple entities, see the following example. This feature is available in version 38.0 and later.

```
...search/suggestions?q=Smith
&sobject=Account,Contact,KnowledgeArticleVersion,CollaborationGroup,Topic,FeedItem

// Specifies a global where clause (to filter Account and
 Contact)
&where=name%20LIKE%20%27Smith%25%27
// Overrides the global where clause for Knowledge Article

(filtering by PublishStatus and Language is required for
 KnowledgeArticle)
&knowledgearticleversion.where=PublishStatus='online'+and+language='en_US'
// Overrides the global where clause for Topic
&topic.where=networkid=<1234567891>
// Overrides the global where clause for
CollaborationGroup
&collaborationgroup.where=networkid=<1234567891>
// FeedItem-Question doesn't support where clauses, but
we can filter
the type and networkId&type=question
&networkId==<1234567891>
```

Example

Example Response Body

```
{
  "autoSuggestResults" : [ {
      "attributes" : {
        "type" : "Account",
        "url" : "/services/data/v66.0/sobjects/Account/001xx000003DH6WAAW"
      },
      "Id" : "001xx000003DH6WAAW",
      "Name" : "National Utility Service"
    }, {
    {
      "attributes" : {
        "type" : "Account",
        "url" : "/services/data/v66.0/sobjects/Account/001xx000003DHJ4AAO"
      },
      "Id" : "001xx000003DHJ4AAO",
      "Name" : "National Utility Service"
    }, {
    {
      "attributes" : {
        "type" : "Account",
```

```
        "url" : "/services/data/v66.0/sobjects/Account/001xx000003DHscAAG"
      },
      "Id" : "001xx000003DHscAAG",
      "Name" : "National Urban Technology Center"
    } ],
  "hasMoreResults" : false,
  "meta" : {
    "nameFields" : [ {
      "entityApiName" : "Account",
      "fieldApiName" : "Name"
    } ],
    "secondaryFields" : [ ]
  }
}
```

Example Response Body for a Multiple Object Request

```
{
  "autoSuggestResults" : [ {
      "attributes" : {
        "type" : "Account",
        "url" : "/services/data/v66.0/sobjects/Account/001xx000003DMEKAA4"
      },
      "Id" : "001xx000003DMEKAA4"
      "Name" : "Joe Doe Printing"
    }, {
    {
      "attributes" : {
        "type" : "Account",
        "url" : "/services/data/v66.0/sobjects/Account/001xx000003DLjvAAG"
      },
      "Id" : "001xx000003DLjvAAGO"
      "Name" : "Joe Doe Plumbing"
    }, {
    {
      "attributes" : {
        "type" : "Contact",
        "url" : "/services/data/v66.0/sobjects/Contact/003xx000004U9Y9AAK"
      },
      "Id" : "003xx000004U9Y9AAK"
      "Name" : "John Doe"
    } ],
  "hasMoreResults" : false,
  "meta" : {
    "nameFields" : [ {
      "entityApiName" : "Account",
      "fieldApiName" : "Name"
    }, {
      "entityApiName" : "Contact",
      "fieldApiName" : "Name"
    } ],
    "secondaryFields" : [ ]
  }
}
```

Example XML Response Body

```xml
<?xml version="1.0" encoding="UTF-8"?
<suggestions>
  <autoSuggestResults type="Account"
url="/services/data/v66.0/sobjects/Account/001xx000003DH6WAAW">
    <Id>001xx000003DH6WAAW</Id>
    <Name>National Utility Service</Name>
  </autoSuggestResults>
  <autoSuggestResults type="Account"
url="/services/data/v66.0/sobjects/Account/001xx000003DHJ4AAO">
    <Id>001xx000003DHJ4AAO</Id>
    <Name>National Utility Service</Name>
  </autoSuggestResults>
  <autoSuggestResults type="Account"
url="/services/data/v66.0/sobjects/Account/001xx000003DHscAAG">
    <Id>001xx000003DHscAAG</Id>
    <Name>National Urban Technology Center</Name>
  </autoSuggestResults>
  <hasMoreResults>true</hasMoreResults>
  <meta>
    <nameFields>
      <entityApiName>Account</entityApiName>
      <fieldApiName>Name</fieldApiName>
    </nameFields>
    <nameFields>
      <entityApiName>ContentDocument</entityApiName>
      <fieldApiName>Title</fieldApiName>
    </nameFields>
  </meta>
</suggestions>
```

Search Suggested Article Title Matches

Returns a list of Salesforce Knowledge article titles that match the user's search query string. Provides a shortcut to navigate directly to likely relevant articles before the user performs a search. This resource is available in REST API version 30.0 and later.

Salesforce Knowledge must be enabled in your organization. The user must have the "View Articles" permission enabled. The articles suggested include only the articles the user can access, based on the data categories and article types the user has permissions to view.

The Suggest Article Title Matches resource is designed to return display-ready data about likely relevant articles. Articles are suggested if their titles contain the entire query string, except stopwords, such as "a," "for," and "the."

For example, a search for `Backpacking for desert` returns the article, "Backpacking in the desert."

> **Note:** Articles with titles that include stopwords from the query string, such as "Backpacking for desert survival" in this example, appear before matching articles with titles that don't include the stopwords.
>
> Stopwords at the end of the query string are treated as search terms.

A wildcard is automatically appended to the last token in the query string.

> **Note:** If the user's search query contains quotation marks or wildcards, those symbols are automatically removed from the query string in the URI along with any other special characters.

If the number of suggestions returned exceeds the limit specified in the request, the end of the response contains a field called `hasMoreResults`. Its value is `true` if the suggestions returned are only a subset of the suggestions available, and `false` otherwise.

Syntax

URI

/services/data/v*XX.X*/search/suggestTitleMatches?q=*searchString*
&language=*articleLanguage*&publishStatus=*articlePublicationStatus*

Formats

JSON, XML

HTTP methods

GET

Authentication

Authorization: Bearer *token*

Request body

None required

Request parameters

Parameter	Description
articleTypes	Optional. Three-character ID prefixes indicating the desired article types. You can specify multiple values for this parameter in a single REST call, by repeating the parameter name for each value. For example, `articleTypes=ka0&articleTypes=ka1`.
categories	Optional. The name of the data category group and name of the data category for desired articles, expressed as a JSON mapping. You can specify multiple data category group and data category pairs in this parameter. For example, `categories={"Regions":"Asia","Products":"Laptops"}`. Characters in the URL might need to be encoded. For this example, `categories=%7B%22Regions%22%3A%22Asia %22%2C%22Products%22%3A%22Laptops%22%7D`.
channel	Optional. The channel where the matching articles are visible. Valid values: • `AllChannels`–Visible in all channels the user has access to • `App`–Visible in the internal Salesforce Knowledge application • `Pkb`–Visible in the public knowledge base • `Csp`–Visible in the Customer Portal • `Prm`–Visible in the Partner Portal If `channel` isn't specified, the default value is determined by the type of user. • `Pkb` for a guest user • `Csp` for a Customer Portal user • `Prm` for a Partner Portal user • `App` for any other type of user If `channel` is specified, the specified value may not be the actual value requested, because of certain requirements.

Parameter	Description
	• For guest, Customer Portal, and Partner Portal users, the specified value must match the default value for each user type. If the values don't match or `AllChannels` is specified, then `App` replaces the specified value.
	• For all users other than guest, Customer Portal, and Partner Portal users:
	– If `Pkb`, `Csp`, `Prm`, or `App` are specified, then the specified value is used.
	– If `AllChannels` is specified, then `App` replaces the specified value.
language	Required. The language of the user's query. Specifies the language that matching articles are written in.
limit	Optional. Specifies the maximum number of articles to return. If there are more suggested articles than the limit specified, the response body's `hasMoreResults` property is `true`.
publishStatus	Required. The article's publication status. Valid values:
	• `Draft`—Articles aren't published in Salesforce Knowledge.
	• `Online`—Articles are published in Salesforce Knowledge.
	• `Archived`—Articles aren't published and are available in Archived Articles view.
q	Required. The user's search query string, properly URL-encoded. Suggestions are returned only if the user's query string meets the minimum length requirements: one character for queries in Chinese, Japanese, and Korean, and three characters for all other languages. Query strings exceeding the maximum length of 250 characters return an error.
topics	Optional. The topic of the returned articles. For example: `topics=outlook&topics=email`.
validationStatus	Optional. The validation status of returned articles.

Example

Example Request

```
curl
https://MyDomainName.my.salesforce.com/services/data/v66.0/search/suggestTitleMatches?
q=orange+banana&language=en_US&publishStatus=Online -H "Authorization: Bearer token"
```

Example Response Body

```
{
  "autoSuggestResults" : [ {
    "attributes" : {
    "type" : "KnowledgeArticleVersion",
    "url" : "/services/data/v66.0/sobjects/KnowledgeArticleVersion/ka0D00000004CcQ"
    },
  "Id" : "ka0D00000004CcQ",
  "UrlName" : "orange-banana",
```

```
  "Title" : "orange banana",
  "KnowledgeArticleId" : "kA0D00000004Cfz"
  } ],
  "hasMoreResults" : false
}
```

Search Suggested Queries

Returns a list of suggested searches based on the user's query string text matching searches that other users have performed in Salesforce Knowledge. Provides a way to improve search effectiveness, before the user performs a search. This resource is available in REST API version 30.0 and later.

Salesforce Knowledge must be enabled in your organization.

Queries are suggested if they exactly match the query string text. The text string must be a prefix within the query; it's not considered a match if it appears within a word. For example, the text string `app` would return suggested queries *apple banana* and *banana apples* but not *pineapple*.

If the number of suggestions returned exceeds the limit specified in the request, the end of the response contains a field called `hasMoreResults`. Its value is `true` if the suggestions returned are only a subset of the suggestions available, and `false` otherwise.

If the user's search query contains quotation marks or wildcards, those symbols are automatically removed from the query string in the URI.

Syntax

URI

/services/data/v*XX.X*/search/suggestSearchQueries?q=*searchString*
&language=*languageOfQuery*

Formats
JSON, XML

HTTP methods
GET

Authentication
Authorization: Bearer *token*

Request body
None required

Request parameters

Parameter	Description
channel	Optional. Specifies the Salesforce Knowledge channel where the article can be viewed. Valid values:
	• `AllChannels`–Visible in all channels the user has access to
	• `App`–Visible in the internal Salesforce Knowledge application
	• `Pkb`–Visible in the public knowledge base
	• `Csp`–Visible in the Customer Portal
	• `Prm`–Visible in the Partner Portal

Parameter	Description
	If `channel` isn't specified, the default value is determined by the type of user.
	• `Pkb` for a guest user
	• `Csp` for a Customer Portal user
	• `Prm` for a Partner Portal user
	• `App` for any other type of user
	If `channel` is specified, the specified value may not be the actual value requested, because of certain requirements.
	• For guest, Customer Portal, and Partner Portal users, the specified value must match the default value for each user type. If the values don't match or `AllChannels` is specified, then `App` replaces the specified value.
	• For all users other than guest, Customer Portal, and Partner Portal users:
	– If `Pkb`, `Csp`, `Prm`, or `App` are specified, then the specified value is used.
	– If `AllChannels` is specified, then `App` replaces the specified value.
`language`	Required. The language of the user's query.
`limit`	Optional. Specifies the maximum number of suggested searches to return. If there are more suggested queries than the limit specified, the response body's `hasMoreResults` property is `true`.
`q`	Required. The user's search query string, properly URL-encoded. Suggestions are returned only if the user's query string meets the minimum length requirements: one character for queries in Chinese, Japanese, and Korean, and three characters for all other languages. Query strings exceeding the maximum length of 250 characters return an error.

Example

Example Request

```
curl
https://MyDomainName.my.salesforce.com/services/data/v66.0/search/suggestSearchQueries?
q=app&language=en_US -H "Authorization: Bearer token"
```

Example Response Body

```
{
  "autoSuggestResults" : [ {
    "0" : "apple",
    "1" : "apple banana",
  } ],
  "hasMoreResults" : false
}
```

Support Knowledge with REST API

Knowledge Support REST APIs allow both authorized and guest users to retrieve the user's visible data categories and their associated articles. This resource is available in REST API version 38.0 and later.

Authenticated users need the `UserProfile.apiEnabled` permission, Knowledge enabled in the organization, read rights on the article type, and any other knowledge specific permission or preference that controls visibility to articles.

Guest users need the `Guest Access to the Support API` preference enabled on the relevant Site, Knowledge enabled in the organization, and read rights on the article type and article channel that controls the visibility for guest users.

Syntax

URI
    `/services/data/v**XX.X**/support`

Method
    GET

Formats
    JSON, XML

Authentication
    `Authorization: Bearer **token**`

Example

Example Response Body

```
{
"dataCategoryGroups" : "/services/data/vXX.X/support/dataCategoryGroups",
"knowledgeArticles" : "/services/data/vXX.X/support/knowledgeArticles"
:
}
```

IN THIS SECTION:

[Data Category Groups](#)

Get data category groups that are visible to the current user. This resource is available in REST API version 38.0 and later.

[Data Category Detail](#)

Gets data category details and the child categories by a given category. This resource can be used in API version 38.0 and later.

[Articles List](#)

Get a page of online articles for the given language and category through either search or query. This resource is available in REST API version 38.0 and later.

[Articles Details](#)

Gets all online article fields, accessible to the user. This resource is available with article IDs in REST API version 38.0 and later, and this resource is available with article URL names in version 44.0 and later.

Data Category Groups

Get data category groups that are visible to the current user. This resource is available in REST API version 38.0 and later.

Salesforce Knowledge must be enabled in your organization. This resource can be used in API version 38.0 and later. Use the language code format used in *Which Languages Does Salesforce Support?*.

Only the user's visible data categories are returned. A user might be able to see several sub trees in the category group, therefore, the top categories that are visible to the user in each group are returned.

Syntax

URI

`/services/data/v`**`XX.X`**`/support/dataCategoryGroups`

Method

GET

Formats

JSON, XML

Authentication

`Authorization: Bearer` **`token`**

HTTP headers

Accept: Optional. Can be either `application/json` or `application/xml`.

Accept-language: Optional. Language to translate the categories. Any ISO-639 language abbreviation, and an ISO-3166 country code subtag in the HTTP Accept-Language header. Only one language accepted. If no language specified, the non-translated labels are returned.

Input:

string `sObjectName`: Required. `KnowledgeArticleVersion` only.

boolean `topCategoriesOnly`: Optional. Defaults to true

- True returns only the top level categories.
- False returns the entire tree.

> **Note:** All the input parameters are case-sensitive.

Output:

A list of the active data category groups that are visible to the current user in the site context. Returns id, name, label, and their top level categories or the entire data category group tree that are visible to the current user. The labels must be translated to the given language if they are available.

- **Data Category Group List**

  This payload lists the active root Data Category Groups that can be used in other requests to return the data categories and articles related to it.

  ```
  {
      "categoryGroups": [ Data Category Group, ....],
  }
  ```

  > **Note:** Returns only the active groups that are associated to the given entity (by `sObjectName`). Only `KnowledgeArticleVersion` is supported.

- **Data Category Group**

    This represents an individual data category group, and its root category.

    ```
    {
        "name": String,  // the unique name of the category group
        "label": String,  // returns the translated version if it is available
        "objectUsage" : String, // currently only "KnowledgeArticleVersion" is available.

        "topCategories": [ Data Category Summary, ....]
    }
    ```

- **Data Category Summary**

    This provides a summary of data category information. The Summary and Detail responses share common properties, with the goal of providing only as much information as is necessary from associated resources.

    ```
    {
        "name": String, // the unique name of the category
        "label":  String,  // returns the translated version if it is available
        "url":  URL,     // the url points to the data category detail API
     "childCategories": [ Data Category Summary, ....] // null if topCategoriesOnly is
    true
    }
    ```

    > 📝 Note: The URL property is a pre-calculated path to the unique resource representing this data category, in this case it is a Data Category Detail API.

Example

Example Request

```
curl
https://MyDomainName.my.salesforce.com/services/data/v66.0/support/dataCategoryGroups?sObjectName=KnowledgeArticleVersion
 -H "Authorization: Bearer token"
```

Example Response Body

```
{
  "categoryGroups" : [ {
"label" : "Doc",
"name" : "Doc",
"objectUsage" : "KnowledgeArticleVersion",
"topCategories" : [ {
  "childCategories" : null,
  "label" : "All",
  "name" : "All",
  "url" :
"/services/data/v66.0/support/dataCategoryGroups/Doc/dataCategories/All?sObjectName=KnowledgeArticleVersion"

} ]
 }, {
"label" : "Manual",
"name" : "Manual",
"objectUsage" : "KnowledgeArticleVersion",
"topCategories" : [ {
```

```
    "childCategories" : null,
    "label" : "All",
    "name" : "All",
    "url" :
"/services/data/v66.0/support/dataCategoryGroups/Manual/dataCategories/All?sObjectName=KnowledgeArticleVersion"

  } ]
   } ]
}
```

Data Category Detail

Gets data category details and the child categories by a given category. This resource can be used in API version 38.0 and later.

Salesforce Knowledge must be enabled in your organization. Use the language code format used in *Which Languages Does Salesforce Support?*.

Syntax

URI

```
/services/data/vXX.X/support/dataCategoryGroups/group/dataCategories/category
```

Method

GET

Formats

JSON, XML

Authentication

```
Authorization: Bearer token
```

HTTP headers

Accept: Optional. Can be either `application/json` or `application/xml`.

Accept-language: Optional. Language to translate the categories. Any ISO-639 language abbreviation, and an ISO-3166 country code subtag in the HTTP Accept-Language header. Only one language accepted. If no language specified, the non-translated labels are returned.

Input:

string `sObjectName`: Required. `KnowledgeArticleVersion` only.

Output:

Details of the category and a list of child categories (name, label, etc.).

- **Data Category Detail**

  Used for situations where the hierarchical representation of data categories is important. The child property contains a list of child data categories.

  ```
  {
      "name": String, // the unique name of the category
      "label":  String, // returns the translated version if it is available
      "url":  URL,
      "childCategories": [ Data Category Summary, ....],
  }
  ```

📝 **Note:** If the category isn't visible to the current user the return is empty.

Example

Example Request

```
curl
https://MyDomainName.my.salesforce.com/services/data/v66.0/support/dataCategoryGroups/Doc/dataCategories/All?sObjectName=KnowledgeArticleVersion
 -H "Authorization: Bearer token"
```

Example Response Body

```
{
  "childCategories" : [ {
    "childCategories" : null,
    "label" : "Help",
    "name" : "Help",
    "url" :
"/services/data/v66.0/support/dataCategoryGroups/Doc/dataCategories/Help?sObjectName=KnowledgeArticleVersion"

  }, {
    "childCategories" : null,
    "label" : "QA",
    "name" : "QA",
    "url" :
"/services/data/v66.0/support/dataCategoryGroups/Doc/dataCategories/QA?sObjectName=KnowledgeArticleVersion"

  } ],
  "label" : "All",
  "name" : "All",
  "url" :
"/services/data/v66.0/support/dataCategoryGroups/Doc/dataCategories/All?sObjectName=KnowledgeArticleVersion"
}
```

Articles List

Get a page of online articles for the given language and category through either search or query. This resource is available in REST API version 38.0 and later.

