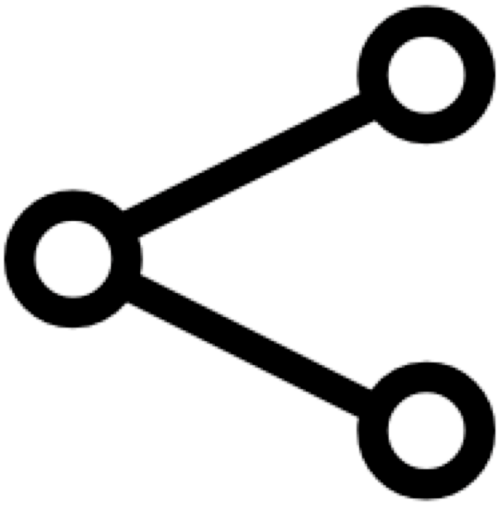


Skyword Publishing Feed

Integration Guide

skyword

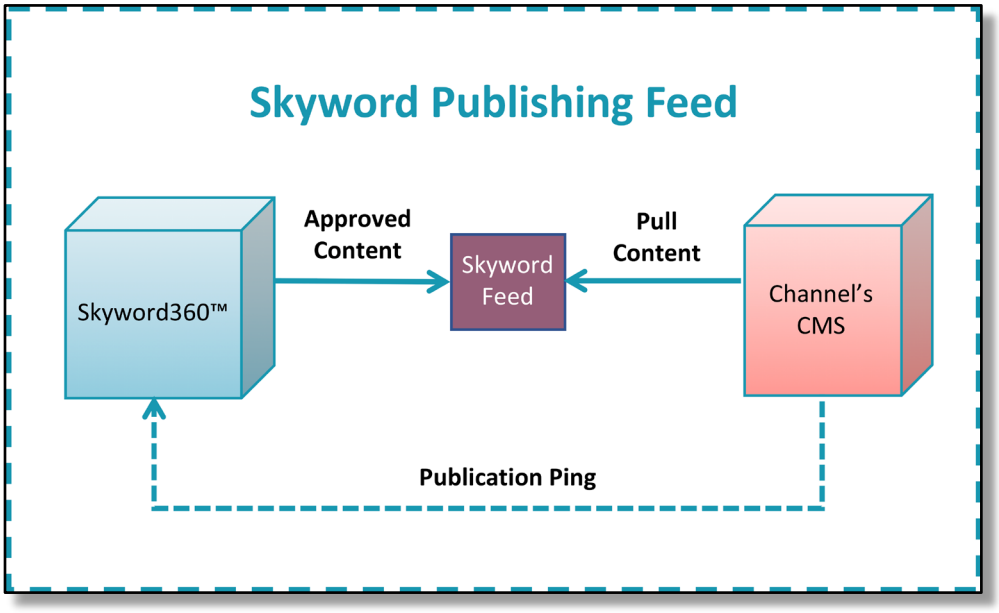
190716

Table of Contents

Skyword Inc., 38 Chauncy Street, 14th Floor, Boston, MA 02111

1 Introduction

The ***Skyword Publishing Feed*** is one of several integration routes for delivery of content from Skyword360™ to your content management system (CMS). With this method, the Skyword Integrations Team provides a protected uniform resource locator (URL) that outputs an XML or JSON feed of content approved in Skyword360.



Use the Publishing Feed to publish approved content to your CMS.

1.1 Feed Format Summary

The URL structure to retrieve the feed is of the following form.

```
https://api.skyword.com/feed?key=XXXXXXXXXXXX
```

- This URL is accessible via HTTP or HTTPS; the use of HTTPS, however, is strongly encouraged.
- The "key" URL parameter is a 20-digit secure password that is unique to every client and generated by the platform. Please contact Skyword for your API key or navigate to **Channel Settings** > **Delivery Method** in Skyword360.
- Since the key is in the URL path it is also encrypted as part of the SSL specification if you choose to use HTTPS.

- The content of the feed is an XML or JSON document that can include multiple content items.
- The XML or JSON format can be customized to a large degree depending on your data model or template of content.
- For additional security, the Skyword Integrations Team can restrict access for your API key to an IP address range you specify.

1.2 Implementation Tasks

Your tasks in integrating the Skyword Publishing Feed into your CMS are the following.

- Develop a process that routinely downloads and checks the URL above for new content. Recommended polling frequency is once every 10 minutes, but should not be more frequent than once per minute.
- Parse the contents and store any new content found in your CMS.
- Download and store any associated images or binary attachments in your CMS or file system.
- Optionally (but recommended), ping the Skyword servers to inform Skyword360 that you have published the content.

2 Skyword Publishing Feed Integration Kit for Java

Skyword provides a Java integration kit with code examples and a framework for downloading and parsing the XML stream. This kit contains freely reusable code to kick-start your integration development.

