

# Handshake User Guide

*Prepared By: Perficient, Inc.  
© 2020 Perficient, Inc.  
All rights reserved.*

## Table of Contents

Prerequisites .....	3
Getting Started .....	3
Web UI URL.....	3
Creating a Connector.....	3
Connector Configuration .....	4
Setting a Schedule for a Connector .....	6
Enabling a Connector .....	7
Source Configurations.....	7
Database Source.....	7
FileNet Source .....	9
GCS .....	12
Transformation Configurations.....	14
ContentFilter Transformer .....	14
Case Transformer .....	14
Match Transformer .....	15
Metadata Create Transformer .....	15
Remove Metafields Transformer .....	16
Replace URL Transformer.....	16

## Prerequisites

This user guide assumes that the user has completed the installation of Handshake and has validated the application is running as expected. Please refer to the Deployment Guide for information on how to complete this task.

## Getting Started

Handshake provides a user-friendly interface for creating connector configurations on the fly. After Handshake is installed and validated, this interface may be accessed via the web, as long as the target workstation has the correct network access to the Handshake hosting server.

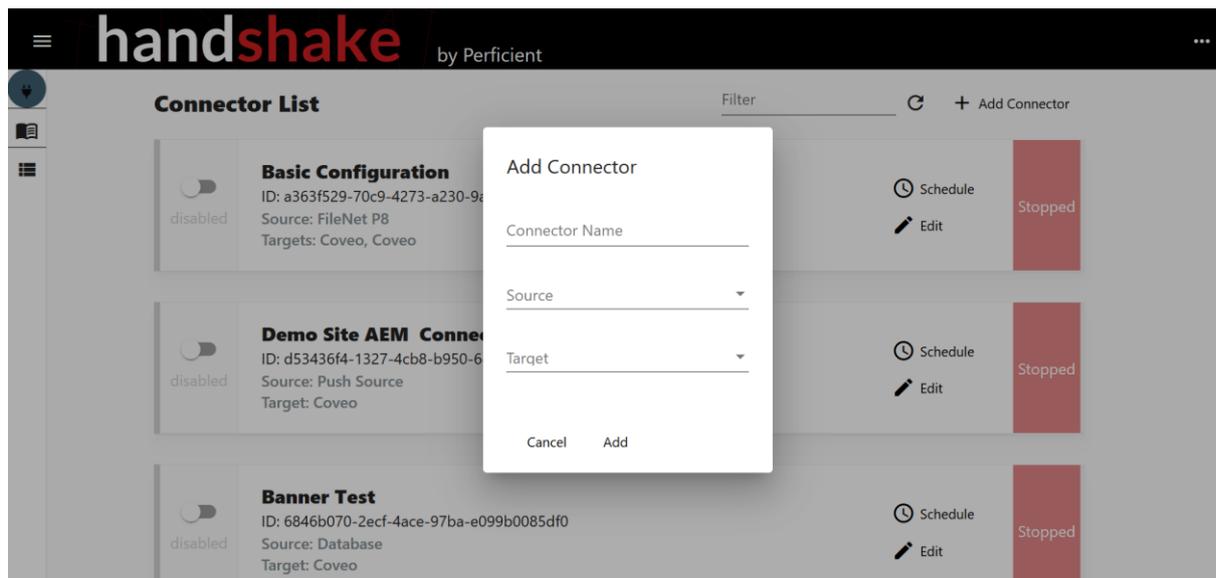
### Web UI URL

The default URL for accessing the web UI is <http://handshake.servername:8080/handshake/>

This URL may vary based on installation choices. Confirm your web UI URL with your Handshake administrator.

The main components of a search connector are the Source system (where searchable content is coming from), Transformations (updates to the content metadata, content to exclude, etc.) and the target (where the search index will reside). All of these items can be created and managed in the Web UI, as well as scheduling runs of the connector.

### Creating a Connector



You can create as many connectors as needed to effectively build out data pipelines from source to target. To add a new connector:

1. From the main Handshake dashboard, click "+ Add Connector" in the top right.
2. Enter a Name for the Connector
3. From the dropdown, select a source type.
4. From the dropdown, select a base target type (you can add more later).
5. Click 'Add.' You will be brought to the Connector Configuration screen.
6. Later, you can return to the Connector Configuration screen by clicking the 'Edit' label for a connector in the Disabled state..