Syntax

URI

/services/data/v**XX.X**/support/knowledgeArticles

Method

GET

Formats

JSON, XML

Authentication

Authorization: Bearer **token**

HTTP headers

Accept: Optional. Can be either `application/json` or `application/xml`.

Accept-language: Required. The article must be an active language in the user's organization

- If the language code isn't valid, an error message is returned: "The language code is not valid or not supported by Knowledge."
- If the language code is valid, but not supported by Knowledge, then an error message is returned: "Invalid language code. Check that the language is included in your Knowledge language settings."

Input:

string `q`: Optional, Performs an SOSL search. If the query string is null, empty, or not given, an SOQL query runs.

The characters `?` and `*` are used for wildcard searches. The characters `(`, `)`, and `"` are used for complex search terms. See https://developer.salesforce.com/docs/atlas.en-us.soql_sosl.meta/soql_sosl/sforce_api_calls_sosl_find.htm.

string `channel`: Optional, defaults to user's context. For information on channel values, see Valid `channel` values.

- **App**: Visible in the internal Salesforce Knowledge application
- **Pkb**: Visible in the public knowledge base
- **Csp**: Visible in the Customer Portal
- **Prm**: Visible in the Partner Portal

string `categories` in map json format {"group1":"category1","group2":"category2",...})

Optional, defaults to None. Category group must be unique in each group:category pair, otherwise you get `ARGUMENT_OBJECT_PARSE_ERROR`. There is a limit of three data category conditions, otherwise you get `INVALID_FILTER_VALUE`.

string `queryMethod` values are: `AT, BELOW, ABOVE, ABOVE_OR_BELOW`. Only valid when categories are specified, defaults to ABOVE_OR_BELOW.

string `sort`: Optional, a sortable field name `LastPublishedDate, CreatedDate, Title, ViewScore`. Defaults to `LastPublishedDate` for query and relevance for search.

> 📝 Note: When sorting on `ViewScore` it is only available for query, not search, and no pagination is supported. You only get one page of results.

string `order`: Optional, either ASC or DESC, defaults to DESC. Valid only when sort is valid.

integer `pageSize`: Optional, defaults to 20. Valid range 1 to 100.

integer `pageNumber` : Optional, defaults to 1.

Output:

A page of online articles in the given language and category visible to the current user.

- **Article Page**

  A page of articles. The individual entries are article summaries so the size can be kept at a minimum.

  ```
  {
     "articles": [ Article Summary, … ],  // list of articles
        "currentPageUrl": URL,   // the article list API with current page number
          "nextPageUrl": URL,   // the article list API with next page number,
        which can be null if there are no more articles.
        "pageNumber": Int  // the current page number, starting at 1.
      }
  ```

  > 📝 Note: The API supports paging. Each page of responses includes a URL to its page, as well as the URL to the next page of articles.

> **Note:** if the user input parameter has the default value, the API does not show this parameter in either
> `currentPageUrl` or `nextPageUrl`.

- **Article Summary**

  A summary of an article used in a list of article responses. It shares similar properties to the Article Detail representation, as one
  is a superset of the other.

```
{
      "id":  Id,     // articleId
      "articleNumber": String,
      "articleType": String,  // apiName of the article type, available in API v44.0
 and later
      "title":  String,
      "urlName": String,  // available in API v44.0 and later
      "summary":  String,
      "url":  URL,  // to the Article Detail API
      "viewCount": Int,        // view count in the interested channel
      "viewScore": double (in xxx.xxxx precision),   // view score in the interested
 channel.
      "upVoteCount": int,  // up vote count in the interested channel.
      "downVoteCount": int, // down vote count in the interested channel.
      "lastPublishedDate": Date // last publish date in ISO8601 format
      "categoryGroups":  [ Data Category Group, …. ]}
```

The "url" property always points to the Article Details resource endpoint. For information on valid channel values, see the channel
parameter description

- **Data Category Group**

  An individual data category group, its root category, and a list of selected data categories in the group.

```
{
      "groupName": String,  // the unique name of the category group
      "groupLabel": String,  // returns the translated version
      "selectedCategories": [ Data Category Summary, … ]
    }
```

- **Data Category Summary**

  Provides a summary of data category information. The Summary and Detail responses share common properties.

```
{
            "categoryName": String, // the unique name of the category
            "categoryLabel":  String, // returns the translated version, per the API
 language specified
            "url":  String // returns the url for the DataCategory REST API.
    }
```

> **Note:** The outputs of Data Category Group and Data Category Summary in Article List API are different from the Data
> Category Groups API.

Example

Example Request

```
curl
https://MyDomainName.my.salesforce.com/services/data/v66.0/support/knowledgeArticles?sort=ViewScore&channel=Pkb&pageSize=3

    HTTP Headers:
        Content-Type: application/json; charset=UTF-8
        Accept: application/json
        Accept-Language: en-US
```

Example Response Body

```
{
 "articles" : [ {
 "articleNumber" : "000001002",
 "categoryGroups" : [ ],
 "downVoteCount" : 0,
 "id" : "kA0xx000000000BCAQ",
 "lastPublishedDate" : "2015-02-25T02:07:18Z",
 "summary" : "With this online Chinese input tool, you can type Chinese characters
through your web browser without installing any Chinese input software in your system.
 The Chinese online input tool uses the popular Pin Yin input method. It is a fast and
 convenient tool to input Chinese on English OS environments.",
 "title" : "Long text test",
 "upVoteCount" : 0,
 "url" : "/services/data/v66.0/support/knowledgeArticles/kA0xx000000000BCAQ",
 "viewCount" : 4,
 "viewScore" : 100.0
 }, {
 "articleNumber" : "000001004",
 "categoryGroups" : [ ],
 "downVoteCount" : 0,
 "id" : "kA0xx000000000LCAQ",
 "lastPublishedDate" : "2016-06-21T21:11:02Z",
 "summary" : "The number of characters required for complete coverage of all these
languages' needs cannot fit in the 256-character code space of 8-bit character encodings,
 requiring at least a 16-bit fixed width encoding or multi-byte variable-length encodings.
 \r\n\r\nAlthough CJK encodings have common character sets, the encodings often used to
 represent them have been developed separately by different East Asian governments and
 software companies, and are mutually incompatible. Unicode has attempted, with some
controversy, to unify the character sets in a process known as Han unification.\r\n\r\nCJK
 character encodings should consist minimally of Han characters p",
 "title" : "Test Images",
 "upVoteCount" : 0,
 "url" : "/services/data/v66.0/support/knowledgeArticles/kA0xx000000000LCAQ",
 "viewCount" : 0,
 "viewScore" : 0.0
 }, {
 "articleNumber" : "000001012",
 "categoryGroups" : [ ],
 "downVoteCount" : 0,
 "id" : "kA0xx000000006GCAQ",
 "lastPublishedDate" : "2016-06-21T21:10:48Z",
```

```
"summary" : null,
"title" : "Test Draft 2",
"upVoteCount" : 0,
"url" : "/services/data/v66.0/support/knowledgeArticles/kA0xx000000006GCAQ",
"viewCount" : 0,
"viewScore" : 0.0
} ],
"currentPageUrl" :
"/services/data/v66.0/support/knowledgeArticles?channel=Pkb&amp;pageSize=3&amp;sort=ViewScore",
"nextPageUrl" : null,
"pageNumber" : 1
}
```

Usage

Salesforce Knowledge must be enabled in your organization. This resource can be used in API version 38.0 and later. The Custom File Field is not supported because it returns a link to a binary stream. Use the language code format used in *Which Languages Does Salesforce Support?*.

Valid `channel` Values

- When using the options string `channel`, where the matching articles are visible, the following values are valid.

  - `App`–Visible in the internal Salesforce Knowledge application

  - `Pkb`–Visible in the public knowledge base

  - `Csp`–Visible in the Customer Portal

  - `Prm`–Visible in the Partner Portal

- If `channel` isn't specified, the default value is determined by the type of user.

  - `Pkb` for a guest user

  - `Csp` for a Customer Portal user

  - `Prm` for a Partner Portal user

  - `App` for any other type of user

- If `channel` is specified, the specified value may be used to retrieve articles.

  - For guest, Customer Portal, and Partner Portal users, if the specified channel is other than the channel accessible to the user, an error is returned.

  - For all users other than guest, Customer Portal, and Partner Portal users, the specified channel value is used.

Articles Details

Gets all online article fields, accessible to the user. This resource is available with article IDs in REST API version 38.0 and later, and this resource is available with article URL names in version 44.0 and later.

Salesforce Knowledge must be enabled in your organization. This resource can be used in API version 38.0 and later. The Custom File Field is not supported because it returns a link to a binary stream. Use the language code format used in *Which Languages Does Salesforce Support?*.

A lookup custom field is visible to guest users depending on the lookup entity type. For example, User is visible, but Case and Account are not visible. Following standard fields are not visible to a guest user, even if they are in the layout:

- archivedBy
- isLatestVersion
- translationCompletedDate
- translationImportedDate
- translationExportedDate
- versionNumber
- visibleInInternalApp
- visibleInPKB
- visibleToCustomer
- visbileToPartner

Valid `channel` Values

- When using the options string `channel`, where the matching articles are visible, the following values are valid.

    - `App`–Visible in the internal Salesforce Knowledge application
    - `Pkb`–Visible in the public knowledge base
    - `Csp`–Visible in the Customer Portal
    - `Prm`–Visible in the Partner Portal

- If `channel` isn't specified, the default value is determined by the type of user.

    - `Pkb` for a guest user
    - `Csp` for a Customer Portal user
    - `Prm` for a Partner Portal user
    - `App` for any other type of user

- If `channel` is specified, the specified value may be used to retrieve articles.

    - For guest, Customer Portal, and Partner Portal users, if the specified channel is other than the channel accessible to the user, an error is returned.
    - For all users other than guest, Customer Portal, and Partner Portal users, the specified channel value is used.

Syntax

Method
GET

Formats
JSON, XML

Authentication
`Authorization: Bearer `***`token`***

Endpoint
`/services/data/v`***`XX.X`***`/support/knowledgeArticles/`***`articleId_or_articleUrlName`***

HTTP headers

Accept: Optional. Can be either `application/json` or `application/xml`.

Accept-language: Required. The article must be an active language in the user's organization

- If the language code isn't valid, an error message is returned: "The language code is not valid or not supported by Knowledge."

- If the language code is valid, but not supported by Knowledge, then an error message is returned: "Invalid language code. Check that the language is included in your Knowledge language settings."

Input:

string `channel`: Optional, defaults to user's context. For information on channel values, see Valid `channel` Values.

- **App**: Visible in the internal Salesforce Knowledge application

- **Pkb**: Visible in the public knowledge base

- **Csp**: Visible in the Customer Portal

- **Prm**: Visible in the Partner Portal

boolean `updateViewStat`: Optional, defaults to true. If true, API updates the view count in the given channel as well as the total view count.

boolean `isUrlName`: Optional, defaults to false. If true, indicates that the last portion of the endpoint is a URL name instead of an article ID. Available in API v44.0 and later

Output:

The detailed fields of the article, if the article is online and visible to the current user.

- **Article Detail**

  Full detail of an article, with complete metadata and layout-driven fields used for display of an article. It includes all the same properties as an Article Summary representation.

```
{
        "id":  Id,     // articleId,
        "articleNumber": String,
        "articleType": String,  // apiName of the article type, available in API
v44.0 and later
        "title":  String,
        "urlName": String,    // available in API v44.0 and later
        "summary":  String,
        "url":  URL,
        "versionNumber": Int,
        "createdDate": Date,  // in ISO8601 format
        "createdBy": User Summary on page 126,
        "lastModifiedDate": Date,    // in ISO8601 format
        "lastModifiedBy": User Summary on page 126,
        "lastPublishedDate": Date, // in ISO8601 format
      "layoutItems": [ Article Field, ... ],  // standard and custom fields visible
 to the user, sorted based on the layouts of the article type.
        "categories": [ Data Category Groups, ... ],
        "appUpVoteCount": Int,
        "cspUpVoteCount": Int,
        "prmUpVoteCount": Int,
        "pkbUpVoteCount": Int,
        "appDownVoteCount": Int,
        "cspDownVoteCount": Int,
        "prmDownVoteCount": Int,
```

```
                    "pkbDownVoteCount": Int,
                    "allViewCount": Int,
                    "appViewCount": Int,
                    "cspViewCount": Int,
                    "prmViewCount": Int,
                    "pkbViewCount": Int,
                    "allViewScore": Double,
                    "appViewScore": Double,
                    "cspViewScore": Double,
                    "prmViewScore": Double,
                    "pkbViewScore": Double
                    }
```

- **User Summary**

```
{
        "id": String
        "isActive": boolean    // true/false
        "userName": String   // login name
        "firstName": String
        "lastName": String
        "email": String
        "url": String                  // to the chatter user detail url:
/services/data/vXX.X/chatter/users/{userId}, for guest user, it will return null.
        }
```

- **Article Field**

    An individual field of article information, which is listed in an Article Detail in the order required by the administrator's layout.

```
{
        "type":  Enum,          // see the Notes
        "name": String,         // In API v43.0 and earlier, the developer name. In
API v44.0 and later, the API name.
        "label": String,          // label
        "value": String,
        }
```

Example

Example Request

```
curl
https://MyDomainName.my.salesforce.com/services/data/v66.0/support/knowledgeArticles/kA0xx000000000LCAQ

        HTTP Headers:
        Content-Type: application/json; charset=UTF-8
        Accept: application/json
        Accept-Language: en-US
```

Example Response Body

```
{
        "allViewCount" : 17,
        "allViewScore" : 100.0,
```

```
            "appDownVoteCount" : 0,
            "appUpVoteCount" : 0,
            "appViewCount" : 17,
            "appViewScore" : 100.0,
            "articleNumber" : "000001004",
            "categoryGroups" : [ ],
            "createdBy" : {
            "email" : "user@company.com",
            "firstName" : "Test",
            "id" : "005xx000001SvoMAAS",
            "isActive" : true,
            "lastName" : "User",
            "url" : "/services/data/v66.0/chatter/users/005xx000001SvoMAAS",
            "userName" : "admin@salesforce.org"
            },
            "createdDate" : "2016-06-21T21:10:54Z",
            "cspDownVoteCount" : 0,
            "cspUpVoteCount" : 0,
            "cspViewCount" : 0,
            "cspViewScore" : 0.0,
            "id" : "kA0xx000000000LCAQ",
            "lastModifiedBy" : {
            "email" : "user@company.com",
            "firstName" : "Test",
            "id" : "005xx000001SvoMAAS",
            "isActive" : true,
            "lastName" : "User",
            "url" : "/services/data/v66.0/chatter/users/005xx000001SvoMAAS",
            "userName" : "admin@salesforce.org"
            },
            "lastModifiedDate" : "2016-06-21T21:11:02Z",
            "lastPublishedDate" : "2016-06-21T21:11:02Z",
            "layoutItems" : [ {
            "label" : "Out of Date",
            "name" : "IsOutOfDate",
            "type" : "CHECKBOX",
            "value" : "false"
            }, {
            "label" : "sample",
            "name" : "sample",
            "type" : "PICK_LIST",
            "value" : null
            }, {
            "label" : "Language",
            "name" : "Language",
            "type" : "PICK_LIST",
            "value" : "en_US"
            }, {
            "label" : "MyNumber",
            "name" : "MyNumber",
            "type" : "NUMBER",
            "value" : null
            }, {
            "label" : "My File",
```

```
      "name" : "My_File",
      "type" : "FILE",
      "value" : null
      } ],
      "pkbDownVoteCount" : 0,
      "pkbUpVoteCount" : 0,
      "pkbViewCount" : 0,
      "pkbViewScore" : 0.0,
      "prmDownVoteCount" : 0,
      "prmUpVoteCount" : 0,
      "prmViewCount" : 0,
      "prmViewScore" : 0.0,
     "summary" : "The number of characters required for complete coverage of all these
 languages' needs cannot fit in the 256-character code space of 8-bit character encodings,
 requiring at least a 16-bit fixed width encoding or multi-byte variable-length encodings.
 \r\n\r\nAlthough CJK encodings have common character sets, the encodings often used to
 represent them have been developed separately by different East Asian governments and
 software companies, and are mutually incompatible. Unicode has attempted, with some
controversy, to unify the character sets in a process known as Han unification.\r\n\r\nCJK
 character encodings should consist minimally of Han characters p",
      "title" : "Test Images",
      "url" : "/services/data/v66.0/support/knowledgeArticles/kA0xx000000000LCAQ",
      "versionNumber" : 7
      }
```

Usage

CHAPTER 4   Salesforce Knowledge Metadata API

You can manage articles, page layouts, and data categories with the Metadata API.

The main purpose of Metadata API is to move metadata between Salesforce orgs during the development process. Use Metadata API to deploy, retrieve, create, update, or delete customization information, such as custom object definitions and page layouts. Metadata API doesn't work directly with business data.

For generic Metadata API information see the Metadata API Developer Guide.

ArticleType

Represents the metadata associated with an article type.

All articles in Salesforce Knowledge are assigned to an *article type*. An article's type determines the type of content it contains, its appearance, and which users can access it. For example, a simple FAQ article type can have two custom fields, `Question` and `Answer`, where article managers enter data when creating or updating FAQ articles. A more complex article type can have dozens of fields organized into several sections. Using layouts and templates, administrators can structure the article type in the most effective way for its particular content. User access to article types is controlled by permissions. For each article type, an administrator can grant "Create," "Read," "Edit," or "Delete" permissions to users. For example, the article manager can allow internal users to read, create, and edit FAQ article types, but let partner users only read FAQs. See "Knowledge Article Types" in the Salesforce online help and Knowledge in the *SOAP API Developer Guide*.