You can download the Java integration kit:

http://www.skyword.com/wp-content/uploads/2013/01/skyword-xml-api-java-1.0.zip

3 XML Format

Skyword can customize the XML format to accommodate your unique content type, depending on your unique page template and/or CMS needs. An example XML feed could resemble the following.

```
<?xml version="1.0" ?>
<entries>
  <entry>
     <action>create</action>
     <id>1234567890</id>
     <title><![CDATA[How to Lay a Tile Floor]]></title>
     <description>
     <![CDATA[A short description of the article (meta-description).]]>
     </description>
     <abstract><![CDATA[An abstract of the article.]]>
     </abstract>
     <body><![CDATA[This is the body in <strong>HTML</strong>]]>
     </body>
     <keywords><![CDATA[tile floor, grout]]></keywords>
     <image>
         <file>1234</file>
         <alt><![CDATA[This is the image alt text]]></alt>
     </image>
  </entry>
  <entry>
     <action>create</action>
     <id>1234567890</id>
     <title><![CDATA[How to Lay a Tile Floor]]></title>
     <description>
     <![CDATA[A short description of the article (meta-description).]]>
     </description>
     <abstract><![CDATA[An abstract of the article.]]>
     </abstract>
     <body><![CDATA[This is the body in <strong>HTML</strong>]]>
     </body>
     <keywords><![CDATA[tile floor, grout]]></keywords>
     <image>
         <file>1234</file>
         <alt><![CDATA[This is the image alt text]]></alt>
     </image>
  </entry>
</entries>
```

The format and schema of the XML closely follows the definition of your data model for the type of content you are receiving. For example, you may have fields specified for

“title”, “body”, “image”, “lead in”, “tags”, “part number”, and “city”. The collection of fields, their corresponding data types, and their validation requirements defines your content template.

> **Note:** You can also have multiple content types defined in Skyword360. All of your content is delivered in the same stream.

In addition to your content templates field information, Skyword360 also sends the following nodes for every content item:

- <id> — This is the Skyword360 identifier of the content.
- <action> — The action to perform in your CMS (create, update, or delete).

It is strongly recommended that you store in your CMS the Skyword360 ID of the article transmitted to the platform. This is useful when checking for duplicate content as well as future debugging.

You have control over what actions you may permit Skyword360 to take. While you must implement the “create” functionality, the “update” and “delete” functions are recommended but optional. The platform can be configured to permit only the actions you choose.

If you choose to permit updates, Skyword360 transmits the entire record and does not transmit the partial delta. You should replace all data elements in your CMS with the information in the XML feed entry.

4 Image Delivery/Binary Attachments

If your content type requires images or any form of file attachment, these must be downloaded separately. Binary data is not transmitted or encoded in the content XML stream. Instead a reference node is sent that contains the identifier of the file in Skyword360. For example, your XML stream may contain code similar to this snippet.

```
<image>
    <nodename>12345</nodename>
    <nodename_name>image.jpg</nodename_name>
    <nodename_url><![CDATA[https://api.skyword.com/file?key=123456ABCDE&file=
12345]]></nodename_url>
```

```
    <alt><![CDATA[image alt text]]></alt>
</image>
```

Where:

<nodename>: This tag must be renamed with the element name in your content template for image attachments (for example, <featured_image>). The value in this line is the attachment ID that you must use to download the binary information using this URL:

https://api.skyword.com/file?key=XXXXXXXXX&file=12345

<nodename_name>: The value in this line is the filename of the image file (for example, image.jpg).

<nodename_url>: The URL in this line is the link that you must use to download the image. The key value is your channel's API Key; the file value is the attachment ID.

Your API key must be sent along with the file identifier in order to download the data. You should retrieve this URL via an HTTP Get.

This URL returns the file as any normal HTTP image request would. The binary data is contained in the HTTP body and the mime type is transmitted in the HTTP header "Content-Type". For example:

```
Content-Type: image/jpeg
```

This file should be saved into your CMS or file system and is served through your own website. Skyword360 supports a wide range of validation criteria for files and images. You may specify the maximum file size, what mime types to allow, maximum image width/height, and/or cropped aspect ratio. Please contact the Skyword Integrations Team to configure these options.

Note: You may also need to associate or "attach" the downloaded file with the article/content in your CMS depending on your CMS content template.

4.1 Embedded Images

Embedded images are images that have been placed directly *inside* the HTML of the body of an article or content. These are embedded within the HTML via simple `<img>` tags.

You may choose whether or not to permit embedded images within your content. Typically, images are "attached" to the content and are not embedded directly within the HTML body of the content. However, you may choose to have your images embedded. By allowing images to be embedded you are giving more control to writers to place images inline with the content anywhere they choose. Images embedded in the HTML body of content may also reference external websites when entered by a writer or editor.