## Connector Configuration

The screenshot shows the 'handshake by Perficient' interface for configuring a connector named 'Secure Content: P8 to GCS, Coveo'. The interface is divided into three main sections:

- Source:** A dropdown menu currently showing 'FileNet P8'.
- Transformations:** A list of transformation types with a '+ Add' button. The list includes 'Coveo DocumentId Transformer', 'Content Filter Transformer' (highlighted in blue), and another 'Content Filter Transformer'.
- Targets:** A list of target types with a '+ Add' button. The list includes 'Google Cloud Search' and 'Coveo'.

The right-hand side of the interface is the 'Configuration' section, which features a code editor with the following JSON configuration:

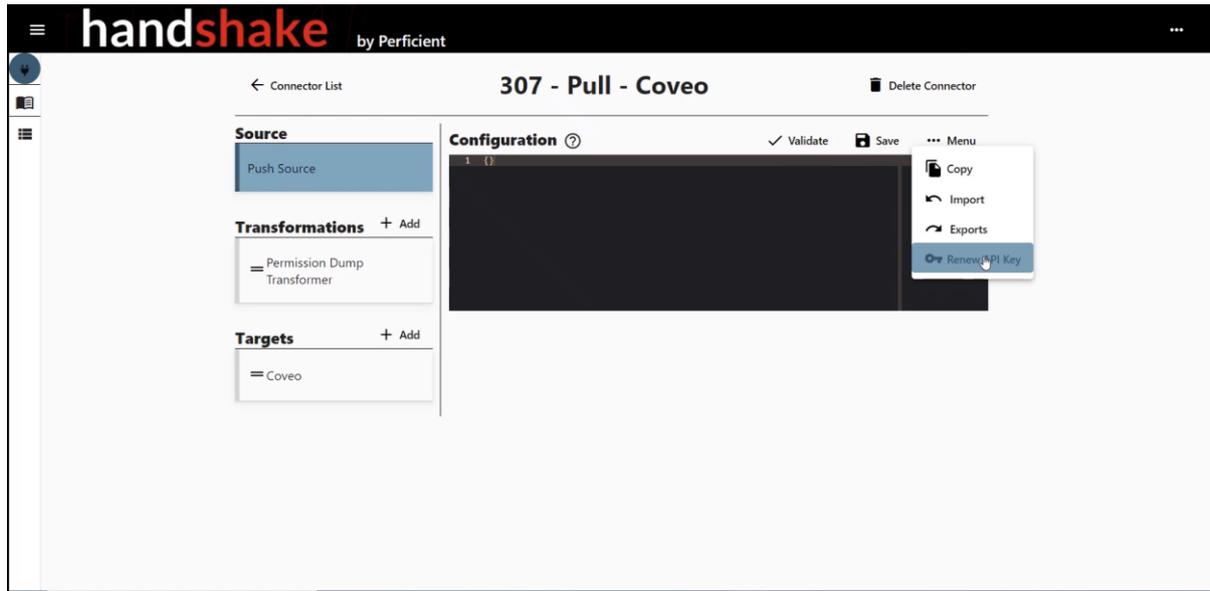
```
1 {  
2   "mode": "contentFilter",  
3   "filterField": "",  
4   "sourceValue": "",  
5   "filterCriteria": "metadata[*DocumentTitle].contains(\"drake\")"  
6 }
```

At the top right of the configuration section, there are buttons for 'Validate', 'Save', and 'Menu'.

Each connector type has unique configuration parameters. See the Source and Target configuration sections for specific configuration details. Even if a connector is not running, *a connector cannot be edited if it is enabled*. Disable a connector to change its configuration:

1. On the left hand card view, single click the card to see its configuration.
2. New Transformations and Targets can be added by clicking the '+' icon to right of the section label. New elements are added to the bottom of the list, with their default configuration.
3. Within Transformation and Target sections, elements can be moved by dragging them. Transforms and Targets execute in sequence.