Declarative Metadata File Suffix and Directory Location

An ArticleType is defined as a custom object and is stored in the `objects` folder. ArticleTypes have a suffix `__kav` (instead of `__c` for custom objects). ArticleType field names have a suffix of `__c` like other custom objects, and must be dot-qualified with the name of the article type to which they belong. This is shown in the following sample `package.xml` file:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <fullName>articlefilemetadata</fullName>
    <apiAccessLevel>Unrestricted</apiAccessLevel>
    <types>
        <members>newarticle__kav.description__c</members>
        <name>CustomField</name>
    </types>
    <types>
        <members>newarticle__kav</members>
        <name>CustomObject</name>
    </types>
</Package>
```

Version

ArticleTypes are available in API version 19.0 and later.

Fields

Field Name	Field Type	Description
articleTypeChannel Display	articleTypeChannelDisplay	Represents the article-type templates used to display an article in the various channels. See "Article Type Templates" in the Salesforce online help.
deploymentStatus	DeploymentStatus (enumeration of type string)	A string which represents the deployment status of a custom object or field. Valid values are: • `InDevelopment` • `Deployed`

Field Name	Field Type	Description
description	string	A description of the article type. Maximum of 1000 characters.
fields	CustomField[]	Represents one or more fields in the article type.
gender	Gender	Indicates the gender of the noun that represents the object. This is used for languages where words need different treatment depending on their gender.
label	string	Label that represents the object throughout the Salesforce user interface.
pluralLabel	string	Plural version of the `label` value.
startsWith	StartsWith (enumeration of type string)	Indicates whether the noun starts with a vowel, consonant, or is a special character. This is used for languages where words need different treatment depending on the first character. Valid values are listed in StartsWith.

ArticleTypeChannelDisplay

Determines the article-type templates that are used to display an article in its channels. Unless otherwise noted, all fields are createable, filterable, and nillable.

Field Name	Field Type	Description
articleTypeTemplates	ArticleTypeTemplate on page 131[]	Indicates which article-type template applies in the specified channel.

ArticleTypeTemplate

Sets the article-type template for a specific channel. If not specified, the default article-type template applies.

Field Name	Field Type	Description
channel	string	Specifies the channel where the article-type template applies: • `AllChannels`: all the available channels. • `App`: the Articles tab in Salesforce Knowledge. • `Pkb`: the public knowledge base. • `Csp`: the Customer Portal. • `Prm`: the partner portal.
page	string	Represents the name of the custom Visualforce page used as a custom article-type template. Use this field when you select `Page` in the template field.
template	string	Indicates the article-type template used for the specified channel: • `Page`: custom Visualforce page. When specifying this value, you must also set the `page` field with the Visualforce page name. • `Tab`: display the sections you defined in the layout as tabs.

Field Name	Field Type	Description
		• `Toc`: display the sections you defined in the layout as table of content.

Declarative Metadata Sample Definitions

A sample article type definition follows:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
    <articleTypeChannelDisplay>
        <articleTypeTemplates>
            <channel>App</channel>
            <template>Tab</template>
        </articleTypeTemplates>
        <articleTypeTemplates>
            <channel>Prm</channel>
            <template>Tab</template>
        </articleTypeTemplates>
        <articleTypeTemplates>
            <channel>Csp</channel>
            <template>Tab</template>
        </articleTypeTemplates>
        <articleTypeTemplates>
            <channel>Pkb</channel>
            <template>Toc</template>
        </articleTypeTemplates>
    </articleTypeChannelDisplay>
    <deploymentStatus>Deployed</deploymentStatus>
    <description>Article type with custom fields</description>
    <fields>
        <fullName>description__c</fullName>
        <label>Description</label>
        <length>48</length>
        <type>Text</type>
    </fields>
    <label>newarticle</label>
    <pluralLabel>newarticles</pluralLabel>
</CustomObject>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

IN THIS SECTION:

ArticleType Layout

Represents the metadata associated with an article type page layout. Article type layouts determine which fields users can view and edit when entering data for an article. Article type layouts also determine which sections appear when users view articles.

ChannelLayout

Represents the metadata associated with a communication channel layout. Communication channel layouts let admins share article content inline into communication channels (for example, in email publishers, Experience Builder sites, or social media publishers). Admins can create a list of fields for an article type or record type that they want to share for each communication channel. You can customize the order of the fields.

ArticleType CustomField

Represents the metadata associated with an article type custom field. Use this metadata type to create, update, or delete article type custom field definitions.

ArticleType Layout

Represents the metadata associated with an article type page layout. Article type layouts determine which fields users can view and edit when entering data for an article. Article type layouts also determine which sections appear when users view articles.

The format of the article, for example whether layout sections display as subtabs or as a single page with links, is defined by the article-type template. Each article type has only one layout, but you can choose a different template for each of the article type's four channels. See Knowledge in *SOAP API Developer Guide*.

File Suffix and Directory Location

ArticleType layouts are stored in the `layouts` directory of the corresponding package directory. The prefix must match with the article type API name. The extension is `.layout`.

Version

ArticleType layouts are available in API version 19.0 and later.

Fields

Field Name	Field Type	Description
layoutSections	LayoutSection[]	The main sections of the layout containing the article fields. The order here determines the layout order.

LayoutSection

LayoutSection represents a section of an ArticleType layout.

Field Name	Field Type	Description
customLabel	boolean	Indicates if this section's label is custom or standard (built-in). Custom labels can be any text, but must be translated. Standard labels have a predefined set of valid values, for example 'System Information', which are automatically translated.
label	string	The label; either standard or custom, based on the `customLabel` flag.

Field Name	Field Type	Description
layoutColumns	LayoutColumn[]	The columns of the layout, depending on the style. Salesforce Knowledge only supports one column in article type layouts.
style	LayoutSectionStyle (enumeration of type string)	The style of the layout. Salesforce Knowledge only supports the value OneColumn, which displays a one-column page.

LayoutColumn

LayoutColumn represents the items in a column within a layout section.

Field Name	Field Type	Description
layoutItems	LayoutItem[]	The individual items within a column (ordered from top to bottom).

LayoutItem

LayoutItem represents the valid values that define a layout item.

Field Name	Field Type	Description
field	string	The field name reference, for example MyField__c.

Declarative Metadata Sample Definition

The following is the definition of an ArticleType page layout:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Layout xmlns="http://soap.sforce.com/2006/04/metadata">
    <layoutSections>
        <customLabel>true</customLabel>
        <label>Description</label>
        <layoutColumns>
            <layoutItems>
                <field>description__c</field>
            </layoutItems>
            <layoutItems>
                <field>dateTime__c</field>
            </layoutItems>
        </layoutColumns>
        <style>OneColumn</style>
    </layoutSections>
    <layoutSections>
        <label>Data Sheet</label>
        <layoutColumns>
            <layoutItems>
                <field>file__c</field>
            </layoutItems>
        </layoutColumns>
```

```
        <style>OneColumn</style>
    </layoutSections>
</Layout>
```

ChannelLayout

Represents the metadata associated with a communication channel layout. Communication channel layouts let admins share article content inline into communication channels (for example, in email publishers, Experience Builder sites, or social media publishers). Admins can create a list of fields for an article type or record type that they want to share for each communication channel. You can customize the order of the fields.

File Suffix and Directory Location

Channel layout components have the suffix `.channelLayout` and are stored in the `channelLayouts` folder of the corresponding package directory. The prefix must match with the article type API name. In Lightning Knowledge, the prefix must match the API name for the knowledge object.

Version

Channel layout components are available in API version 32.0 and later.

Fields

Field Name	Field Type	Description
doesExcludeFieldLabels	boolean	Indicates whether field labels are excluded from the field contents in the communication channels where this layout applies (`true`) or not (`false`). The default is `false`, meaning field labels are inserted. Available when Lightning Knowledge is enabled in API version 48.0 and later.
doesExcludeFiles	boolean	Indicates whether related files are left off emails (`true`) or attached to emails (`false`). The default is `false`, meaning related files are attached. Available when Lightning Knowledge is enabled in API version 48.0 and later.
enabledChannels	string[]	The communication channels where this layout applies. In API version 32.0 to 46.0, the only valid value is `Email`. When Lightning Knowledge is enabled in API version 47.0 and later, `Chat`, `Messaging`, and `Social` are added valid values.
label	string	Required. The label for this configuration.
layoutItems	ChannelLayoutItem on page 136[]	The article fields contained in the layout. The order here determines the field order.
recordType	string	The name of the record type that the channel layout applies to. The default is the primary record type. Available in API version 41.0 and later.

ChannelLayoutItem

Field Name	Field Type	Description
field	string	Required. Name of the field. The format is *ArticleTypeName.FieldName* or, in Lightning Knowledge, *KnowledgeBaseName.FieldName*.

Declarative Metadata Sample Definition

The following is an example of a ChannelLayout component.

```
<?xml version="1.0" encoding="UTF-8"?>
<ChannelLayout xmlns="http://soap.sforce.com/2006/04/metadata">
    <label>Layout for Email</label>
    <layoutItems>
        <field>Knowledge.Question</field>
    </layoutItems>
    <layoutItems>
        <field>Knowledge.Answer</field>
    </layoutItems>
    <enabledChannels>Email</enabledChannels>
    <enabledChannels>Social</enabledChannels>
    <enabledChannels>Chat</enabledChannels>
    <doesExcludeFiles>false</doesExcludeFiles>
    <doesExcludeFieldLabels>true</doesExcludeFieldLabels>
</ChannelLayout>
```

The following is an example `package.xml` that references the previous definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>*</members>
        <name>ChannelLayout</name>
    </types>
    <version>41.0</version>
</Package>
```

ArticleType CustomField

Represents the metadata associated with an article type custom field. Use this metadata type to create, update, or delete article type custom field definitions.

This type extends the Metadata metadata type and inherits its `fullName` field.

Always specify the full name whenever you create or update a custom field. For example, a custom field on a custom object:

```
MyArticleType__kav.MyCustomField__c
```

Declarative Metadata File Suffix and Directory Location

Custom fields are defined as part of the article type. ArticleType field names have a suffix of __c like other custom objects, and must be dot-qualified with the name of the article type to which they belong. See ArticleType for more information.

Retrieving Custom Fields on Custom or Standard Objects

When you retrieve a custom or standard object, you return everything associated with the object. However, you can also retrieve only the custom fields for an object by explicitly naming the object and fields in `package.xml`. The following definition in `package.xml` retrieves the files `objects/MyCustomObject__c.object`, `objects/Account.object__c.object`, and `objects/MyArticleType__kav.object`, each containing one custom field definition.

```
<types>
    <members>MyCustomObject__c.MyCustomField__c</members>
    <members>Account.MyCustomAccountField__c</members>
    <members>MyArticleType__kav.MyOtherCustomField__c</members>
    <name>CustomField</name>
</types>
```

Version

ArticleTypes custom fields are available in API version 19.0 and later.

Fields for ArticleType

Unless otherwise noted, all fields are createable, filterable, and nillable.

> **Note:** If you create a knowledge validation rule, the errors always display at the top of the page, even if you add it beside the field. Therefore, write the errors descriptively so authors know how to satisfy the validation rule. For example, identify which field is causing the error. The Salesforce Classic user interface does not support field level error messages for articles.

Field Name	Field Type	Description
defaultValue	string	If specified, represents the default value of the field. This field was deprecated in API version 48.0.
deleteConstraint	Metadata Field Types (enumeration of type string)	Provides deletion options for lookup relationships. Valid values are: • `Cascade`—Deletes the lookup record as well as associated lookup fields. • `Restrict`—Prevents the record from being deleted if it's in a lookup relationship. • `SetNull`—This is the default. If the lookup record is deleted, the lookup field is cleared. For more information on lookup relationships, see "Object Relationships" in Salesforce Help.
description	string	Description of the field.

Field Name	Field Type	Description
formula	string	If specified, represents a formula on the field.
formulaTreatBlankAs	Metadata Field Types (enumeration of type string)	Indicates how to treat blanks in a formula. Valid values are: `BlankAsBlank` and `BlankAsZero`.
fullName	string	Inherited from Metadata, this field is defined in the WSDL for this metadata type. It must be specified when creating, updating, or deleting. See `createMetadata()` to see an example of this field specified for a call.
		This value cannot be `null`.
inlineHelpText	string	Represents the content of field-level help. For more information, see "Define Field-Level Help" in Salesforce Help.
label	string	Label for the field. You cannot update the label for standard fields in Article Type such as Title, UrlName, Summary, etc.
length	int	Length of the field.
picklist	Picklist (Including Dependent Picklist)	(**Deprecated.** Use this field in API version 37.0 and earlier only. In later versions, use `valueSet` instead.) If specified, the field is a picklist, and this field enumerates the picklist values and labels.
referenceTo	string	If specified, indicates a reference this field has to another object.
relationshipLabel	string	Label for the relationship.
relationshipName	string	If specified, indicates the value for one-to-many relationships. For example, in the object MyObject that had a relationship to YourObject, the relationship name might be YourObjects.
required	boolean	Indicates whether the field requires a value on creation (`true`) or not (`false`).
type	FieldType	Required. Indicates the field type for the field. Valid values are:

For the `type` field:

- `Checkbox` available in version 30.0 and later
- `Currency`
- `ArticleCurrency`
- `Date`
- `DateTime`
- `Email`
- `File`
- `Formula`
- `Html`
- `Lookup`
- `Number`

Field Name	Field Type	Description
		- `Percent`
		- `Phone`
		- `Picklist`
		- `DependentPicklist`
		- `MultiselectPicklist`
		- `Text`
		- `TextArea`
		- `LongTextArea`
		- `URL`
`visibleLines`	int	Indicates the number of lines displayed for the field.

Declarative Metadata Sample Definition

```xml
<?xml version="1.0" encoding="UTF-8"?>
<CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
    <fields>
        <fullName>Comments__c</fullName>
        <description>add your comments about this object here</description>
        <label>Comments</label>
        <length>32000</length>
        <type>LongTextArea</type>
        <visibleLines>30</visibleLines>
    </fields>
</CustomObject>
```

KnowledgeSettings

Represents the metadata used to manage settings for Salesforce Knowledge.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for more details.

File Suffix and Directory Location

KnowledgeSettings values are stored in a single file named `Knowledge.settings` in the `settings` directory. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

KnowledgeSettings is available in API version 27.0 and later.

Fields

Field Name	Field Type	Description
`answers`	KnowledgeAnswerSettings	Represents the metadata used to manage settings for Salesforce Knowledge and Answers.
`cases`	KnowledgeCaseSettings	Represents the metadata used to manage settings for Salesforce Knowledge and Cases.
`defaultLanguage`	string	Required. The default language for Salesforce Knowledge. Use the abbreviation for the language, for example, en_US for United States English.
`enableChatterQuestionKBDeflection`	boolean	Indicates whether tracking for case deflection via Chatter is enabled (`true`) or not (`false`).
`enableCreateEditOnArticlesTab`	boolean	Indicates whether users can create and edit articles on the articles tab (`true`) or not (`false`).
`enableExternalMediaContent`	boolean	Indicates whether connecting to external media is enabled (`true`) or not (`false`).
`enableKbStandardSharing`	boolean	Indicates whether standard Salesforce sharing is enabled (`true`) or not (`false`).
`enableKnowledge`	boolean	Indicates whether Salesforce Knowledge is enabled (`true`) or not (`false`). This field is `false` by default.
`enableKnowledgeAgentContribution`	boolean	Indicates whether a user can create an article from a case (`true`) or not (`false`). (Classic only)
`enableKnowledgeArticleTextHighlights`	boolean	Indicates whether text snippet highlights in Salesforce Knowledge search results are enabled (`true`) or not (`false`). This field is `true` by default. Available in API version 47.0 and later.
`enableKnowledgeAnswersPromotion`	boolean	Indicates whether a user can create an article from a reply (`true`) or not (`false`). (Classic Only)
`enableKnowledgeCaseRL`	boolean	Indicates whether creating a list of cases linked to an article is enabled (`true`) or not (`false`). (Classic Only)
`enableKnowledgeKeywordAutoComplete`	boolean	Indicates whether auto-complete for keywords is enabled (`true`) or not (`false`) when searching Salesforce Knowledge. This field is `true` by default. Available in API version 47.0 and later.

Field Name	Field Type	Description
enableKnowledgeTitleAutoComplete	boolean	Indicates whether auto-complete for article titles is enabled (`true`) or not (`false`) when searching Salesforce Knowledge. This field is `true` by default. Available in API version 47.0 and later.
enableLightningKbAutoLoadRichTextField	boolean	Indicates whether rich text fields are enabled for editing when an article loads in Lightning Knowledge (`true`) or not (`false`). This field is `false` by default. Available in API version 47.0 and later.
enableLightningKnowledge	boolean	Indicates whether Lightning Knowledge is enabled (`true`) or not (`false`).
languages	KnowledgeLanguageSettings	A list of languages enabled for Salesforce Knowledge.
showArticleSummariesCustomerPortal	boolean	Indicates whether article summaries appear in the Customer Portal (`true`) or not (`false`).
showArticleSummariesInternalApp	boolean	Indicates whether article summaries appear in the internal knowledge base (`true`) or not (`false`).
showArticleSummariesPartnerPortal	boolean	Indicates whether article summaries appear in the partner portal (`true`) or not (`false`).
showValidationStatusField	boolean	Indicates whether validation status appears on articles (`true`) or not (`false`).
suggestedArticles	KnowledgeSuggestedArticlesSettings	Represents the metadata used to manage settings for the case fields used to suggest articles for cases. Available in API version 37.0 and later.
votingEnabled	boolean	When `true`, enables users to vote for a product or feature that uses Vote, such as Articles in Knowledge. Available in API version 50.0 and later.

KnowledgeAnswerSettings

Represents the metadata used to manage settings for Salesforce Knowledge and Answers.

Field Name	Field Type	Description
assignTo	string	Specifies the username an article is assigned to from Answers.
defaultArticleType	string	The default article type for articles created from Answers. Uses the API name of the article type.

Field Name	Field Type	Description
`enableArticleCreation`	boolean	Indicates whether users can create articles from Answers (`true`) or not (`false`).

KnowledgeCaseSettings

Represents the metadata used to manage settings for Salesforce Knowledge and Cases.

Field Name	Field Type	Description
`articlePDFCreationProfile`	string	The profile used to create a PDF of an article from Cases.
`articlePublicSharingSites`	[KnowledgeSitesSettings](#)	Represents the metadata used to manage settings for Salesforce Knowledge and Sites.
`articlePublicSharingCommunities`	[KnowledgeSitesSettings](#)	Represents the metadata used to manage settings for Salesforce Knowledge and Experience Cloud sites.
`articlePublicSharingSitesChatterAnswers`	[KnowledgeSitesSettings](#)	Represents the metadata used to manage settings for Salesforce Knowledge and Sites with Chatter Answers.
`assignTo`	string	Specifies the username an article is assigned to from Cases.
`customizationClass`	string	Specifies the Apex class used for customization.
`defaultContributionArticleType`	string	The default article type for articles created from Cases.
`editor`	KnowledgeCaseEditor (enumeration of type string)	Indicates the rich text editor type. Valid values are: • `simple` • `standard`
`enableArticleCreation`	boolean	Indicates whether users can create articles from Cases (`true`) or not (`false`). Controls whether other fields on KnowledgeCaseSettings can be set.
`enableArticlePublicSharingSites`	boolean	Indicates whether articles can be shared via a public site (URL) from Cases (`true`) or not (`false`).
`enableCaseDataCategoryMapping`	boolean	Indicates whether Case Data Category mapping is enabled (`true`) or not (`false`).
`useProfileForPDFCreation`	boolean	Indicates whether a profile is used to create a PDF of an article from Cases (`true`) or not (`false`).

KnowledgeSitesSettings

Represents the metadata used to manage settings for Salesforce Knowledge and Sites.

Field Name	Field Type	Description
site	string[]	Specifies the site used for Salesforce Knowledge and Sites.

KnowledgeLanguageSettings

A list of languages enabled for Salesforce Knowledge. KnowledgeLanguageSettings is available in API version 28.0 and later.

Field Name	Field Type	Description
language	KnowledgeLanguage[]	Represents the metadata used to manage settings for the languages enabled for Salesforce Knowledge.

KnowledgeLanguage

Represents the metadata used to manage settings for the languages enabled for Salesforce Knowledge. KnowledgeLanguage is available in API version 28.0 and later.

Field Name	Field Type	Description
active	boolean	Indicates whether the language is enabled (true) or not (false).
defaultAssignee	string	The default assignee for articles in the language.
defaultAssigneeType	KnowledgeLanguageLookupValueType (enumeration of type string)	Indicates the default assignee type. Valid values are: • User • Queue
defaultReviewer	string	The default reviewer for articles in the language.
defaultReviewerType	KnowledgeLanguageLookupValueType (enumeration of type string)	Indicates the default reviewer type. Valid values are: • User • Queue
name	string	The code for the language name, for example: English is en. See "What languages does Salesforce support?" in the Salesforce Help for a list of supported languages and their codes.

KnowledgeSuggestedArticlesSettings

Represents the metadata used to manage settings for the articles suggested for cases, work orders, and work order line items. The Work Order and Work Order Line Item objects must be enabled in the org to use the associated fields.

Field Name	Field Type	Description
caseFields	KnowledgeCaseFieldsSettings	Represents a list of the case fields used to suggest articles for the case.
useSuggestedArticlesForCase	boolean	Indicates whether case content is used to suggest articles for cases (true) or not (false).
workOrderFields	KnowledgeWorkOrderFieldsSettings	Represents a list of the work order fields used to suggest articles for the work order.
workOrderLineItemFields	KnowledgeWorkOrderLineItemFieldsSettings	Represents a list of the work order line item fields used to suggest articles for the work order line item.

KnowledgeCaseFieldsSettings

Represents a list of the case fields used to suggest articles for the case. Available in API version 37.0 and later.

Field Name	Field Type	Description
field	KnowledgeCaseField[]	Specifies the names of the case fields used to suggest articles for the case.

KnowledgeCaseField

Represents the name of the case field used to suggest articles for the case. Available in API version 37.0 and later.

Field Name	Field Type	Description
name	string	Specifies the name of the case field used to suggest articles for the case.

KnowledgeWorkOrderFieldsSettings

Represents a list of the work order fields used to suggest articles for the work order. Available in API version 39.0 and later.

Field Name	Field Type	Description
field	KnowledgeWorkOrderField[]	Specifies the names of the work order fields used to suggest articles for the work order.

KnowledgeWorkOrderField

Represents the name of the work order field used to suggest articles for the work order. Available in API version 39.0 and later.

Field Name	Field Type	Description
name	string	Specifies the name of the work order field used to suggest articles for the work order.

KnowledgeWorkOrderLineItemFieldsSettings

Represents a list of the work order line item fields used to suggest articles for the work order line item. Available in API version 39.0 and later.

Field Name	Field Type	Description
field	KnowledgeWorkOrderLineItemField[]	Specifies the names of the work order line item fields used to suggest articles for the work order line item.

KnowledgeWorkOrderLineItemField

Represents the name of the work order line item field used to suggest articles for the work order line item. Available in API version 39.0 and later.

Field Name	Field Type	Description
name	string	Specifies the name of the work order line item field used to suggest articles for the work order line item.

Declarative Metadata Sample Definition

This is a sample Knowledge settings file.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<KnowledgeSettings xmlns="http://soap.sforce.com/2006/04/metadata">
    <answers>
        <enableArticleCreation>false</enableArticleCreation>
    </answers>
    <cases>
        <articlePDFCreationProfile>partner portal knowledge
profile</articlePDFCreationProfile>
        <articlePublicSharingSites>
            <site>KnowledgeSite</site>
            <site>PKB2Site</site>
            <site>ChatterAnswersSite</site>
        </articlePublicSharingSites>
        <articlePublicSharingSitesChatterAnswers>
            <site>ChatterAnswersSite</site>
        </articlePublicSharingSitesChatterAnswers>
        <assignTo>testall@kb.org</assignTo>
        <defaultContributionArticleType>Support</defaultContributionArticleType>
        <editor>simple</editor>
        <enableArticleCreation>true</enableArticleCreation>
        <enableArticlePublicSharingSites>true</enableArticlePublicSharingSites>
```

```
            <useProfileForPDFCreation>true</useProfileForPDFCreation>
        </cases>
        <defaultLanguage>ja</defaultLanguage>
        <enableCreateEditOnArticlesTab>true</enableCreateEditOnArticlesTab>
        <enableExternalMediaContent>true</enableExternalMediaContent>
        <enableKnowledge>true</enableKnowledge>
        <showArticleSummariesCustomerPortal>true</showArticleSummariesCustomerPortal>
        <showArticleSummariesInternalApp>true</showArticleSummariesInternalApp>
        <showArticleSummariesPartnerPortal>true</showArticleSummariesPartnerPortal>
        <showValidationStatusField>true</showValidationStatusField>
        <suggestedArticles>
          <caseFields>
             <field>
                <name>Subject</name>
             </field>
             <field>
                <name>SuppliedEmail</name>
             </field>
          </caseFields>
          <useSuggestedArticlesForCase>true</useSuggestedArticlesForCase>
        </suggestedArticles>
</KnowledgeSettings>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

DataCategoryGroup

Represents a data category group.

This type extends the Metadata metadata type and inherits its `fullName` field.

⚠️ Warning: Using Metadata API to deploy category changes from one organization to another permanently removes categories and record categorizations that are not specified in your XML file. Salesforce recommends that you manually create data categories and record associations in an organization from Setup by entering *Data Categories* in the Quick Find box, then selecting **Data Categories** rather than deploying changes from a sandbox to a production organization. For more information, see Usage.

Data category groups are provided to:

- Classify and filter data.
- Share data among users.

Every data category group contains items or data categories that can be organized hierarchically.

The example below shows the `Geography` data category group and its data categories.