When allowing embedded images in your content, Skyword360 provides the following choices for processing them:

- No special processing. Image references in the HTML are simply sent along to your CMS as is.
- The platform can download and temporarily store the image, then transmit the binary data with the file identifier for you to download.

If you require Skyword360 to download these images, the HTML `<img>` in the article body is referenced as follows.

```
<img src="SKYWORD-FILE:12345">
```

You must parse the HTML of the article body, download the image via the file URL above, and replace all "file" references with the final URL of the image that is hosted on your server. It is up to your application to make this substitution since Skyword360 does not know the final URL of the image at the time the article feed is transmitted.

4.2 Image Metadata

Image metadata (such as alt text or source) can be communicated with the XML of the content. For example:

```
<file>12345</file>
<alt><![CDATA[This is the image alt text]]></alt>
```

```
<imageDescription>![CDATA[This is the image description]]</imageDescription>
```

Skyword360 supports a wide range of image metadata depending upon your content type.

5 Skyword Tracking Tag

Every piece of content transmitted by Skyword360 contains a small snippet of JavaScript code that is used by the platform to track performance and optionally detect publication. This tracking tag is typically added to the bottom of the `<body>` content. However, it can also be transmitted separately in its own XML node. The format of the tracking tag is similar to example displayed below:

```
<script src=http://tracking.skyword.com/tracking.js?contentId=12345></script>
```

The “contentId” URL parameter is unique for each and every piece of content transmitted by the platform.

6 Publication Acknowledgement/Detection

For every piece of content that you receive via the feed, Skyword360 must be informed of the final URL to which the content is published. There are two ways the platform can obtain this information:

- Client pings a Skyword URL that informs Skyword360 of publication
- Skyword360 auto-detects the URL when published

6.1 Publication Ping

Issue an HTTP GET call to the following URL:

https://api.skyword.com/publish?key=XXX&contentId=YYYY&url=ZZZZZ

- You must specify your API key
- The “contentId” parameter is the Skyword360 identifier sent in the feed as <id>
- The “url” parameter is the encoded, fully qualified URL of the content item.

When the client pings this URL, Skyword360 returns a success message or error message to the client. Upon notification of a successful publish, the platform *no longer* transmits the respective article or content.

6.2 Automated Publication Detection by Skyword360

The ***Skyword Tracking Tag*** can be used to inform Skyword360 that content has been published. Every article that the platform transmits has a special tracking tag embedded in the body HTML. As described above, this tracking tag is similar to a Google analytics tracking tag. Using this tracking tag, the platform can detect when an article gets initial public traffic and marks the content as published in the database.

Keep in mind the following caveats when letting Skyword360 detect publication:

- Skyword360 continues to transmit the same content (with the same id) until it detects that the article was successfully published. You must *ensure* that you do not create duplicated content.
- A human using a web browser must actually view the content in order for Skyword360 to detect the publication.
- You must provide a valid regex to match your public URL structure.

6.3 Regular Expression Matching

Skyword360 can distinguish between live public traffic and internal hits from your local CMS administration viewing. In order to enable this, you must provide the Skyword Integrations Team with a proper regex to validate the proper external public URL format. The goal is to provide the most specific regex that matches the public content URL and does not match any internal URL that might be initiated by someone reviewing the content through your CMS.

The simplest regular expression simply matches the public domain of your site:

```
^https?://(www.)?yoursite.com/
```

This expression matches any URL starting with http or https and is referencing yoursite.com. For example, this matches:

```
http://www.yoursite.com/
https://www.yoursite.com
http://yoursite.com/
https://yoursite.com/
```

A more specific (and preferred) regular expression would be:

```
^https?://(www.)?mysite.com/article/\w+.html
```

This matches any URL in the directory /article/ and ending in ".html". This example matches the following URLs:

```
http://www.mysite.com/article/this-is-my-new-article.html
http://www.mysite.com/article/this_is_my_new-article.html
http://www.mysite.com/article/This-my-new-99-article.html
```

Here are some other common regular expressions:

```
^https?://(www.)?mysite.com/article/[a-zA-Z][a-zA-Z]/[a-zA-Z]+/[a-zA-Z0-9-]+-id
^https?://(www.)?mysite.com/\w+.html
^https?://(www.)?mysite.com/\w+/\w+/\w+.html
^https?://(www.)?mysite.com/[a-zA-Z][a-zA-Z]+/\w+.html
```

See the following resources to learn more about regular expressions.

- http://www.regular-expressions.info/javascriptexample.html
- http://www.regular-expressions.info/tutorial.html
- http://en.wikipedia.org/wiki/Regular_expression
- http://regexlib.com/