Configuration Edit



When a card is selected, a Configuration Editor displays the current configuration.

1. Configuration is stored as a JSON. Configuration must be written in standard JSON or the Save 'label' will not be enabled.
2. When you have finished editing a Configuration element, click 'Save' to save your changes. Changes must be saved before validating a configuration.
3. To test that a configuration is valid, click the 'Validate' label. Handshake will attempt test that the configuration is valid. In the case of Sources and Targets, Handshake will attempt to connect to the source system.
4. The dropdown menu provides options to copy configuration elements to other connector configurations and to import or export a configuration.
5. 'Delete Connector' will delete the connector configuration.

## Setting a Schedule for a Connector

The screenshot shows the 'Secure Content: P8 to GCS, Coveo' connector configuration page. At the top, there is a 'disabled' toggle switch, the connector ID 'f8aeefd8-effa-4bab-98d8-8342d24f9142', source 'FileNet P8', and targets 'Google Cloud Search, Coveo'. A 'Schedule' button and an 'Edit' button are visible. A red 'Stopped' status bar is on the right. Below this is the 'handshake by Perficient' header. The main content area shows a 'Connector List' with a 'Paused' status. A 'Schedules' table lists a 'full' traversal with details: Last Run: 07/31/2020 12:54 PM, Next: PM, Run: 07/31/2020 12:55 PM, and Cron: 0 \*\*\*\* ? \*. An 'Add Traversal' modal is open, showing a 'Traversal Name' input field, radio buttons for 'Full' and 'Incremental', and 'Cancel' and 'Add' buttons.

Traversals manage the schedule for when a connector runs, and can either fully or incrementally traverse the source content. Traversals are only scheduled for a pull source.

Once a connector has been configured, you can configure the traversal. To do so, navigate to the appropriate connector, click Schedule, and Add Traversal.

Click on a card to edit the frequency of the job. You have 5 GUI assisted options to set time, or use advance to write your own CRON expression.

### *Full Traversals*

Full traversals read through and process all items in the configured source. Items processed in this way will be marked as “full” and provides instructions to the target to be added to a new index. Full traversals are needed to capture deleted items. Full traversals operate in the following manner.

1. Reads checkpoint to compute next runtime
2. Updates Object and Scheduling Schemas per object or traversal
3. Runs on a CRON scheduler

### *Incremental Traversals*

Incremental traversal must be implemented for sources in a certain way that is suitable for every source. They will identify the last object processed during the previous run, and will begin processing items from where that run left off. Incremental traversals operate in the following manner.

1. Gets Last incremental date, incremental status, incremental checkpoint JSON
2. Computes and schedules next incremental run
3. Runs on a CRON scheduler

## Enabling a Connector



Once a connector has a schedule and a complete configuration, click the Enable/Disable toggle from the main Handshake dashboard. The connector will execute on its traversal timer. Note: schedules may be modified while a connector is enabled, but configuration may not. Disable a connector to edit its configuration. The right of the card will turn green and show that the connector is running.

## Source Configurations

### Nuxeo Source

**Source Type:** PULL

**Requires Scheduling:** Yes

The Nuxeo Source leverages Nuxeo APIs to connect to a Nuxeo instance and assemble content, metadata, and security for each searchable content item. Metadata fields to send to search are defined in the NxQL or Query Parameters configuration snippets and also allow users to limit what content a Connector Pipeline will push through Handshake.

#### Configuration

```
batchsize": "",
"connection": {
  "uri": "",
  "username": "",
  "passwordOrApiKey": ""
},
"NxQL": "",
"queryParameters": [
  ""
],
"SecurityEnabled": "T/F"
```

### Database Source

**Source Type:** PULL

**Requires Scheduling:** Yes

The Database Source supports extracting SQL data from supported databases. OOTB, the source builds with the following drivers.