```
Geography
    Worldwide
        North America
            United States of America
```

```
        Canada
        Mexico
    Europe
    Asia
```

Note: See "Work with Data Categories" in the Salesforce online help for more information on data category groups, data categories, parent and sub categories.

File Suffix and Directory Location

The file suffix is `.datacategorygroup`. There is one file for each data category group stored in the `datacategorygroups` folder in the corresponding package directory.

Version

Data category groups are available in API version 18.0 and later.

Fields

This metadata type contains the following fields:

Field Name	Field Type	Description
active	boolean	Required. The status of the category group. Indicates whether this category group is active, (`true`), or not active (`false`).
dataCategory	DataCategory on page 147	Required. The top-level category within the data category group.
description	string	The description of the data category group.
fullName	string	Required. The unique name of the data category group. When creating a data category group, the `fullName` field and the file name (without its suffix) must match.The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This field is inherited from the Metadata component.
label	string	Required. Label that represents the object in Salesforce.
objectUsage	ObjectUsage on page 148	The objects that are associated with the data category group.

DataCategory

Represents an item (or data category) in the data category group. A data category can recursively contain a list of other data categories.

147

Field Name	Field Type	Description
dataCategory	DataCategory[]	A recursive list of sub data categories. For example, a list of countries within a continent. You can create up to 100 categories in a data category group and have up to 5 levels in a data category group hierarchy.
label	string	Required. Label for the data category throughout the Salesforce user interface.
name	string	Required. The developer name of the data category used as a unique identifier for API access. The name can only contain characters, letters, and the underscore (_) character, must start with a letter, and cannot end with an underscore or contain two consecutive underscore characters.

⊘ Important: The value for this field is defined once and cannot be changed later.

⚠ Warning: If you deploy a category group that already exists in an organization, any category that is not defined in the XML file is permanently removed from your organization. For more information see Usage.

ObjectUsage

Represents the objects that can be associated with the data category group. This association allows the object to be classified and filtered using the data categories.

Field Name	Field Type	Description
object	string[]	A list of the object names that can be associated with the data category group. Valid values are:

- `KnowledgeArticleVersion`—to associate articles. See "Modify Default Category Group Assignments for Articles" in the Salesforce online help for more information on data category groups association to articles.

- `Question`—to associate questions. You can associate the `Question` object with at most one category group.

⚠ Warning: If you deploy a category group that already exists in an organization, any object association that is not defined in the XML file is permanently removed from your organization. Ensure that your XML file specifies all the records associated with your category group in the organization. For more information see Usage.

Declarative Metadata Sample Definition

This sample is the definition of the `Geography` data category group and its data categories:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<DataCategoryGroup xmlns="http://soap.sforce.com/2006/04/metadata">
   <label>Geography</label>
   <description>Geography structure of service center locations</description>
   <fullName>geo</fullName>

   <dataCategory> <name>WW</name> <label>Worldwide</label>
      <dataCategory> <name>AMER</name> <label>North America</label>
         <dataCategory>
             <name>USA</name>
             <label>United States of America</label>
         </dataCategory>
         <dataCategory>
             <name>CAN</name>
             <label>Canada</label>
         </dataCategory>
         <dataCategory>
             <name>MEX</name>
             <label>Mexico</label>
         </dataCategory>
      </dataCategory>
      <dataCategory> <name>EMEA</name> <label>Europe, Middle East, Africa</label>
         <dataCategory>
             <name>FR</name>
             <label>France</label>
         </dataCategory>
         <dataCategory>
             <name>SP</name>
             <label>Spain</label>
       </dataCategory>
        <dataCategory>
            <name>UK</name>
            <label>United-Kingdom</label>
         </dataCategory>
      </dataCategory>
      <dataCategory>
          <name>APAC</name>
          <label>Asia</label>
      </dataCategory>
   </dataCategory>

   <objectUsage>
      <object>KnowledgeArticleVersion </object>
   <objectUsage>
</DataCategoryGroup>
```

Usage

When you deploy a category group XML file, Metadata API checks whether the category group exists in the target organization. If the category group does not exist, it is created. If the category group already exists, then Metadata API:

- Adds any new category or object defined in the XML file.

- Deletes any category that is not defined in the XML file. Records associated with the deleted categories are re-associated with the parent category.

- Deletes any object association that is not defined in the XML file.

- Moves any category if its hierarchical position differs from the position specified in the XML file.

> **Note:** When a category moves to a new parent category, users that have no visibility on the new parent category lose their visibility to the repositioned category.

> **Note:** For more information about category deletion, category repositioning and its impact on record categorization and visibility see "Delete a Data Category" and "Modify and Arrange Data Categories" in the Salesforce online help.

Using Metadata API to deploy category changes from one organization to another permanently removes categories and record categorizations that are not specified in your XML file. Salesforce recommends that you manually create data categories and record associations in an organization from Setup by entering `Data Categories` in the `Quick Find` box, then selecting **Data Categories** rather than deploying changes from a sandbox to a production organization.

The following example illustrates what happens if you deploy an XML representation of a `Geography` data category group hierarchy to an organization that already has this data category group defined. Note that the organization contains a `US` category, while the XML file includes a `USA` category in the same hierarchical position. The Metadata API deployment process deletes the `US` category from the organization and moves associations for any records from `US` to the parent `AMER` category. It also adds the `USA` category under `AMER`. Note that all records that were previously categorized with `US` are now associated with the `AMER` category.
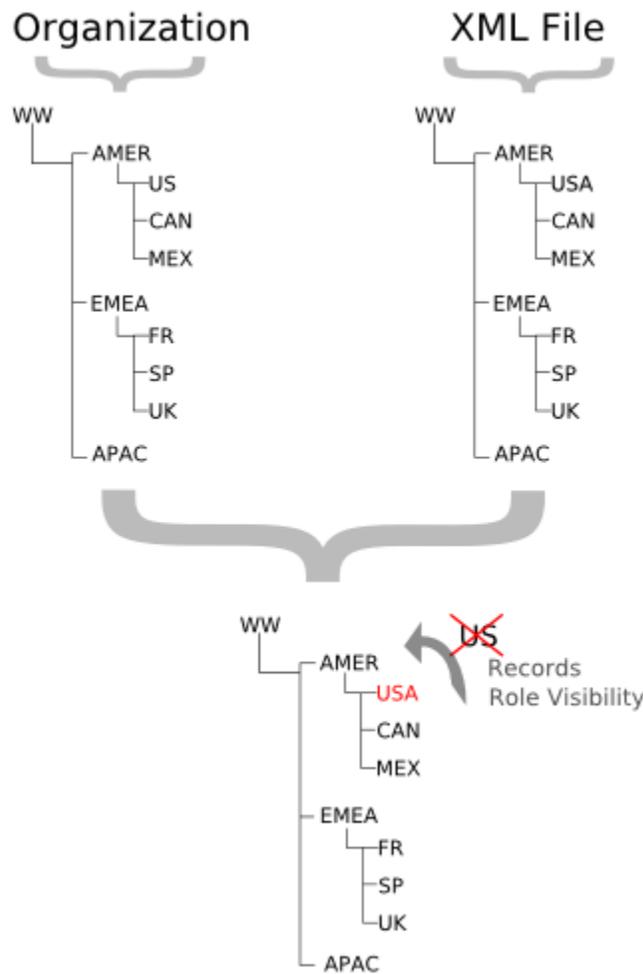
The next example illustrates what can happen when you delete or move a category in a data category group and deploy its XML representation from a sandbox to a production organization that already has this data category group defined. Hierarchy 1 shows the initial data category group in the sandbox organization. In hierarchy 2, we add an EU category under EMEA and move FR, SP and UK below EU. In hierarchy 3, we delete FR and associate its records with its new parent, EU. Finally, we deploy the changes from the sandbox to the production organization.
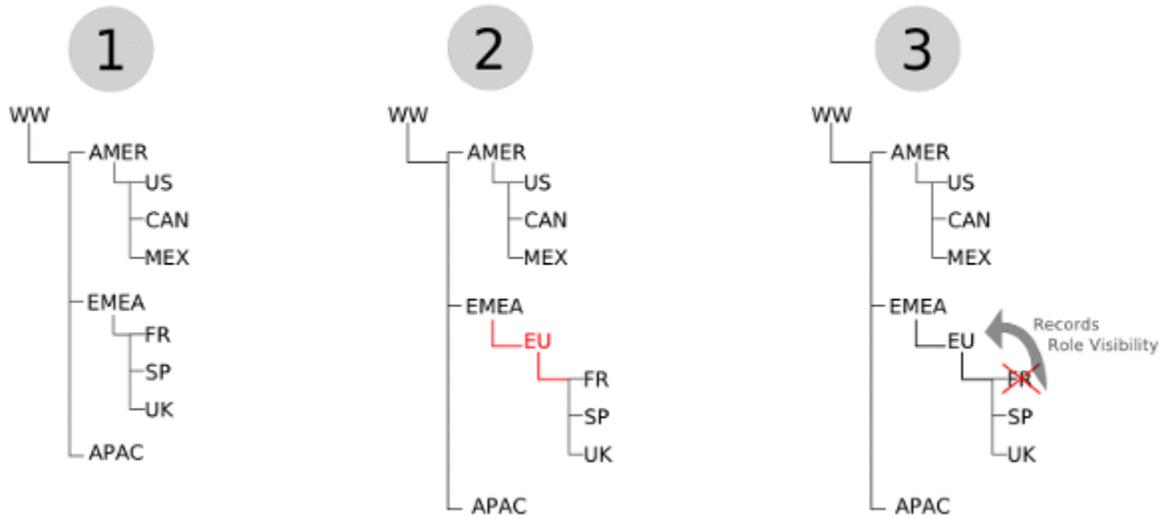
Metadata API has no concept of the order of the changes made to the sandbox organization. It just deploys the changes from one organization to another. During the deployment, it first notices the deletion of the `FR` category and removes it from the production organization. Consequently, it moves associations for any records from `FR` to its parent on the production organization, `EMEA`. Metadata API then adds the `EU` category and moves `SP` and `UK` below it. Although the category group hierarchy looks the same in both organizations, record categorization in production is different from the sandbox organization. The records that were originally associated with `FR` in hierarchy 1 are associated with `EU` in the sandbox organization, but are associated with `EMEA` in the production organization.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SearchSettings

Represents an org's search settings.

This type extends the Metadata metadata type and inherits its `fullName` field.

In the package manifest, all organization settings metadata types are accessed using the Settings name. See Settings for more details.

File Suffix and Directory Location

SearchSettings values are stored in a single file named `Search.settings` in the `settings` folder. The `.settings` files are different from other named components because there's only one settings file for each settings component.

Version

SearchSettings is available in API version 37.0 and later.

Fields

Field Name	Field Type	Description
documentContentSearchEnabled	boolean	Indicates whether a full-text document search is performed.
enableAdvancedSearchInAlohaSidebar	boolean	Indicates whether advanced search is available in the search sidebar (`true`) or not (`false`). Available in Salesforce Classic only. Available in API version 46.0 and later.
enableEinsteinSearchAssistantDialog	boolean	Indicates whether the Einstein search experience is enabled (`true`) or not (`false`). Available in API version 50.0 and later.
enableEinsteinSearchEs4kPilot	boolean	Indicates whether Einstein Search for Knowledge enhancements are enabled (`true`) or not (`false`). Available in API version 54.0 and later. This feature became generally available in Winter '23. In API version 56.0 and later, the default value is `true`.
enableEinsteinSearchNaturalLanguage	boolean	Indicates whether natural language search is enabled (`true`) or not (`false`). Available in API version 50.0 and later.
enableEinsteinSearchNLSFilters	boolean	Indicates whether the Natural Language Search Filters (Pilot) feature is enabled (`true`) or not (`false`). Available in API version 54.0 and later.
enableEinsteinSearchPersonalization	boolean	Indicates whether search personalization is enabled (`true`) or not (`false`). Available in Lightning Experience only. Available in API version 47.0 and later.
enablePersonalTagging	boolean	Indicates whether users are allowed to group records from various objects by a common theme (`true`) or not (`false`). Personal tags are visible to the user only. Available in Salesforce Classic only. Available in API version 48.0 and later.
enablePublicTagging	boolean	Indicates whether users are allowed to group records from various objects by a common theme (`true`) or not (`false`). Personal tags are visible to all users. Available in Salesforce Classic only. Available in API version 48.0 and later.
enableSalesforceGeneratedSynonyms	boolean	Indicates whether search synonyms are enabled (`true`) or not (`false`). Available in API version 47.0 and later.
enableSearchTermHistory	boolean	Indicates whether users are allowed to group records from various objects by a common theme (`true`) or not (`false`). Public tags are visible to everyone in the organization. Available in Salesforce Classic only. Available in API version 48.0 and later.
enableSetupSearch	boolean	Indicates whether the search box in the Setup sidebar returns matching custom fields, custom objects, and other supported setup items when you press Enter (`true`) or not (`false`). The default is `true` in Developer, Performance, Professional,

Field Name	Field Type	Description
		Enterprise, and Unlimited editions, and `false` in all other editions. Available in API version 47.0 and later.
`enableSuggestArticlesLinksOnly`	boolean	Indicates whether links are provided to knowledge articles from Cases similar to the current Case (`true`) or not (`false`). Available in API version 48.0 and later.
`enableUseDefaultSearchEntity`	boolean	Indicates whether to use the admin-specified default entity in sidebar search (`true`) or not (`false`). Available in Salesforce Classic only. Available in API version 48.0 and later.
`optimizeSearchForCJKEnabled`	boolean	Required. Indicates whether the search is optimized for the Japanese, Chinese, and Korean languages (`true`) or not (`false`). This setting affects sidebar search and the account search for **Find Duplicates** on a lead record in sidebar search and global search. Enable this option if users are searching mostly in Japanese, Chinese, or Korean, and if the text in searchable fields is mostly in those languages.
`recentlyViewedUsersForBlankLookupEnabled`	boolean	Required. Indicates whether the list of records that are returned from a user autocomplete lookup and from a blank user lookup is taken from the user's recently viewed user records (`true`). Otherwise this setting is `false` if the lookup shows a list of recently accessed user records from across your org (`false`). Only applies to User object blank lookup searches.
`searchSettingsByObject`	SearchSettingsByObject	Required. Represents a list of search settings for each object.
`sidebarAutoCompleteEnabled`	boolean	Required. Indicates whether autocomplete is enabled for sidebar search (`true`) or not (`false`). Autocomplete is when users start typing search terms and sidebar search displays a matching list of recently viewed records.
`sidebarDropDownListEnabled`	boolean	Required. Indicates whether a dropdown list appears in the sidebar search section (`true`) or not (`false`). From this list, users can select to search within tags, within a specific object, or across all objects.
`sidebarLimitToItemsIOwnCheckboxEnabled`	boolean	Required. Indicates whether the **Limit to Items I Own** checkbox appears (`true`) or not (`false`). The checkbox allows your users to include only records for which they are the record owner when entering search queries in the sidebar.
`singleSearchResultShortcutEnabled`	boolean	Required. Indicates whether a shortcut is enabled (`true`) or not (`false`). With the shortcut, users skip the search results page and go directly to the record's detail page when their search returns only a single item. This setting doesn't apply to tags, case comments (in advanced search), and global search.
`spellCorrectKnowledgeSearchEnabled`	boolean	Required. Indicates whether spell check is enabled for Knowledge search (`true`) or not (`false`).

SearchSettingsByObject

Field Name	Field Type	Description
searchSettingsByObject	ObjectSearchSetting	Contains a list of search settings for each object.

ObjectSearchSetting

A list of search settings for each object.

Field Name	Field Type	Description
enhancedLookupEnabled	boolean	Required. Indicates whether enhanced lookups is enabled for the object (`true`) or not (`false`).
lookupAutoCompleteEnabled	boolean	Required. Indicates whether autocomplete is enabled for lookup search (`true`) or not (`false`). Autocomplete is when users edit the lookup field inline by choosing an autosuggestion.
name	string	Required. The entity name of the object being configured.
resultsPerPageCount	int	Required. The number of search results per page.

Declarative Metadata Sample Definition

The following is an example of the `Search.settings` file.

```
<?xml version="1.0" encoding="UTF-8"?>
    <SearchSettings xmlns="http://soap.sforce.com/2006/04/metadata">
        <enableSetupSearch>false</enableSetupSearch>
        <enableAdvancedSearchInAlohaSidebar>false</enableAdvancedSearchInAlohaSidebar>
        <enableQuerySuggestionPigOn>false</enableQuerySuggestionPigOn>
        <enableSalesforceGeneratedSynonyms>false</enableSalesforceGeneratedSynonyms>
        <enableSearchTermHistory>false</enableSearchTermHistory>
        <enablePublicTagging>false</enablePublicTagging>
        <enablePersonalTagging>false</enablePersonalTagging>
        <enableSuggestArticlesLinksOnly>false</enableSuggestArticlesLinksOnly>
        <enableUseDefaultSearchEntity>false</enableUseDefaultSearchEntity>
            <documentContentSearchEnabled>true</documentContentSearchEnabled>
            <optimizeSearchForCJKEnabled>true</optimizeSearchForCJKEnabled>

<recentlyViewedUsersForBlankLookupEnabled>true</recentlyViewedUsersForBlankLookupEnabled>

            <searchSettingsByObject>
            <searchSettingsByObject>
                <enhancedLookupEnabled>false</enhancedLookupEnabled>
                <lookupAutoCompleteEnabled>false</lookupAutoCompleteEnabled>
                <name>Account</name>
                <resultsPerPageCount>25</resultsPerPageCount>
            </searchSettingsByObject>
            <searchSettingsByObject>
```

```
                    <enhancedLookupEnabled>false</enhancedLookupEnabled>
                    <lookupAutoCompleteEnabled>false</lookupAutoCompleteEnabled>
                    <name>Activity</name>
                    <resultsPerPageCount>25</resultsPerPageCount>
            </searchSettingsByObject>
            <searchSettingsByObject>
                    <enhancedLookupEnabled>false</enhancedLookupEnabled>
                    <lookupAutoCompleteEnabled>false</lookupAutoCompleteEnabled>
                    <name>Asset</name>
                    <resultsPerPageCount>25</resultsPerPageCount>
            </searchSettingsByObject>
            </searchSettingsByObject>
            <sidebarAutoCompleteEnabled>true</sidebarAutoCompleteEnabled>
            <sidebarDropDownListEnabled>true</sidebarDropDownListEnabled>

<sidebarLimitToItemsIOwnCheckboxEnabled>true</sidebarLimitToItemsIOwnCheckboxEnabled>
            <singleSearchResultShortcutEnabled>true</singleSearchResultShortcutEnabled>
            <spellCorrectKnowledgeSearchEnabled>true</spellCorrectKnowledgeSearchEnabled>

        <enableEinsteinSearchPersonalization>true</enableEinsteinSearchPersonalization>
    </SearchSettings>
```

Example Package Manifest

The following is an example package manifest used to deploy or retrieve the Search settings metadata for an organization.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Search</members>
        <name>Settings</name>
    </types>
    <version>37.0</version>
</Package>
```

Wildcard Support in the Manifest File

The wildcard character `*` (asterisk) in the `package.xml` manifest file doesn't apply to metadata types for feature settings. The wildcard applies only when retrieving all settings, not for an individual setting. For details, see Settings. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SearchLayouts

Represents the metadata associated with the search layouts for an object. You can customize which fields to display for users in search results, search filter fields, lookup dialogs, and recent record lists on tab home pages. You can access SearchLayouts only by accessing its encompassing CustomObject.

For more information, see Customize Layouts for Search Results and Customize Search Layouts for Custom Objects in Salesforce Help.

Version

Search layouts for custom objects are available in API version 14.0 and later. The ability to modify search layouts for standard objects (except events and tasks) is available in API version 27.0 and later.

Fields

When defining metadata for search layouts:

- Any Name field defined as a text type is mandatory; it's always displayed as the first column in the search results page. When you query for a list of fields; the name field isn't returned but all other fields are. If you define the Name field as an autonumber type, it's not mandatory and you can remove it from the list, but when you import the search layout with Metadata API, it will always add the Name field back. These rules apply to `customTabListAdditionalFields`, `lookupDialogsAdditionalFields`, `lookupPhoneDialogsAdditionalFields`, and `searchResultsAdditionalFields`

- For custom objects, the search layout uses the API name, for example, MyCustomField__c instead of the field name My Custom Field.

Field	Field Type	Description
`customTabListAdditionalFields`	string[]	The list of fields displayed in the Recent *Object Name* list view for an object.
`excludedStandardButtons`	string[]	The list of standard buttons excluded from the search layout.
`listViewButtons`	string[]	The list of buttons available in list views for an object. This field is equivalent to the Buttons Displayed value in the *Object Name* `List View` in the related list of the object detail page in the UI.
`lookupDialogsAdditionalFields`	string[]	The list of fields displayed in a lookup dialog for the object. Salesforce objects often include one or more *lookup fields* that allow users to associate two records together in a relationship. For example, a contact record includes an `Account` lookup field that represents the relationship between the contact and the organization with which the contact is associated. A lookup search dialog helps you search for the record associated with the one being edited. Lookup filter fields allow you to filter your lookup search by a customized list of fields in the object. This field is equivalent to the `Lookup Dialogs` related list on the object detail page in the UI.
`lookupFilterFields`	string[]	The list of fields that can be used to filter enhanced lookups for an object. Enhanced lookups are optionally enabled by your administrator. This field is equivalent to the `Lookup Filter Fields` related list on the object detail page in the application user interface.

Field	Field Type	Description
lookupPhoneDialogsAdditionalFields	string[]	The list of phone-related fields displayed in a lookup dialog for the object.
		This list enables integration of the fields with a softphone dial pad.
		This field is equivalent to the `Lookup Phone Dialogs` related list on the object detail page in the application user interface.
massQuickActions	string[]	The list of actions that you can use to perform mass quick action on records. Use this field to add an existing create or update action.
		You can perform mass quick actions on custom objects and all standard objects that support quick actions and have a search layout in Lightning Experience. This includes but isn't limited to cases, leads, accounts, campaigns, contacts, opportunities, and work orders.
searchFilterFields	string[]	The list of fields that can be used to filter a search for the object.
		This field is equivalent to the `Search Filter Fields` related list on the object detail page in the application user interface.
searchResultsAdditionalFields	string[]	The list of fields displayed in a search result for the object.
		This field is equivalent to the `Search Results` related list on the object detail page in the application user interface.
searchResultsCustomButtons	string[]	The list of custom buttons available in a search result for the object. The actions associated with the buttons can be applied to any of the records returned in the search result.

Declarative Metadata Sample Definition

A sample definition of object's search layout is shown..

```
<?xml version="1.0" encoding="UTF-8"?>
        <CustomObject xmlns="http://soap.sforce.com/2006/04/metadata">
        . . .
        <searchLayouts>
        <listViewButtons>New</listViewButtons>
        <listViewButtons>Accept</listViewButtons>
        <listViewButtons>ChangeOwner</listViewButtons>
        <lookupDialogsAdditionalFields>firstQuote__c</lookupDialogsAdditionalFields>

        <lookupDialogsAdditionalFields>finalQuote__c</lookupDialogsAdditionalFields>
```

```
            <massQuickActions>Create_MQA_Contact</massQuickActions>
        <searchResultsAdditionalFields>CREATEDBY_USER</searchResultsAdditionalFields>

        </searchLayouts>
        . . .
        </CustomObject>
```

Wildcard Support in the Manifest File

This metadata type doesn't support the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

SynonymDictionary

Represents a set of synonym groups, which are groups of words or phrases that are treated as equivalent in users' searches. You can define synonym groups to optimize search results for acronyms, variations of product names, and other terminology unique to your organization.

Synonyms are available in Service Cloud features such as Salesforce Knowledge. This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

SynonymDictionary components have the suffix `.synonymDictionary` and are stored in the `synonymDictionaries` folder.

Version

SynonymDictionary components are available in API version 29.0 and later.

Special Access Rules

Synonyms must be enabled in your organization. Only users with the "Manage Synonyms" permission can access this object.

Fields

Field Name	Field Type	Description
groups	SynonymGroup	The synonym groups defined in this dictionary.
isProtected	boolean	Indicates whether this component is protected (`true`) or not (`false`). Protected components cannot be linked to or referenced by components created in the installing organization.
label	string	Required. Specifies the display name of the synonym dictionary.

SynonymGroup

Represents a group of synonymous words or phrases.

Field Name	Field Type	Description
languages	Language	Required. Specifies the languages the synonym group applies to. If synonyms are specific to a single language, specify only that language. If the synonyms apply to multiple languages, specify multiple languages for one synonym group.
terms	string	Required. A word or phrase synonymous with other terms in the group. Maximum of 50 characters. Minimum of two terms per group.
		Synonym groups are symmetric, which means that if oranges and apples are defined in a synonym group, a search for *oranges* will return a match for *apples*, and vice versa for a search for *apples*.

Declarative Metadata Sample Definition

The following is an example of a SynonymDictionary component:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<SynonymDictionary xmlns="http://soap.sforce.com/2006/04/metadata">
    <groups>
        <languages>en_US</languages>
        <terms>Salesforce</terms>
        <terms>salesforce.com</terms>
        <terms>The Customer Company</terms>
        <terms>SFDC</terms>
    </groups>
    <groups>
        <languages>fr</languages>
        <terms>renault</terms>
        <terms>clio</terms>
    </groups>
    <label>Sample Dictionary</label>
</SynonymDictionary>
```

The following is an example `package.xml` that references the SynonymDictionary component.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
    <types>
        <members>Sample Dictionary</members>
        <name>SynonymDictionary</name>
    </types>
    <version>66.0</version>
</Package>
```

Usage

If you have existing synonym groups defined before API version 29.0, your existing groups are associated with a default dictionary called `_Default`.

If you have a set of synonyms that require frequent updates, we recommend assigning the synonym group or groups to a dedicated dictionary with a small number of groups. Each time you deploy an existing dictionary, all of its synonym groups are overwritten. We don't support deploying updates to only a single synonym group within a dictionary.

Wildcard Support in the Manifest File

This metadata type supports the wildcard character `*` (asterisk) in the `package.xml` manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

ExternalDataSource

Represents the metadata associated with an external data source. Create external data sources to manage connection details for integration with data and content that are stored outside your Salesforce org.

> 📝 Note: All credentials stored within this entity are encrypted under a framework that is consistent with other encryption frameworks on the platform. Salesforce encrypts your credentials by auto-creating org-specific keys. Credentials encrypted using the previous encryption scheme are migrated to the new framework.

This type extends the Metadata metadata type and inherits its `fullName` field.

File Suffix and Directory Location

ExternalDataSource components are stored in the `dataSources` directory of the corresponding package directory. ExternalDataSource components have the suffix `.dataSource`, and the prefix is the name of the external data source.

Version

ExternalDataSource components are available in API version 28.0 and later.

Special Access Rules

As of Spring '20 and later, only authenticated internal and external users can access this type.

Fields

Field Name	Field Type	Description
authProvider	string	The authentication provider represented by the AuthProvider component.
certificate	string	If you specify a certificate, your Salesforce org supplies it when establishing each two-way SSL connection with the external system.

161

Field Name	Field Type	Description
		The certificate is used for digital signatures, which verify that requests are coming from your Salesforce org. 💡 **Tip:** For best performance, verify that your remote HTTPS encrypted sites have OCSP (Online Certificate Status Protocol) stapling turned on.
customConfiguration	string	A string of configuration parameters that are specific to the external data source's `type`. - customConfiguration for Salesforce Connect—Cross-Org Adapter - customConfiguration for Salesforce Connect—OData 2.0 or 4.0 Adapter - customConfiguration for Salesforce Connect—Custom Adapter
customHttpHeaders	CustomHttpHeaders[]	Represents custom HTTP headers used with OData 2.0 or OData 4.0 connectors. Available in API version 43.0 or later.
endpoint	string	The URL of the external system, or if that URL is defined in a named credential, the named credential URL. Corresponds to the `URL` in the user interface. A named credential URL contains the scheme `callout:`, the name of the named credential, and an optional path. For example: `callout:My_Named_Credential/some_path`. You can append a query string to a named credential URL. Use a question mark (?) as the separator between the named credential URL and the query string. For example: `callout:My_Named_Credential/some_path?format=json`.
externalDataSrcDescriptors	ExternalDataSrcDescriptors[]	Represents schema descriptors for an external data source used with the Salesforce Connect adapter for Amazon DynamoDB (available in API version 55.0 or later) or Amazon Athena (available in API version 56.0 or later).
isWritable	boolean	Allows the Lightning Platform and users in this org to create, update, and delete records for external objects associated with the external data source. The external object data is stored outside the org. By default, external objects are read-only. Corresponds to `Writable External Objects` in the user interface. Available in API version 35.0 and later. However, with the cross-org adapter for Salesforce Connect, you can set this field to `true` only in API version 39.0 and later.
label	string	A name for the external data source. The label is displayed in the Salesforce user interface, such as in list views. Examples include Acme Team Marketing Site or Acme SharePoint.

Field Name	Field Type	Description
`namedCredential`	string	Represents the definition of the referenced named credential for an external data source of the type Amazon DynamoDB or Amazon Athena.
`oauthRefreshToken`	string	The OAuth refresh token. Used to obtain a new access token for an end user when a token expires.
`oauthScope`	string	Specifies the scope of permissions to request for the access token. Corresponds to the `Scope` in the user interface.
`oauthToken`	string	The access token issued by the external system.
`password`	string	The password your org uses to access the external system. Make sure that the credentials you use have adequate privileges to access the external system, perform searches, return data, and return information about the external system's metadata.
`principalType`	`External PrincipalType` (enumeration of type string)	Determines whether you're using one set or multiple sets of credentials to access the external system. Corresponds to `Identity Type` in the user interface. The valid values are: • `Anonymous` • `PerUser` • `NamedUser`
`protocol`	`Authentication Protocol` (enumeration of type string)	The authentication protocol that's required to access the external system. The valid values are: • `NoAuthentication` • `Oauth` • `Password` For cloud-based Files Connect external systems, select **Oauth 2.0**. For on-premises systems, select **Password Authentication**. For Simple URL data sources, select **No Authentication**.
`repository`	string	Used for SharePoint Online. If metadata isn't accessible, use this field to create tables and default table fields.
`type`	`ExternalDataSourceType` (enumeration of type string)	Required. For Salesforce Connect, specifies the adapter that connects to the external system. The valid values are: • `AmazonAthena`—Amazon Athena • `AmazonDynamoDB`—Amazon DynamoDB • `OData`—OData 2.0 adapter • `OData4`—OData 4.0 adapter • `SfdcOrg`—cross-org adapter • *`ApexClassId`*—`DataSource.Provider` class that defines the custom adapter created via the Apex Connector Framework

Field Name	Field Type	Description
		For Files Connect, specifies the data source type. The valid values are:

- `ContentHubSharepoint`—SharePoint 2010 or 2013
- `ContentHubSharepointOffice365`—SharePoint Online
- `ContentHubSharepointOneDrive`—OneDrive for Business
- `ContentHubGDrive`—Google Drive
- `ContenHubIsotope`—Isotope

If Chatter is enabled, you can also specify `SimpleURL` to access data hosted on a web server that doesn't require authentication.

- `outgoingemail`—A data source used for sending an email through a quick action.

For Digital Lending Configurator, the valid value is:

- `AFPPAttribute`—The data source name for the Application Form Product Proposal Attribute virtual object.

For the federated search external data source type, the valid value is:

- `OpenSearch`

For Transaction Management in Revenue Cloud, the valid values are:

- `ASPAttribute`—The data source name for the Asset State Period Attribute virtual object. Available in API version 63.0 and later.
- `OIAttribute`—The data source name for the Order Item Attribute virtual object. Available in API version 63.0 and later.
- `QLIAttribute`—The data source name for the Quote Line Item Attribute virtual object. Available in API version 63.0 and later.

For SalesAgreement in Manufacturing Cloud, the valid values are:

- `SAPAttribute`—The data source name for the SalesAgreement Product Attribute virtual object. Available in API version 60.0 and later.

These values are reserved for internal use:

- `AssetAttribute`
- `ClaimAttributeDS`
- `ClaimItemAttributeDS`
- `CryptoTrEnvChgLogSnp`
- `CtrtGrpPlnAttr`
- `CtrtGrpPlnGrpClsAttr`

Field Name	Field Type	Description
		• FAAttribute
		• FLAttribute
		• IAItemProdtAttr
		• Identity
		• InsPolicyAttribute
		• IPAAttribute
		• IPCAttribute
		• IPCvrBnftAttribute
		• IPPAttribute
		• SdbOvenPODataSource
		• Wrapper
username	string	The user name that your org uses to access the external system. Make sure that the credentials you use have adequate privileges to access the external system, perform searches, return data, and return information about the external system's metadata.
version	string	Reserved for future use.

CustomHttpHeaders

Represents a custom HTTP header used with OData 2.0 or OData 4.0 connectors. Available in API version 43.0 or later.

Field Name	Field Type	Description
description	string	A text description of the header field's purpose.
headerFieldName	string	Required. Name of the header field. The name must contain at least one alphanumeric character or underscore. It can also include these characters: ! # $ % & ' * + - . ^ _ ` | ~.
headerFieldValue	string	Required. A formula that resolves to the value for the header. The values in the formula must evaluate to a string. If the formula resolves to null and an empty string, the header isn't sent.
isActive	boolean	Indicates whether the custom HTTP header is available to use (true) or unavailable (false).

customConfiguration for Salesforce Connect—Cross-Org Adapter

This sample JSON-encoded configuration string defines parameters that apply when the external data source's type is set to SfdcOrg.