- \* ojdbc8
- \* mysql-connector-java
- \* postgresql
- \* mssql-jdbc
- \* sqlite-jdbc

#### *Configuration*

```
{
  "db.driverClass": "fully qualified driver class name",
  "db.url": "Database URL",
  "db.user": "username",
  "db.password": "password",
  "db.allRecordsSql": "SQL query ended with ",
  "db.allColumns": "all columns to be included to crawl",
  "db.contentColumns": "content columns",
  "db.uniqueKeyColumns": "unique key column",
  "url.columns": "id",
  "contentTemplate.db.title": "db",
  "contentTemplate.db.quality.high": "high quality columns",
  "contentTemplate.db.quality.medium": "medium quality columns",
  "contentTemplate.db.quality.low": "low quality columns",
  "db.blobColumn": "blob columns if any",
  "db.allRecordsSql.pagination": "values offset or none for no
pagination",
  "db.deletedColumnName": "priority",
  "db.incrementalActionDeleteValue": "deleted column value"
}
```

#### *Example Configuration*

```
{
  "db.driverClass": "org.apache.derby.jdbc.EmbeddedDriver",
  "db.url": "jdbc:derby:/opt/handshake/handshake-quartz",
  "db.user": "",
  "db.password": "",
  "db.allRecordsSql": "select sched_name, job_name, job_group,
description, job_class_name, is_durable, is_nonconcurrent, is_update_data,
requests_recovery, job_data FROM qrtz_job_details",
  "db.allColumns": "sched_name, job_name, job_group, description,
job_class_name, is_durable, is_nonconcurrent, is_update_data,
requests_recovery, job_data",
  "db.contentColumns": "",
  "db.uniqueKeyColumns": "sched_name",
  "url.columns": "sched_name",
  "contentTemplate.db.title": "sched_name",
  "contentTemplate.db.quality.high": "sched_name, job_name, job_group",
  "contentTemplate.db.quality.medium": "description, job_class_name",
  "contentTemplate.db.quality.low": "is_durable, is_nonconcurrent,
is_update_data, requests_recovery",
  "db.blobColumn": "",
  "db.allRecordsSql.pagination": "none",
  "db.deletedColumnName": "",
  "db.incrementalActionDeleteValue": ""
}
```

## FileNet Source

**Source Type:** PULL

**Requires Scheduling:** Yes

FileNet is traversed using Content Engine Web Service REST APIs. As such, installation is dependent on FileNet's deployment and security configuration. Some instances, Handshake can be installed anywhere and connection is brokered through a service account. Other instances, Handshake must be installed behind the Firewall or collocated on the same server to connect.

### Configuration

```
"ceurl": "hostname/webservicePath/webservicesEndpoint",
"username": "serviceAccount",
"password": "serviceAccount password",
"objectstore": "objectStoreSymbolicName",
"sql": "Selection criteria is generated from the ACCE search view. See note below"
```

**Note:** To generate the SQL statement, use the search interface in ACCE, then click SQL view to get a valid SQL string. This parameter allows you to control the fields to send to handshake, as well as document classes or folders to crawl. In addition, `#{CONTENT_ELEMENTS}` must be included in the select statement, though they are not searchable in ACCE.

### Example Configuration

```
"ceurl": "https://<host>/wsi/FNCEWS40MTOM",
"username": "",
"password": "",
"objectstore": "OS1",
"sql": "SELECT [This], [RPGEdition], [RPGPremium], [RPGSourceBook],
[DocumentTitle], [Name], [Owner], [Id], [DateLastModified],
$CONTENT_ELEMENTS FROM [RPGContent] ORDER BY [DateLastModified] ASC"
```

## Adobe Experience Manager {#SourceAdobe}

**Source Type:** PUSH **Requires Scheduling:** No AEM requires a local component installed on the AEM server.

### Configuration

The Source configuration for a push source (such as AEM), does not require any input values. Use the following steps to create the connection

1. In Handshake UI, create a new connector with source type PUSH (no parameters required)
2. Note the 'ID' that is created

3. Login to AEM and go to tools -> General -> Search Connector -> Handshake Remote connection. Point to Handshake using the Connector ID you got in previous steps, server URL for Handshake Services, and a (currently hardcoded) API Token.\
4. In AEM, admins can control what content paths are valid for publishing, what fields to exclude, and what service account to use for publishing. Use the Converter Configuration to exclude fields;\
5. Use content Rules to set include and exclude rules. All content that falls within the included content filter is send to Handshake whenever it is published:\
6. Users can also synchronize (send to Coveo) already published content by specifying a path structure in Bulk Export\

#### Test Configuration (in AEM)

Server URL: `http://handshake.prftdata.com:8080/handshake-services`  
Connector ID: ConnectorID  
Authentication Token: `2e82d0ee-fc41-4f83-89d1-fc4d3d77b438`

#### JIRA (On Prem or Cloud) {#SourceJIRA}

**Source Type: PULL Requires Scheduling:** Yes JIRA connector can traverse both on prem or cloud deployments of JIRA. On prem must typically be deployed collocated on the JIRA server, or connect through a whitelisted proxy / on a VPN connection. Cloud permissions are managed by an API token.

#### Configuration

```
"jiraUrl": "host",
"jiraUsername": "serviceAccount. No Domain is needed (i.e. zachary.fischer is valid, zachary.fischer@perficient.com is invalid)",
"jiraPasswordOrToken": "Password is used for on Prem. Token is used for cloud. To generate a token for your user, go to this URL and create one for your user (it will be valid for any cloud space your user has priveleges in): https://id.atlassian.com/manage-profile/security/api-tokens",
"jiraIsCloud": "true/false",
"jiraApiUrl": "/rest/api/2",
"jiraFilterProjects": "true: control which projects. false: get everything",
"jiraProjectsToExclude": "Project ID to exclude",
"jiraProjectsToInclude": "Project ID to include",
"jiraIndexAttachments": "true/false",
"jiraAttachmentMimeTypesToIgnore":
"image/png,image/jpg,image/jpeg,image/tiff",
"jiraAppendCommentsToContent": "true/false",
"jiraIndexComments": "true/false"
```

#### Test JIRA Cloud - Handshake project

```
"jiraUrl": "https://prftdata.atlassian.net/",
"jiraUsername": "username",
"jiraPasswordOrToken": "apiToken",
"jiraIsCloud": "true",
"jiraApiUrl": "/rest/api/2",
"jiraFilterProjects": "true",
```

```
"jiraProjectsToExclude": "",  
"jiraProjectsToInclude": "SCF",  
"jiraIndexAttachments": "false",  
"jiraAttachmentMimeTypesToIgnore":  
"image/png,image/jpeg,image/jpg,image/tiff",  
"jiraAppendCommentsToContent": "false",  
"jiraIndexComments": "false"
```

#### Test JIRA On Prem - Trex Project. Must be on VPN

```
"jira.url": "https://jira.perficient.com",  
"jira.username": "username",  
"jira.passwordOrToken": "password",  
"jira.isCloud": "false",  
"jira.api.url": "/rest/api/2",  
"jira.filterProjects": "true",  
"jira.projectsToExclude": "",  
"jira.projectsToInclude": "TX",  
"jira.indexAttachments": "false",  
"jira.attachmentMimeTypesToIgnore":  
"image/png,image/jpeg,image/jpg,image/tiff",  
"jira.appendCommentsToContent": "false",  
"jira.indexComments": "false"
```

#### Confluence (On Prem or Cloud) {#SourceConfluence}

**Source Type:** PULL **Requires Scheduling:** Yes Confluence connector can traverse both on prem or cloud deployments of Confluence. On prem must typically be deployed collocated on the Confluence server, or connect through a whitelisted proxy / on a VPN connection. Cloud permissions are managed by an API token.

#### Configuration

```
"confluence-url": "host/wiki/ for cloud, host for on prem",  
"username": "username@domain",  
"password": "token for cloud, password for on prem",  
"spacekey": "Space ID to filter",  
"filterContentByStartDateString": "",  
"filterContentByEndDateString": ""
```

#### Test Configuration: Cloud Confluence

```
"confluence-url": "https://shaking-hands.atlassian.net/wiki/",  
"username": "username@domain.com",  
"password": "y49AUn77pzodzZpQfncLD008"
```

#### Norconex File System {#SourceFile}

**Source Type:** PUSH **Requires Scheduling:** Yes (in config properties file) Norconex is not packaged with handshake-services.war. It is built from the filesystem-source project by invoking:

```
sudo mvn clean install -Dmaven.test.skip=true -Pprod
```

The JAR must be deployed on the same file system it will crawl.