```
{"apiVersion":"32.0","environment":"CUSTOM",
"searchEnabled":"true","timeout":"120"}
```

The parameters correspond to these fields in the user interface:

- `apiVersion`—API Version
- `environment`—Connect to
- `searchEnabled`—Enable Search
- `timeout`—Connection Timeout

`customConfiguration` for Salesforce Connect—OData 2.0 or 4.0 Adapter

This JSON-encoded configuration string defines parameters that apply when the external data source's `type` is set to `OData` or `OData4`.

```
{"inlineCountEnabled":"true","csrfTokenName":"X-CSRF-Token",
"requestCompression":"false","pagination":"CLIENT",
"noIdMapping":"false","format":"ATOM",
"searchFunc":"","compatibility":"DEFAULT",
"csrfTokenEnabled":"true","timeout":"120",
"searchEnabled":"true"}
```

The parameters correspond to these fields in the user interface.

- `compatibility`—Special Compatibility
- `csrfTokenEnabled`—Cross-Site Request Forgery (CSRF) Protection
- `csrfTokenName`—Anti-CSRF Token Name
- `format`—Format
- `inlineCountEnabled`—Request Row Counts
- `noIdMapping`—High Data Volume
- `pagination`—Server Driven Pagination
- `requestCompression`—Compress Requests
- `searchEnabled`—Enable Search
- `searchFunc`—Custom Query Option for Salesforce Search
- `timeout`—Connection Timeout

Declarative Metadata Sample Definition: OData 2.0 or 4.0

The following is the definition of an external data source for Salesforce Connect—OData 2.0 or 4.0 adapter.

```
<?xml version="1.0" encoding="UTF-8"?>
<ExternalDataSource xmlns="http://soap.sforce.com/2006/04/metadata">
    <authProvider>FacebookAuth</authProvider>
    <customConfiguration>{"compatibility":"DEFAULT",
    "noIdMapping":"false","inlineCountEnabled":"true",
    "searchEnabled":"true","format":"ATOM",
    "requestCompression":"false","pagination":"SERVER",
    "timeout":"120"}</customConfiguration>
    <customHttpHeaders>
        <headerFieldName>X-User</headerFieldName>
        <headerFieldValue>$User.Username</headerFieldValue>
    </customHttpHeaders>
    <endpoint>http://myappname.herokuapp.com/DataHub.svc</endpoint>
    <label>DataHub</label>
    <principalType>NamedUser</principalType>
    <protocol>Oauth</protocol>
```

```
    <type>OData</type>
</ExternalDataSource>
```

customConfiguration for Salesforce Connect—Custom Adapter

This sample JSON-encoded configuration string defines the parameter that applies when the external data source's `type` is set to the ID of a `DataSource.Provider` class.

```
{"noIdMapping":"false"}
```

The `noIdMapping` parameter corresponds to the `High Data Volume` field in the user interface.

ExternalDataSrcDescriptors for Salesforce Connect Adapter for Amazon DynamoDB and for Amazon Athena

Represents schema descriptors for an external data source used with the Salesforce Connect adapter. The schema descriptors are for Amazon DynamoDB (available in API version 55.0 or later) or Amazon Athena (available in API version 56.0 or later).

Field Name	Field Type	Description
customObject	string	If set, the external object associated with the descriptor.
descriptor	string	Required. The descriptor document that contains the metadata information.
descriptorVersion	string	If the external system supports schema versioning for the data source, the optional descriptor document version tracks the external system's schema version. Several descriptors with different document versions can be active.
developerName	string	Required. The unique name of the child-level setup entity.
externalDataSource	string	Required. The name of the external data source associated with the descriptor.
subtype	ExternalDataSrcDescSubtype (enumeration of type string)	Required. The subtype of the descriptor. Values are: • `SchemaTableMetadata`— Used to cache information about the external system. • `SchemaTableQualifiers`— Used to customize the data retrieval query to the external system.
systemVersion	int	Required. The version that defines the descriptor format and provides compatibility with descriptor formats between Salesforce releases.
type	ExternalDataSrcDescType (enumeration of type string)	Required. The type of the descriptor. Valid value: • `Schema`

Declarative Metadata Sample Definition: Amazon DynamoDB

The following is an example of an external data source for the Salesforce Connect adapter for Amazon DynamoDB that uses
`ExternalDataSrcDescriptor` component.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ExternalDataSource xmlns="http://soap.sforce.com/2006/04/metadata">
    <customConfiguration>{"timeout":"120"}</customConfiguration>
    <externalDataSrcDescriptors>
        <fullName>MyQualifierName</fullName>
        <customObject>MyExternalObject__x</customObject>
        <descriptor>
        {
          "tableName": "MyDynamoDBTable",
          "columns": {
            "MyField": {"presence": "exists"}
          }
        }
        </descriptor>
        <developerName>MyQualifierName</developerName>
        <externalDataSource>MyDataSource</externalDataSource>
        <subtype>SchemaTableQualifiers</subtype>
        <systemVersion>0</systemVersion>
        <type>Schema</type>
    </externalDataSrcDescriptors>
    <isWritable>true</isWritable>
    <label>MyDataSource</label>
    <namedCredential>MyNamedCredential</namedCredential>
    <principalType>Anonymous</principalType>
    <protocol>NoAuthentication</protocol>
    <type>AmazonDynamoDb</type>
</ExternalDataSource>
```

Declarative Metadata Sample Definition: Amazon Athena

The following is an example of an external data source for the Salesforce Connect adapter for Amazon Athena that uses
`ExternalDataSrcDescriptor` component.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ExternalDataSource xmlns="http://soap.sforce.com/2006/04/metadata">
    <customConfiguration>
    {
      "DataCatalog": "AwsDataCatalog",
      "timeout": "120"
    }
    </customConfiguration>
    <externalDataSrcDescriptors>
        <fullName>MyAthenaQualifierName</fullName>
        <customObject>MyAthenaExternalObject__x</customObject>
        <descriptor>
        {
          "tableName": "myathenadatabase.myathenatable",
          "extendedQualifiers": {"workgroup": "primary"},
          "keyColumns": ["ExternalIdComponent", "OtherExternalIdComponent"]
        }
        </descriptor>
```

```
            <developerName>MyAthenaQualifierName</developerName>
            <externalDataSource>MyAthenaDataSource</externalDataSource>
            <subtype>SchemaTableQualifiers</subtype>
            <systemVersion>0</systemVersion>
            <type>Schema</type>
    </externalDataSrcDescriptors>
    <isWritable>false</isWritable>
    <label>MyAthenaDataSource</label>
    <namedCredential>MyAthenaNamedCredential</namedCredential>
    <principalType>Anonymous</principalType>
    <protocol>NoAuthentication</protocol>
    <type>AmazonAthena</type>
</ExternalDataSource>
```

Wildcard Support in the Manifest File

This metadata type supports the wildcard character * (asterisk) in the package.xml manifest file. For information about using the manifest file, see Deploying and Retrieving Metadata with the Zip File.

CHAPTER 5   Salesforce Knowledge with SOQL and SOSL

Use the Salesforce Object Query Language (SOQL) to search your organization's Knowledge data for specific information. SOQL is similar to the SELECT statement in the widely used Structured Query Language (SQL) but is designed specifically for Salesforce. Use the Salesforce Object Search Language (SOSL) to construct text-based search queries against the search index.

Knowledge Object	Limits

KnowledgeArticleVersion	- Filter on a single value of `PublishStatus` for best results. To find all versions of each article, omit the `PublishStatus` filter, but do filter on one or more master key IDs. To retrieve all archived versions for a given article, specify a SOQL filter where `IsLatestVersion` is `false`. - In API version 46.0 and earlier, queries without a filter on `PublishStatus` return published articles by default. In API version 47.0 and later, draft, published, and archived articles are returned when Lightning Knowledge is enabled. - To support security, only users with the "View Draft Articles" permission see articles whose `PublishStatus` value is `Draft`. Similarly, only users with the "View Archived Articles" permission see articles whose `PublishStatus` value is `Archived` - Archived article versions are stored in the *`Knowledge__`*`kav` object. To query archived article versions, specify the article `Id` and set `IsLatestVersion='0'`. - You can't use binding variables in Apex SOQL statements with KnowledgeArticleVersion objects. For example, the following SOQL statement causes a compilation error.
	
	`final String`
	`PUBLISH_STATUS_ONLINE =`
	`'Online';`
	`List<Knowledge__kav> articles`

170

Knowledge Object	Limits
	``` = [SELECT Id FROM Knowledge__kav WHERE PublishStatus = :PUBLISH_STATUS_ONLINE]; ``` Instead, use dynamic SOQL as follows. See Dynamic SOQL in *Apex Developer Guide*. ``` final String PUBLISH_STATUS_ONLINE = 'Online'; final String q = 'SELECT Id, PublishStatus FROM Knowledge__kav WHERE PublishStatus = :PUBLISH_STATUS_ONLINE'; List<Knowledge__kav> articles = Database.query(q); ```

For generic SOQL and SOSL information see the SOQL and SOSL Reference.

UPDATE

Track keywords that are used in Salesforce Knowledge article searches by using the `UPDATE TRACKING` optional clause. Determine how many hits a Salesforce Knowledge article has had by using the `UPDATE VIEWSTAT` optional clause.

Update an Article's Keyword Tracking with SOQL

`UPDATE TRACKING` is an optional clause that can be added to a `SELECT` statement of a SOQL query to report on article searches and views. Developers can use `UPDATE TRACKING` to track the keywords that are used in Salesforce Knowledge article searches.

```
SELECT Title FROM FAQ__kav
WHERE Keyword='Apex' and
Language = 'en_US' and
KnowledgeArticleVersion = 'ka230000000PCiy'
UPDATE TRACKING
```

Update an Article Viewstat with SOQL

The `UPDATE VIEWSTAT` clause is used in a `SELECT` statement to report on Salesforce Knowledge article searches and views. You can get a view count for every article that you have access to online. Developers can use `UPDATE VIEWSTAT` to update an article's view statistics.

You can use this syntax to increase the view count for every article you have access to online.

```
SELECT Title FROM Knowledge__kav
   WHERE PublishStatus='online' and
   Language = 'en_US' and
   KnowledgeArticleVersion = 'ka230000000PCiy'
   UPDATE VIEWSTAT
```

WITH

You can filter records based on field values, for example, to filter according to category or to query and retrieve changes that are tracked in a user's profile feed by using `WITH` *`filteringExpression`*. This optional clause can be added to a `SELECT` statement of a SOQL query.

Unlike the `WHERE` clause, which only supports fields from the object specified in the `FROM` clause, `WITH` allows you to filter by other related criteria. For example, you can use the `WITH` clause to filter articles based on their classification in one or more data category groups. The `WITH` clause can only be used in the following cases:

- To filter records based on their categorization. See `WITH DATA CATEGORY`
- To query and retrieve record changes tracked in a user profile feed. See Custom Object__Feed in the *Object Reference for the Salesforce Platform*.
- To filter records based on user access permissions on fields or objects being queried within Apex code. See Filter SOQL Queries Using WITH SECURITY_ENFORCED in the *Apex Developer Guide*.
- To specify user mode access for Apex database operations. Apex code runs in system mode by default, which means that it runs with substantially elevated permissions over the user running the code. To enhance the security context of Apex, you can specify user mode access by using `WITH USER_MODE`. Field-level security (FLS) and object permissions of the running user are respected

Salesforce Knowledge with SOQL and SOSL

WITH DATA CATEGORY

in user mode, unlike in system mode. User mode always applies sharing rules, but in system mode they're controlled by sharing keywords on the class. See Enforce User Mode for Database Operations in the *Apex Developer Guide*.

> 📝 **Note:** Salesforce recommends that you enforce field- and object-level security permissions by using `WITH USER_MODE` instead of `WITH SECURITY_ENFORCED` because it has fewer limitations.

If `WITH` is specified, the query returns only records that match the filter and are visible to the user. If unspecified, the query returns only the matching records that are visible to the user.

The filtering expression in this statement is highlighted in bold.

- `SELECT Title FROM KnowledgeArticleVersion WHERE PublishStatus='online' `**`WITH DATA CATEGORY Geography__c ABOVE usa__c`**

- `SELECT Id FROM UserProfileFeed `**`WITH UserId='005D0000001AamR'`**` ORDER BY CreatedDate DESC, Id DESC LIMIT 20`

IN THIS SECTION:

WITH DATA CATEGORY

You can search for Salesforce Knowledge articles and questions by their data category in a SOQL query. `WITH DATA CATEGORY` is an optional clause in a `SELECT` statement that's used to filter records that are associated with one or more data categories and are visible to users.

WITH DATA CATEGORY

You can search for Salesforce Knowledge articles and questions by their data category in a SOQL query. `WITH DATA CATEGORY` is an optional clause in a `SELECT` statement that's used to filter records that are associated with one or more data categories and are visible to users.

```
[WITH [DATA CATEGORY] filteringExpression]
```

If `WITH DATA CATEGORY` is specified, the `query()` returns only matching records that are associated with the specified data categories and are visible to the user. If unspecified, the `query()` only returns the matching records that are visible to the user.

> ⛔ **Important:** `CategoryData` is an object and DATA CATEGORY is syntax in a SOQL `WITH` clause. `WITH DATA CATEGORY` is valid syntax, but `WITH CategoryData` is not supported.

A SOQL statement using a `WITH DATA CATEGORY` clause must also include a `FROM `***`ObjectTypeName`*** clause where ***`ObjectTypeName`*** equals:

- `KnowledgeArticleVersion` to query all article types
- an article type `API Name` to query a specific article type
- `Question` to query questions

When ***`ObjectTypeName`*** equals to KnowledgeArticleVersion or any article type `API Name` in the `FROM` clause, a `WHERE` clause must be specified with one of the following parameters:

- `PublishStatus` to query articles depending on their status in the publishing cycle:
  - `WHERE PublishStatus='online'` for published articles
  - `WHERE PublishStatus='archived'` for archived articles
  - `WHERE PublishStatus='draft'` for draft articles
- `Id` to query an article based on its id

For information on article types or questions, see "Knowledge Article Types" in Salesforce Help.

> 📝 **Note:** The `WITH DATA CATEGORY` clause does not support [bind variables](#). The clause can have no more than three data category conditions.

filteringExpression

The *filteringExpression* in the `WITH DATA CATEGORY` clause uses the following syntax:

```
dataCategorySelection [AND [dataCategorySelection][...]
```

The examples in this section are based on the following data category group:

```
Geography__c
    ww__c
        northAmerica__c
            usa__c
            canada__c
            mexico__c
        europe__c
            france__c
            uk__c
        asia__c
```

The category filtering in the statements below is highlighted in bold.

- `SELECT Title FROM KnowledgeArticleVersion WHERE PublishStatus='online'` **`WITH DATA CATEGORY Geography__c ABOVE usa__c`**
- `SELECT Title FROM Question WHERE LastReplyDate > 2005-10-08T01:02:03Z` **`WITH DATA CATEGORY Geography__c AT (usa__c, uk__c)`**
- `SELECT UrlName FROM KnowledgeArticleVersion WHERE PublishStatus='draft'` **`WITH DATA CATEGORY Geography__c AT usa__c AND Product__c ABOVE_OR_BELOW mobile_phones__c`**

You can only use the `AND` logical operator. The following syntax is incorrect as `OR` is not supported:

```
WITH DATA CATEGORY Geography__c ABOVE usa__c OR Product__c AT mobile_phones__c
```

IN THIS SECTION:

[dataCategorySelection](#)
The syntax of the data category selection in a `WITH DATA CATEGORY` clause in a SOQL query includes a category group name to use as a filter, the filter selector, and the name of the category to use for filtering.

dataCategorySelection

The syntax of the data category selection in a `WITH DATA CATEGORY` clause in a SOQL query includes a category group name to use as a filter, the filter selector, and the name of the category to use for filtering.

The *dataCategorySelection* uses the following syntax:

```
dataCategoryGroupName filteringSelector dataCategoryName
```

Syntax	Description
dataCategoryGroupName	The name of the data category group to use as a filter. `Geography__c` is the data category group in the following example.
	You cannot use the same data category group more than once in a query. As an example, the following command is incorrect: `WITH DATA CATEGORY Geography__c ABOVE usa__c AND Geography__c BELOW europe__c`
filteringSelector	The selector used to filter the data in the specified data category. See Filtering Selectors for a list of valid selectors.
dataCategoryName	The name of the data category for filtering. You must have visibility on the category you specify.
	You can use parentheses to apply the filtering operator to more than one data category. Each data category must be separated by a comma.
	Example: `WITH DATA CATEGORY Geography__c AT (usa__c,france__c,uk__c)`
	You can't use the `AND` operator instead of parentheses to list multiple data categories. The following syntax does not work `WITH DATA CATEGORY Geography__c AT usa__c AND france__c`

Filtering Selectors

When specifying filters for a `WITH DATA CATEGORY` clause of a SOQL query, you can use `AT` to select the specified category, `ABOVE` to select the category and all its parent categories, `BELOW` to select the category and all its subcategories, and `ABOVE_OR_BELOW` to select the category, its parent categories, and its subcategories.

The examples in this section are based on the following data category group:

```
Geography__c
    ww__c
        northAmerica__c
            usa__c
            canada__c
            mexico__c
        europe__c
            france__c
            uk__c
        asia__c
```

This table lists the *filteringSelector* values that are used in the *dataCategorySelection* syntax.