A PUSH connector must be configured and enabled in Handshake for the committer to work.

You must provide a configuration file for the connector. This configuration file (for example: filesystem-config.properties) is supplied to the connector via a command line argument - Dconfig=filesystem-config.properties

The file must contain below information for connector to run successfully

```
file-path=D:\foo
cron-schedule=0 14 * * * ? *
connector-id=connectorId
push-uri= http://localhost:8080/handshake-
services/connectors/connectorId/documents
push-token =2e82d0ee-fc41-4f83-89d1-fc4d3d77b438
```

In the jar directory, run:

```
java -cp filesystem-source-1.0-SNAPSHOT.jar
com.perficient.handshake.components.NorconexFileCrawlJob -
Dconfig=filesystem-config.properties
```

## Target Configurations

### Coveo

Coveo requires a PUSH source be created to receive content:

1. Navigate to <https://platform.cloud.coveo.com/login>, log in with your credentials, and select the ID of an existing source or create a new one by going to Sources, Clicking "Add Source", and selecting "Push"
2. Ensure you Name the source and Select "Create an API key"
3. Copy and save the API key from the dialog. You will need this for your configuration in Handshake.
4. To find your Organization ID, view the dropdown list of organizations in the top of your Coveo administration panel.

To find your source ID, find the source in the Coveo Source List. Note the ID listed under your source name. Additionally, if you want fields from your source to be recognized as fields in Coveo, create them in the Coveo console by doing to Fields -> Add Field, and giving them the same name.

### Configuration

```
"endPoint": "https://push.cloud.coveo.com/v1",
"uri": "https://push.cloud.coveo.com/v1",
"organizationId": "orgId in Coveo",
"apiKey": "Source specific API Token or Organization API Token",
"sourceId": "source ID is visible under the Name of a Source in the Source
Menu",
"securityProvider": "DummyValue"
```

Example Configuration for FileNet Source (though these are recreated often, you're better off making your own)

```
"endPoint": "https://push.cloud.coveo.com/v1",  
"uri": "https://push.cloud.coveo.com/v1",  
"organizationId": "prftthierno0a47qyl0",  
"apiKey": "xxb7d9ff28-5b65-4b65-927c-61a94631c924",  
"sourceId": "prftthierno0a47qyl0-vdmqp5mmodphg43zu3ivv5cc44",  
"securityProvider": "FileNetSecurityProvider"
```

## GCS

Google Cloud Search has a lot of details in their configuration. Contact your GCS administrator to get the following keys for your instance:

### Example Configuration

```
"preferredBatchSizeItems": "20",  
"googleProperties":  
  "api.sourceId": "",  
  "defaultAcl.public": true,  
  "itemMetadata.objectType": "RPGContent"  
,  
"serviceAccountPrivateKey":  
  "type": "service_account",  
  "project_id": "handshake-277814",  
  "private_key_id": "",  
  "private_key": "-----BEGIN PRIVATE KEY-----\n-----END PRIVATE KEY-----  
\n",  
  "client_email": "",  
  "client_id": "",  
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",  
  "token_uri": "https://oauth2.googleapis.com/token",  
  "auth_provider_x509_cert_url":  
"https://www.googleapis.com/oauth2/v1/certs",  
  "client_x509_cert_url":  
https://www.googleapis.com/robot/v1/metadata/x509/handshaketarget%40handshake-277814.iam.gserviceaccount.com
```

### File System Target {#TargetFile}

Sends content to an absolute path where Handshake is installed. Separates content and metadata into two JSON files when writing to the target.

#### Configuration

```
"directory-path": "Absolute/Path/To/Folder"
```

#### Test Configuration, writing to S3 Bucket on EC2

```
"directory-path": "/opt/handshak3/destination"
```

## Transformation Configurations

Transformations allow data conversions and filtering to be performed by Handshake, after data is collected from the source and *before* it is sent to the target location. This means that you can use transformations to manipulate and redact sensitive data during processing, never allowing it to leave your Handshake server.