Selector	Description
`AT`	Select the specified data category.
	For example, the following syntax selects `asia__c`.
	`WITH DATA CATEGORY Geography__c AT asia__c`

Selector	Description
ABOVE	Select the specified data category and all its parent categories.
	For example, the following syntax selects `usa__c`, `northAmerica__c`, and `ww__c`.
	``` WITH DATA CATEGORY Geography__c ABOVE usa__c ```
BELOW	Select the specified data category and all its subcategories.
	For example, the following selects `northAmerica__c`, `usa__c`, `canada__c`, and `mexico__c`.
	``` WITH DATA CATEGORY Geography__c BELOW northAmerica__c ```
ABOVE_OR_BELOW	Select the specified data category and
	• all its parent categories
	• all its subcategories
	For example, the following selects `ww__c`, `europe__c`, `france__c`, and `uk__c`.
	``` WITH DATA CATEGORY Geography__c ABOVE_OR_BELOW europe__c ```

Here are some more examples of WITH DATA CATEGORY clauses in a SELECT statement in a SOQL query.

Type of Search	Examples
Select the title from all questions classified with the `mobile_phones__c` data category in the `Product__c` data category group	``` SELECT Title FROM Question WHERE LastReplyDate < 2005-10-08T01:02:03Z WITH DATA CATEGORY Product__c AT mobile_phones__c ```
Select the title and summary from all published Knowledge articles classified: • above or below `europe__c` in the `Geography__c` data category group • below `allProducts__c` in the `Product__c` data category group	``` SELECT Title, Summary FROM KnowledgeArticleVersion WHERE PublishStatus='Online' AND Language = 'en_US' WITH DATA CATEGORY Geography__c ABOVE_OR_BELOW europe__c AND Product__c BELOW All__c ```
Select the ID and title from draft articles of type "Offer__kav" classified : • with the `france__c` or `usa__c` data category in the `Geography__c` data category group • above the `dsl__c` data category in the `Product__` data category group	``` SELECT Id, Title FROM Offer__kav WHERE PublishStatus='Draft' AND Language = 'en_US' WITH DATA CATEGORY Geography__c AT (france__c,usa__c) AND Product__c ABOVE dsl__c ```

Using Relationship Queries with Data Category Selection Objects

Data categories are used to classify records. In SOQL, you can use the *Article*__DataCategorySelection or QuestionDataCategorySelection objects. You can also build a relationship query with the `DataCategorySelections` relationship name in a `FROM` clause.

Imagine an `Offer` article type. The following query returns the ID of any categorization associated with an offer and the ID of the categorized article.

```
SELECT Id,ParentId
FROM Offer__DataCategorySelection
```

The following example uses the `DataCategorySelections` relationship name to build a relationship query that returns the ID of published offers and the ID of all the categorizations associated to these offers.

```
SELECT Id, Title,
  (
    SELECT Id
    FROM DataCategorySelections
  )
FROM Offer__kav WHERE PublishStatus='online'
```

About SOSL

Use the Salesforce Object Search Language (SOSL) to construct text-based search queries against the search index.

You can search text, email, and phone fields for multiple objects, including custom objects, that you have access to in a single query in the following environments.

- SOAP or REST calls
- Apex statements
- Visualforce controllers and getter methods
- Schema Explorer of the Eclipse Toolkit

Note: If your org has relationship queries enabled, SOSL supports SOQL relationship queries.

Compare SOSL and SOQL

Use SOQL when:

- You want to search against the org's database. Results from a database search include matches for the exact search string.
- You know in which objects or fields the data resides.
- You want to:
    - Retrieve data from a single object or from multiple objects that are related to one another
    - Count the number of records that meet specified criteria
    - Sort results as part of the query
    - Retrieve data from number, date, or checkbox fields

Use SOSL when:

- You don't know in which object or field the data resides, and you want to find it in the most efficient and fastest way possible.

- You want to:
  - Retrieve data for a specific term that you know exists within a field. Because SOSL can tokenize multiple terms within a field and build a search index off this, SOSL searches are faster and can return more relevant results. Depending on the search object, SOSL searches also take advantage of the advanced features of the search index, such as out-of-order matching, synonyms, lemmatization, and spell check.
  - Retrieve multiple objects and fields efficiently, and the objects might or might not be related to one another.
  - Retrieve data for a particular division in an org using the divisions feature, and you want to find it in the most efficient way possible.

Define Efficient SOSL Text Searches

If your searches are too general, they are slow and return too many results. Use the following clauses to write more targeted and useful searches.

- `IN`: Limits the types of fields to search, including email, name, or phone.
- `LIMIT`: Specifies the maximum number of rows to return.
- `OFFSET`: Displays the search results on multiple pages.
- `RETURNING`: Limits the objects and fields to return.
- `WITH DATA CATEGORY`: Specifies the data categories to return.
- `WITH DivisionFilter`: Specifies the division field to return.
- `WITH NETWORK`: Specifies the Experience Cloud site ID to return.
- `WITH PricebookId`: Specifies the price book ID to return.

IN THIS SECTION:

Update an Article's Keyword Tracking with SOSL
Track keywords that are used in Salesforce Knowledge article searches with the `UPDATE TRACKING` optional clause on a SOSL query. You can use the language attribute to search by locale.

Update an Article's Viewstat with SOSL
Determine how many hits a Salesforce Knowledge article has had by using the `UPDATE VIEWSTAT` optional clause on a SOSL query. You can use the language attribute to search by locale.

WITH DATA CATEGORY DataCategorySpec
`WITH DATA CATEGORY` is an optional clause that can be added to a SOSL query to filter all search results that are associated with one or more data categories and are visible to users. This clause is used in searches of Salesforce Knowledge articles and questions.

WITH SNIPPET
`WITH SNIPPET` is an optional clause that can be added to a SOSL query for article, case, feed, and idea searches. On the search results page, excerpts below article titles show terms matching the search query highlighted within the context of surrounding text. Snippets make it easier for users to identify the content they're looking for.

Update an Article's Keyword Tracking with SOSL

Track keywords that are used in Salesforce Knowledge article searches with the `UPDATE TRACKING` optional clause on a SOSL query. You can use the language attribute to search by locale.

The `UPDATE TRACKING` clause is used to report on Salesforce Knowledge article searches and views. It allows developers to track the keywords used in Salesforce Knowledge article searches. Also, the language attribute can be used to search by a specific language (locale). However, only one language can be specified in a single query. Make a separate query for each language that you want. Use the Java format, which uses the underscore (for example, fr_FR, jp_JP, and so on), to supply locales. Search the Web for "java locale codes" to get a list of supported locales.

You can use this syntax to track a keyword used in Salesforce Knowledge article search:

```
FIND {Keyword}
RETURNING KnowledgeArticleVersion (Title WHERE PublishStatus="Online" and language="en_US")
UPDATE TRACKING
```

Update an Article's Viewstat with SOSL

Determine how many hits a Salesforce Knowledge article has had by using the `UPDATE VIEWSTAT` optional clause on a SOSL query. You can use the language attribute to search by locale.

The optional `UPDATE VIEWSTAT` clause is used to report on Salesforce Knowledge article searches and views. It allows developers to update an article's view statistics. Also, the language attribute can be used to search by a specific language (locale). However, only one language can be specified in a single query. Make a separate query for each language that you want. Use the Java format, which uses the underscore (for example, fr_FR, jp_JP, and so on), to supply locales. Search the Web for "java locale codes" to get a list of supported locales.

You can use this syntax to increase the view count for every article you have access to online in US English:

```
FIND {Title}
RETURNING FAQ__kav (Title WHERE PublishStatus="Online" and
language="en_US" and
KnowledgeArticleVersion = 'ka230000000PCiy')
UPDATE VIEWSTAT
```

WITH DATA CATEGORY DataCategorySpec

`WITH DATA CATEGORY` is an optional clause that can be added to a SOSL query to filter all search results that are associated with one or more data categories and are visible to users. This clause is used in searches of Salesforce Knowledge articles and questions.

The `WITH DATA CATEGORY` clause can be used in API version 18.0 or later.

Syntax

The `WITH DATA CATEGORY` syntax is:

```
WITH DATA CATEGORY DataCategorySpec [logicalOperator DataCategorySpec2 ... ]
```

Where `DataCategorySpec` consists of a `groupName`, `Operator`, and `category`.

Name	Description
groupName	The name of the data category group to filter. For information on category groups, see "Create and Modify Category Groups" in the Salesforce Help.
Operator	Use one of the following operators: • `AT`—Queries the specified data category.

Name	Description
	• ABOVE—Queries the specified data category and all of its parent categories.
	• BELOW—Queries the specified data category and all of its subcategories.
	• ABOVE_OR_BELOW—Queries the specified data category, all of its parent categories, and all of its subcategories.
category	The name of the category to filter. To include multiple data categories, enclose them in parentheses, separated by commas. For information on categories, see "Add Data Categories to Category Groups" in the Salesforce Help.

You can add multiple data category specifiers by using the logical operator AND. Other operators, such as OR and AND NOT, are not supported.

A SOSL statement using the WITH DATA CATEGORY clause must also include a RETURNING *ObjectTypeName* clause, with a WHERE clause that filters on the PublishStatus field.

In the RETURNING clause, specify one of the following for *ObjectTypeName*:

- To search a specific article type, use the article type name with the suffix __*kav*
- To search all article types, use KnowledgeArticleVersion
- To search questions, use Question

For information on article types, see "Knowledge Article Types" in the Salesforce Help.

The WHERE clause must use one of the following publish statuses:

- WHERE PublishStatus='online' for published articles
- WHERE PublishStatus='archived' for archived articles
- WHERE PublishStatus='draft' for draft articles

Examples

Search Type	Example
| Search all published (online)Salesforce Knowledge articles with a category from one category group. | ```
FIND {tourism} RETURNING KnowledgeArticleVersion
    (Id, Title WHERE PublishStatus='online')
    WITH DATA CATEGORY Location__c AT America__c
``` |
| Search online FAQ articles with categories from two category groups. | ```
FIND {tourism} RETURNING FAQ__kav
    (Id, Title WHERE PublishStatus='online')
    WITH DATA CATEGORY Geography__c ABOVE France__c
    AND Product__c AT mobile_phones__c
``` |
| Search archived FAQ articles from one category group. | ```
FIND {tourism} RETURNING FAQ__kav
    (Id, Title WHERE PublishStatus='archived')
    WITH DATA CATEGORY Geography__c AT Iceland__c
``` |

Search Type	Example

Search all draft Salesforce Knowledge articles from one category group.	`FIND {tourism} RETURNING KnowledgeArticleVersion`
	`    (Id, Title WHERE PublishStatus='draft')`
	`    WITH DATA CATEGORY Geography__c BELOW Europe__c`

For information on the `WITH DATA CATEGORY` clause, see the `WITH DATA CATEGORY` *filteringExpression*.

💡 **Tip:** You can also search for articles by ID, without using the `WITH DATA CATEGORY` clause. For more information, see Example WHERE Clauses.

WITH SNIPPET

`WITH SNIPPET` is an optional clause that can be added to a SOSL query for article, case, feed, and idea searches. On the search results page, excerpts below article titles show terms matching the search query highlighted within the context of surrounding text. Snippets make it easier for users to identify the content they're looking for.

📝 **Note:** To generate search results with highlighted matches but not snippets, use `WITH HIGHLIGHT`.

Search snippets and highlights are generated from the following field types.

- Email
- Text
- Text Area
- Text Area (Long)
- Text Area (Rich)

Search snippets and highlights are *not* generated from the following field types.

- Checkbox
- Currency
- Date
- Date/Time
- File
- Formula
- Lookup Relationship
- Number
- Percent
- Phone
- Picklist
- Picklist (Multi-Select)
- URL

👁 **Example:** The following SOSL statement returns snippets for articles that match the search term `San Francisco`.

```
FIND {San Francisco} IN ALL FIELDS RETURNING KnowledgeArticleVersion(id, title WHERE
PublishStatus = 'Online' AND Language = 'en_US') WITH
        SNIPPET (target_length=120)
```

The matching terms are highlighted with `<mark>` tags within the context of the snippet results. Stemmed forms of the term and any synonyms defined are also highlighted.

👁 **Example:**

```
[ {
  "attributes" : {
    "type" : "KnowledgeArticleVersion",
    "url" : "/services/data/v32.0/sobjects/KnowledgeArticleVersion/kaKD00000000001MAA"

  },
  "Id" : "kaKD00000000001MAA"
  "Title" : "San Francisco"
  "Summary" : "City and County of San Francisco"
  "snippet.text" : "<mark>San</mark> <mark>Francisco</mark>, officially the City and
County of <mark>San</mark> <mark>Francisco</mark> is the... City and County of
<mark>San</mark> <mark>Fran</mark>"
  "highlight.Title" : "<mark>San</mark> <mark>Francisco</mark>"
}, {
    "attributes" : {
      "type" : "KnowledgeArticleVersion",
      "url" : "/services/data/v32.0/sobjects/KnowledgeArticleVersion/kaBD0000000007DMAQ"

    },
    "Id" : "kaBD0000000007DMAQ",
    "Title" : "San Francisco Bay Area",
    "Summary" : "Nine county metropolitan area",
    "snippet.text" : "The <mark>SF</mark> Bay Area, commonly known as the Bay Area, is
 a populated region that"
    "highlight.Title" : "<mark>San</mark> <mark>Francisco</mark> Bay Area"
}, {
    "attributes" : {
      "type" : "KnowledgeArticleVersion",
      "url" : "/services/data/v32.0/sobjects/KnowledgeArticleVersion/ka3D0000000042OIAQ"

    },
    "Id" : "ka3D0000000042OIAQ",
    "Title" : "California",
    "Summary" : "State of California",
    "snippet.text" : "(Greater Los Angeles area and <mark>San</mark>
<mark>Francisco</mark> Bay Area, respectively), and eight of the nation's 50 most"
    } ]
```

📝 **Note:** In this example, "SF" (as a synonym defined for "San Francisco") and "San Fran" (as a stemmed form of "San Francisco") are also highlighted in the results as matching terms.

Usage

For SOSL statements using the `WITH SNIPPET` clause, we recommend using a `RETURNING` *ObjectTypeName* clause, with a `WHERE` clause that filters on the `PublishStatus` field.

In the `RETURNING` clause, specify one of the following for *ObjectTypeName*:

- To search a specific article type, use the article type name with the suffix __*kav*.
- To search all article types, use [KnowledgeArticleVersion](#).
- To search case, case comment, feed, feed comment, idea, and idea comment types, use `Case`, `CaseComment`, `FeedItem`, `FeedComment`, `Idea`, and `IdeaComment`. For example:

```
FIND {San Francisco} IN ALL FIELDS RETURNING FeedItem, FeedComment WITH SNIPPET
(target_length=120)
```

Other objects that are included in searches that contain `WITH SNIPPET` don't return snippets.

Snippets aren't displayed:

- When search terms contain a wildcard.
- When the search doesn't return any articles.
- If the user doesn't have access to the field that contains the snippet.
- When the search term doesn't appear in the first 6000 characters of the field, excluding HTML tags.

Even if you add the `WITH SNIPPET` clause, some searches don't return snippets. If an object contains more than 50 fields, snippets are sometimes not generated.

Snippets are only displayed when 20 or fewer results are returned on a page.

> **Tip:** Use the `LIMIT` clause to return only 20 results at a time.

Escaped HTML Tags

When matching terms within HTML tags are returned in a snippet, the HTML tags are escaped and the matching terms are highlighted in the results.

> **Example:** A search for *salesforce* returns an article with the text "For more information, visit salesforce.com". The original hyperlink tags from the article are escaped (encoded) and "salesforce" is highlighted in the snippet result.

```
For more information, visit &lt;a
href='https://salesforce.com'&gt;salesforce.com&lt;/a&gt;
```

Target Snippet Length

By default, each snippet displays up to approximately 300 characters, which is usually three lines of text in a standard browser window display. The number of characters displayed is the *target length*, within a statistically insignificant degree of variance.

Snippets consist of one or more *fragments* of text that contain the matching terms. If the returned snippet includes multiple text fragments (for example, for matches within multiple fields), the target length is the maximum total length of all the returned fragments.

To specify an alternate target length, add the optional `target_length` parameter to the `WITH SNIPPET` clause. You can specify a target length from 50 to 1,000 characters. When the `target_length` is set to an invalid number, such as *0* or a negative number, the length defaults to *300*.

👁 **Example:** A `target_length` parameter of 120 characters is useful for displaying a snippet of approximately three lines of text in a standard mobile interface.

```
FIND {San Francisco} IN ALL FIELDS RETURNING KnowledgeArticleVersion(id, title WHERE
PublishStatus = 'Online' AND Language = 'en_US') WITH
        SNIPPET(target_length=120)
```

Supported APIs

The `WITH SNIPPET` clause can be used in API version 32.0 or later. The `WITH SNIPPET` clause in SOSL is supported in SOAP API, REST API, and Apex.

CHAPTER 6 Salesforce Knowledge, Apex, and Visualforce

With the power of Apex and Visualforce, you can customize the look and feel of your knowledge base; for example, with custom search pages or a robust article publishing cycle.

Apex is a strongly typed, object-oriented programming language that allows developers to execute flow and transaction control statements on the server in conjunction with calls to the API. Using syntax that looks like Java and acts like database stored procedures, Apex enables developers to add business logic to most system events, including button clicks, related record updates, and Visualforce pages. Apex code can be initiated by Web service requests and from triggers on objects.

Visualforce is a framework that allows developers to build sophisticated, custom user interfaces that can be hosted natively on the Lightning Platform. The Visualforce framework includes a tag-based markup language, similar to HTML, and a set of server-side "standard controllers" that makes basic database operations, such as queries and saves, very simple to perform.

Due to the endless possibilities with Apex and Visualforce, only the Apex Publishing Service class is in this guide. For more Apex and Visualforce information see the Apex Developer Guide and the Visualforce Developer Guide.

PublishingService Class

Use the methods in the `KbManagement.PublishingService` class to manage the lifecycle of an article and its translations.

Namespace

KbManagement

Usage

Use the methods in the `KbManagement.PublishingService` class to manage the following parts of the lifecycle of an article and its translations:

- Publishing
- Updating
- Retrieving
- Deleting
- Submitting for translation
- Setting a translation to complete or incomplete status
- Archiving
- Assigning review tasks for draft articles or translations

> **Note:** Date values are based on GMT.

To use the methods in this class, you must enable Salesforce Knowledge. See Salesforce Knowledge Implementation Guide for more information on setting up Salesforce Knowledge.

IN THIS SECTION:

PublishingService Methods

PublishingService Methods

The following are methods for `PublishingService`. All methods are static.

IN THIS SECTION:

archiveOnlineArticle(articleId, scheduledDate)
Archives an online version of an article. If the specified scheduledDate is null, the article is archived immediately. Otherwise, it archives the article on the scheduled date.

assignDraftArticleTask(articleId, assigneeId, instructions, dueDate, sendEmailNotification)
Assigns a review task related to a draft article.

assignDraftTranslationTask(articleVersionId, assigneeId, instructions, dueDate, sendEmailNotification)
Assigns a review task related to a draft translation.

cancelScheduledArchivingOfArticle(articleId)
Cancels the scheduled archiving of an online article.

cancelScheduledPublicationOfArticle(articleId)

Cancels the scheduled publication of a draft article.

completeTranslation(articleVersionId)

Puts a translation in a completed state that is ready to publish.

deleteArchivedArticle(articleId)

Deletes an archived article.

deleteArchivedArticleVersion(articleId, versionNumber)

Deletes a specific archived version of a published article.

deleteDraftArticle(articleId)

Deletes a draft article.

deleteDraftTranslation(articleVersionId)

Deletes a draft translation.

editArchivedArticle(articleId)

Creates a draft article from the archived primary version and returns the new draft primary version ID of the article.

editOnlineArticle(articleId, unpublish)

Creates a draft article from the online version and returns the new draft primary version ID of the article. Also, unpublishes the online article, if `unpublish` is set to `true`.

editPublishedTranslation(articleId, language, unpublish)

Creates a draft version of the online translation for a specific language and returns the new draft primary version ID of the article. Also, unpublishes the article, if set to `true`.

publishArticle(articleId, flagAsNew)

Publishes an article. If `flagAsNew` is set to `true`, the article is published as a major version.

restoreOldVersion(articleId, versionNumber)

Creates a draft article from an existing online article based on the specified archived version of the article and returns the article version ID.

scheduleForPublication(articleId, scheduledDate)

Schedules the article for publication as a major version. If the specified date is null, the article is published immediately.

setTranslationToIncomplete(articleVersionId)

Sets a draft translation that is ready for publication back to "in progress" status.

submitForTranslation(articleId, language, assigneeId, dueDate)

Submits an article for translation to the specified language. Also assigns the specified user and due date to the submittal and returns new ID of the draft translation.

archiveOnlineArticle(articleId, scheduledDate)

Archives an online version of an article. If the specified scheduledDate is null, the article is archived immediately. Otherwise, it archives the article on the scheduled date.

Signature

```
public static Void archiveOnlineArticle(String articleId, Datetime scheduledDate)
```

Parameters

articleId
    Type: String

scheduledDate
    Type: Datetime

Return Value

Type: Void

Example

```
String articleId = 'Insert article ID';
Datetime scheduledDate = Datetime.newInstanceGmt(2012, 12,1,13,30,0);
KbManagement.PublishingService.archiveOnlineArticle(articleId, scheduledDate);
```

assignDraftArticleTask(articleId, assigneeId, instructions, dueDate, sendEmailNotification)

Assigns a review task related to a draft article.

Signature

```
public static Void assignDraftArticleTask(String articleId, String assigneeId, String
instructions, Datetime dueDate, Boolean sendEmailNotification)
```

Parameters

articleId
    Type: String

assigneeId
    Type: String

instructions
    Type: String

dueDate
    Type: Datetime

sendEmailNotification
    Type: Boolean

Return Value

Type: Void

Example

```
String articleId = 'Insert article ID';
String assigneeId = '';
String instructions = 'Please review this draft.';
Datetime dueDate = Datetime.newInstanceGmt(2012, 12, 1);
KbManagement.PublishingService.assignDraftArticleTask(articleId, assigneeId, instructions,
 dueDate, true);
```

assignDraftTranslationTask(articleVersionId, assigneeId, instructions, dueDate, sendEmailNotification)

Assigns a review task related to a draft translation.

Signature

```
public static Void assignDraftTranslationTask(String articleVersionId, String assigneeId,
String instructions, Datetime dueDate, Boolean sendEmailNotification)
```

Parameters

articleVersionId
    Type: String

assigneeId
    Type: String

instructions
    Type: String

dueDate
    Type: Datetime

sendEmailNotification
    Type: Boolean

Return Value

Type: Void

Example

```
String articleId = 'Insert article ID';
String assigneeId = 'Insert assignee ID';
String instructions = 'Please review this draft.';
Datetime dueDate = Datetime.newInstanceGmt(2012, 12, 1);
KbManagement.PublishingService.assignDraftTranslationTask(articleId, assigneeId,
instructions, dueDate, true);
```

cancelScheduledArchivingOfArticle(articleId)

Cancels the scheduled archiving of an online article.

Signature

```
public static Void cancelScheduledArchivingOfArticle(String articleId)
```

Parameters

articleId
    Type: String

Return Value

Type: Void

Example

```
String articleId = 'Insert article ID';
KbManagement.PublishingService.cancelScheduledArchivingOfArticle (articleId);
```

cancelScheduledPublicationOfArticle(articleId)

Cancels the scheduled publication of a draft article.

Signature

```
public static Void cancelScheduledPublicationOfArticle(String articleId)
```

Parameters

articleId
    Type: String

Return Value

Type: Void

Example

```
String articleId = 'Insert article ID';
KbManagement.PublishingService.cancelScheduledPublicationOfArticle (articleId);
```

completeTranslation(articleVersionId)

Puts a translation in a completed state that is ready to publish.

Signature

```
public static Void completeTranslation(String articleVersionId)
```

Parameters

articleVersionId
    Type: String

Return Value

Type: Void

Example

```
String articleVersionId = 'Insert article ID';
KbManagement.PublishingService.completeTranslation(articleVersionId);
```

deleteArchivedArticle(articleId)

Deletes an archived article.

Signature

```
public static Void deleteArchivedArticle(String articleId)
```

Parameters

articleId
    Type: String

Return Value

Type: Void

Example

```
String articleId = 'Insert article ID';
KbManagement.PublishingService.deleteArchivedArticle(articleId);
```

deleteArchivedArticleVersion(articleId, versionNumber)

Deletes a specific archived version of a published article.

Signature

```
public static Void deleteArchivedArticleVersion(String articleId, Integer versionNumber)
```

Parameters

articleId
    Type: String

versionNumber
    Type: Integer

Return Value

Type: Void

Example

```
String articleId = 'Insert article ID';
Integer versionNumber = 1;
KbManagement.PublishingService.deleteArchivedArticleVersion(articleId, versionNumber);
```

deleteDraftArticle(articleId)

Deletes a draft article.

Signature

```
public static Void deleteDraftArticle(String articleId)
```

Parameters

articleId
    Type: String

Return Value

Type: Void

Example

```
String articleId = 'Insert article ID';
KbManagement.PublishingService.deleteDraftArticle(articleId);
```

deleteDraftTranslation(articleVersionId)

Deletes a draft translation.

Signature

```
public static Void deleteDraftTranslation(String articleVersionId)
```

Parameters

articleVersionId
    Type: String

Return Value

Type: Void

Example

```
String articleVersionId = 'Insert article ID';
KbManagement.PublishingService.deleteDraftTranslation (articleVersionId);
```

editArchivedArticle(articleId)

Creates a draft article from the archived primary version and returns the new draft primary version ID of the article.

Signature

```
public static String editArchivedArticle(String articleId)
```

Parameters

articleId
    Type: String

Return Value

Type: String

Example

```
String articleId = 'Insert article ID';
String id = KbManagement.PublishingService.editArchivedArticle(articleId);
```

editOnlineArticle(articleId, unpublish)

Creates a draft article from the online version and returns the new draft primary version ID of the article. Also, unpublishes the online article, if *unpublish* is set to `true`.

Signature

```
public static String editOnlineArticle(String articleId, Boolean unpublish)
```

Parameters

articleId
    Type: String

unpublish
    Type: Boolean

Return Value

Type: String

Example

```
String articleId = 'Insert article ID';
String id = KbManagement.PublishingService.editOnlineArticle (articleId, true);
```

editPublishedTranslation(articleId, language, unpublish)

Creates a draft version of the online translation for a specific language and returns the new draft primary version ID of the article. Also, unpublishes the article, if set to `true`.

Signature

```
public static String editPublishedTranslation(String articleId, String language, Boolean
unpublish)
```

Parameters

articleId
    Type: String

language
    Type: String

unpublish
    Type: Boolean

Return Value

Type: String

Example

```
String articleId = 'Insert article ID';
String language = 'fr';
String id = KbManagement.PublishingService.editPublishedTranslation(articleId, language,
true);
```

publishArticle(articleId, flagAsNew)

Publishes an article. If *flagAsNew* is set to `true`, the article is published as a major version.

Signature

```
public static Void publishArticle(String articleId, Boolean flagAsNew)
```

Parameters

articleId
    Type: String

flagAsNew
    Type: Boolean

Return Value

Type: Void

Example

```
String articleId = 'Insert article ID';
KbManagement.PublishingService.publishArticle(articleId, true);
```

restoreOldVersion(articleId, versionNumber)

Creates a draft article from an existing online article based on the specified archived version of the article and returns the article version ID.

Signature

```
public static String restoreOldVersion(String articleId, Integer versionNumber)
```

Parameters

articleId
    Type: String

versionNumber
    Type: Integer

Return Value

Type: String

Example

```
String articleId = 'Insert article ID';
String id = KbManagement.PublishingService.restoreOldVersion (articleId, 1);
```

scheduleForPublication(articleId, scheduledDate)

Schedules the article for publication as a major version. If the specified date is null, the article is published immediately.

Signature

```
public static Void scheduleForPublication(String articleId, Datetime scheduledDate)
```

Parameters

articleId
   Type: String

scheduledDate
   Type: Datetime

Return Value

Type: Void

Example

```
String articleId = 'Insert article ID';
Datetime scheduledDate = Datetime.newInstanceGmt(2012, 12,1,13,30,0);
KbManagement.PublishingService.scheduleForPublication(articleId, scheduledDate);
```

setTranslationToIncomplete(articleVersionId)

Sets a draft translation that is ready for publication back to "in progress" status.

Signature

```
public static Void setTranslationToIncomplete(String articleVersionId)
```

Parameters

articleVersionId
   Type: String

Return Value

Type: Void

Example

```
String articleVersionId = 'Insert article ID';
KbManagement.PublishingService.setTranslationToIncomplete(articleVersionId);
```

submitForTranslation(articleId, language, assigneeId, dueDate)

Submits an article for translation to the specified language. Also assigns the specified user and due date to the submittal and returns new ID of the draft translation.

Signature

```
public static String submitForTranslation(String articleId, String language, String
assigneeId, Datetime dueDate)
```

Parameters

articleId
    Type: String

language
    Type: String

assigneeId
    Type: String

dueDate
    Type: Datetime

Return Value

Type: String

Example

```
String articleId = 'Insert article ID';
String language = 'fr';
String assigneeId = 'Insert assignee ID';
Datetime dueDate = Datetime.newInstanceGmt(2012, 12,1);
String id = KbManagement.PublishingService.submitForTranslation(articleId, language,
assigneeId, dueDate);
```

CHAPTER 7 Salesforce Knowledge UI API

Retrieve and update data and metadata of Salesforce Knowledge records with the UI API to help build your Lightning Experience UI for your mobile or custom web app.

📝 Note: UI API isn't supported for Classic Knowledge.

To learn how to use the UI API, go to the User Interface API Developer Guide.

You can find supported Knowledge objects and how to use them in the Supported Objects section of the UI API Developer Guide.

- All Supported Objects—CaseArticle, KnowledgeArticleVersion, and LinkedArticle
- List View Supported Objects—KnowledgeArticleVersion
- Most Recently Used List View Supported Objects—KnowledgeArticleVersion

UI API Limitations

UI API with Lightning Knowledge has these limitations.

RecordTypeId of Linked Record Shows as Null When It Shouldn't

To retrieve the `recordTypeId` of a `LinkedArticle.LinkedEntity` record using the Get a Record UI API, replace `Id` with `RecordTypeId`. This limitation applies to API version 57.0 and later.

```
// This code doesn't retrieve the correct recordTypeId.
GET
/services/data/v57.0/ui-api/records/{LinkedArticleId}?fields=LinkedArticle.LinkedEntity.Id
// This code retrieves the correct recordTypeId.
GET
/services/data/v57.0/ui-api/records/{LinkedArticleId}?fields=LinkedArticle.LinkedEntity.RecordTypeId
```

The response shows the record type ID in two places.

- `LinkedEntity | value | fields | RecordTypeId | value`—Shows the correct record type ID.

- `LinkedEntity | value | recordTypeId`—Shows null even if there's an associated record type.

```
{
    "apiName": "LinkedArticle",
    "childRelationships": {},
    "eTag": "0a4b81f3e165e7feefcc2f0e0f30e6da",
    "fields": {
        "LinkedEntity": {
            "displayValue": "00000001",
            "value": {
                "apiName": "Name",
                "childRelationships": {},
                "eTag": "0917224c30df4322d4e5abbda36a9c25",
                "fields": {
                    "Id": {
                        "displayValue": null,
                        "value": "0WORM0000006nan4AA"
                    },
                    "RecordTypeId": {
                        "displayValue": null,
                        "value": "012RM0000005Yl2YAE"
                    }
                },
                "id": "0WORM0000006nan4AA",
                "lastModifiedById": null,
                "lastModifiedDate": null,
                "recordTypeId": null,
                "recordTypeInfo": null,
                "systemModstamp": null,
                "weakEtag": 0
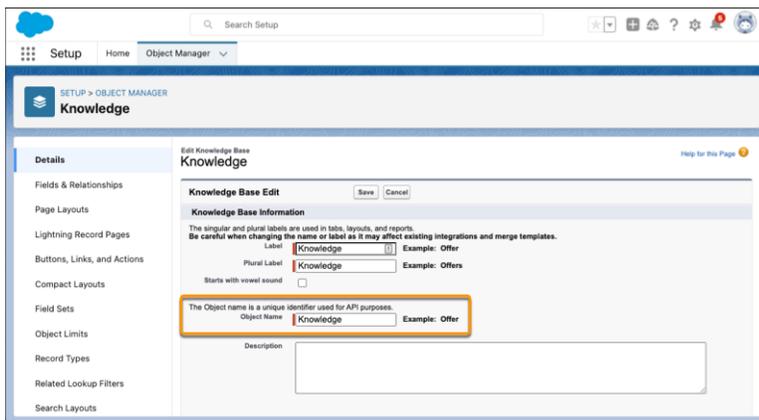            }
        },
        "LinkedEntityId": {
            "displayValue": null,
```

```
            "value": "0WORM0000006nan4AA"
        }
    },
    "id": "1WKRM0000004MKA4A2",
    "lastModifiedById": "005RM000002Z4dkYAC",
    "lastModifiedDate": "2022-09-28T18:12:10.000Z",
    "recordTypeId": "012000000000000AAA",
    "recordTypeInfo": null,
    "systemModstamp": "2022-09-28T18:12:10.000Z",
    "weakEtag": 1664388730000
}
```

object-info UI API Returns an Error

Changing the **Object Name** of the Knowledge object in **Object Manager** results in an error when you run the Get Object Metadata UI API for `LinkedArticle` or `CaseArticle`. Don't change the Knowledge object name. This limitation applies to API version 57.0 and later.

For example, you change the Knowledge object name and run these UI APIs.

```
GET /services/data/v57.0/ui-api/object-info/LinkedArticle
GET /services/data/v57.0/ui-api/object-info/CaseArticle
```

You get an `INSUFFICIENT_ACCESS` error.

```
{
    "errorCode":"INSUFFICIENT_ACCESS",
    "message":"You don't have access to this record. Ask your administrator for help or
to request access."
}
```

Retrieving KnowledgeArticleVersion.RecordTypeId Fails

To retrieve the `KnowledgeArticleVersion.RecordTypeId` of `LinkedArticle` or `CaseArticle` records using the Get a Record UI API, use the `optionaFields` not `fields` request parameter. This limitation applies to API version 57.0 and later.

```
GET
/services/data/v57.0/ui-api/records/{LinkedArticleId}?optionalFields=LinkedArticle.KnowledgeArticleVersion.RecordTypeId
```

```
GET
/services/data/v57.0/ui-api/records/{CaseArticleId}?optionalFields=CaseArticle.KnowledgeArticleVersion.RecordTypeId
```

Here's the error that you get when you try to retrieve this `RecordTypeId` using the `fields` request parameter.

```
{
    "errorCode": "INVALID_FIELD",
    "message": "INVALID_FIELD: \nSELECT LastModifiedDate,
KnowledgeArticleVersion.RecordTypeId\n                              ^\nERROR at
Row:1:Column:26\nNo such column 'RecordTypeId' on entity 'KnowledgeArticleVersion'. If you
 are attempting to use a custom field, be sure to append the '__c' after the custom field
 name. Please reference your WSDL or the describe call for the appropriate names."
}
```

Record Data and Metadata of Related Lists Don't Show Up

If the related object isn't added to the page layout of the parent record, UI API doesn't return the record data or metadata of related lists. This limitation applies to API version 57.0 and later.

The affected endpoints are:

- Get Related List Records with URL Parameters
- Get Related List Records with a Request Body
- Get Related List Metadata

If the related object is missing from the page layout, you encounter one of these problems.

- `NVALID_TYPE` error code
- `UNKNOWN_EXCEPTION` error code
- UI API returns the wrong record, for example, the LinkedArticle record instead of Article

To retrieve the correct record data or metadata with the related list UI API endpoints, add the related object to the parent object's page layout.

```
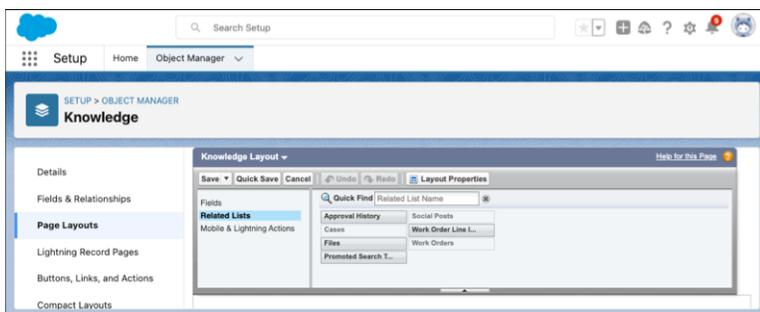/ui-api/related-list-records/{parentRecordId}/{relatedListId}
/ui-api/related-list-info/{parentObjectApiName}/{relatedListId}
```

To add a related list object, go to **Object Manager** | **{PARENT_OBJECT}** | **Page Layouts** | **Related Lists**.

For example, to retrieve the `Articles` related list with these UI API calls, add the **Articles** related list to the **WorkOrder**, **WorkOrderLineItem**, and **Case** page layouts. Apply this solution to other parent object types that **Articles** relate to.

```
GET /services/data/v57.0/ui-api/related-list-records/{WorkOrderId}/LinkedArticles
GET /services/data/v57.0/ui-api/related-list-records/{WorkOrderLineItemId}/LinkedArticles
```

```
GET /services/data/v57.0/ui-api/related-list-info/{WorkOrderId}/LinkedArticles
GET /services/data/v57.0/ui-api/related-list-info/{WorkOrderLineItemId}/LinkedArticles
GET /services/data/v57.0/ui-api/related-list-info/{CaseId}/CaseArticles
```

To retrieve related list records related to Knowledge articles with these UI API calls, add the **WorkOrder**, **WorkOrderLineItem**, and **SocialPosts** related objects to the **Knowledge** page layout.

```
GET
/services/data/v57.0/ui-api/related-list-records/{KnowledgeArticleVersionId}/LinkedWorkOrders
GET
/services/data/v57.0/ui-api/related-list-records/{KnowledgeArticleVersionId}/LinkedWorkOrderLineItems
GET
/services/data/v57.0/ui-api/related-list-records/{KnowledgeArticleVersionId}/LinkedSocialPosts

GET
/services/data/v57.0/ui-api/related-list-info/{KnowledgeArticleVersionId}/LinkedWorkOrders
GET
/services/data/v57.0/ui-api/related-list-info/{KnowledgeArticleVersionId}/LinkedWorkOrderLineItems
GET
/services/data/v57.0/ui-api/related-list-info/{KnowledgeArticleVersionId}/LinkedSocialPosts
```

To retrieve record data or metadata of cases related to Knowledge articles, add the **Cases** related list to the **Knowledge** page layout.

```
GET /services/data/v57.0/ui-api/related-list-info/{Knowledge__kavId}/CaseArticles
```

INDEX

Q

query() call
    SOQL overview

S

SOQL
    overview
SOSL
    about