### *Filter Criteria Spring Expression Language (SpEL)*

Each of the following transformations allow for an additional parameter called `filterCriteria`, which allows you to add an additional filter on what content should be subject to the transformation.

```
"filterCriteria": "metadata['jcr:title'].equals('Equipment')
```

Any content that matches the `filterCriteria` will undergo the transformation as specified.

The `filterCriteria` parameter utilizes the Spring Expression Language (SpEL). Refer to <https://docs.spring.io/spring/docs/3.0.x/reference/expressions.html> for more information on how to write SpEL statements.

### ContentFilter Transformer

The Content Filter transformer will stop a content item which meets the given criteria from being processed by Handshake and drops the message in flight. Spring `filterCriteria` are also accepted by this transform.

```
{
  "mode": "contentFilter", <always set
  "filterField": "Field to filter on",
  "sourceValue": "a specific string to match in that field"
}
```

### Case Transformer

The Case Transformer will convert a designated field to all upper or lower case.

```
{
  "mode": "upper",
  "fields": [
    "fieldName1"
  ]
}
```

parameter	Required	Options
<b>mode</b>	Yes	'upper' or 'lower'
<b>fields</b>	Yes	comma separated list of field names <b>Note:</b> only fields of type string should be designated for this transformer. Numeric, date, and other data types are not supported.

## Match Transformer

The Match performer will find strings matching the input regex statement or string, and replace it with the required string. The output of the transform can be an in place transformation, or a copy to a new metadata value, leaving the original value intact.

```
{
  "mode": "",
  "field": "fieldName1",
  "outputField": "value",
  "sourceValue": "value",
  "replaceValue": "replaceValue1"
}
```

parameter	Required	Options
<b>mode</b>	Yes	"caseSensitive": looks for an exact string or integer match, including case sensitivity "caseInsensitive": looks for an exact string or integer match, case insensitive "regexString": looks for a regex match
<b>field</b>	Yes	A field name of a string type field, which will be used to find a sourceValue match. Example: documentId
<b>outputField</b>	No	Location to generate output of the transformation. If not specified, the transformation output will overwrite values in the field
<b>sourceValue</b>	Yes	Match criteria for the specified field, either in regex format or text format (depending on mode specified)
<b>replaceValue</b>	Yes	String used to replace the sourceValue match

## Metadata Create Transformer

The Metadata Create Transformer will create a new metadata field for the search object, based on a given pattern and source fields.

```
{
  "pattern": "",
  "newField": "",
  "sourceFields": "",
  "toReplace": ""
}
```

parameter	required	options
<b>pattern</b>	Yes	a construction pattern with placeholder such as {0}, {1} to indicate where metadata values/strings should be substituted

parameter	required	options
<b>newField</b>	Yes	Metadata field name where the new value should be stored
<b>sourceFields</b>	Yes	Comma separated list of values (strings or metadata field names), to be used to replace placeholders in pattern

## Remove Metafields Transformer

The Remove Metafields Transformer will remove metadata fields from a content object before sending it for indexing.

```
{
  "RemoveFields": [
    "fieldName1",
    "fieldName2"
  ]
}
```

parameter	Required	Options
<b>RemoveFields</b>	Yes	An array of field names that should be removed from content before being sent to target. May not include the id field.

## Replace URL Transformer

The Replace URL Transformer will replace parts of a URL with a replacement string.

```
{
  "mode": "",
  "field": "",
  "urlProperties": {
    "urlProtocol": "",
    "urlHost": "",
    "urlFile": "",
    "urlPath": "",
    "urlQuery": ""
  }
}
```

parameter	Required	Options
<b>mode</b>	Yes	??
<b>field</b>	Yes	A field of metadata that contains the original URL (must be in valid URL format)
<b>urlProperties</b>	Yes	One or more of the following properties must be included

<b>parameter</b>	<b>Required</b>	<b>Options</b>
<b>urlProtocol</b>	No	http, https, or another protocol to be replaced in the metadata field URL
<b>urlHost</b>	No	Host name to be replaced in the metadata field URL
<b>urlFile</b>	No	File to be replaced in the metadata field URL
<b>urlPath</b>	No	Path to be replaced in the metadata field URL
<b>urlQuery</b>	No	Query parameters to be replaced in the metadata field URL