



# XML API

Skyword Integration Guide

## Table of Contents

1	Introduction.....	1
2	Skyword XML API Integration Kit for Java .....	2
3	Test API Key .....	2
4	XML Format .....	2
5	Image Delivery/Binary Attachments .....	4
5.1	Embedded Images .....	5
5.2	Image Metadata.....	5
6	Skyword Tracking Tag.....	6
7	Publication Acknowledgement/Detection.....	6
7.1	Publication Ping .....	6
7.2	Automated Publication Detection by Skyword .....	7
7.2.1	Regular Expression Matching.....	7

# 1 Introduction

This document outlines an integration route for delivery of content from the Skyword Platform to your content management system (CMS). Skyword will provide you with a protected uniform resource locator (URL) that outputs an XML feed of content approved in the Skyword system. The URL structure to retrieve the XML feed is of the following form:

<https://api.skyword.com/feed?key=XXXXXXXXXXXXX>

- This URL is accessible via HTTP or HTTPS, however the use of HTTPS is strongly encouraged.
- The “key” URL parameter is a 20-digit secure password that is unique to every client and generated by Skyword. Please contact Skyword for your API key or navigate to **Program Settings -> Delivery Method** in the Skyword web application.
- 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 will be an XML document that can include *multiple* content items.
- The XML format can be customized to a large degree depending on your data model or template of content.
- For additional security, Skyword can restrict access for your API key to an IP address range you specify.

Your tasks in integrating the Skyword XML 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 XML 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 Skyword that you have published the content.

## 2 Skyword XML API 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 at:

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

## 3 Test API Key

Skyword provides a default API key that you may use for initial research and testing. The testing API key is API\_TEST\_KEY:

[https://api.skyword.com/feed?key=API\\_TEST\\_KEY](https://api.skyword.com/feed?key=API_TEST_KEY)

**Note:** The API test key is mainly for use *before* your program is set up in the Skyword Platform. Once your program is set up in Skyword, you should use your own unique API key issued by Skyword.

## 4 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>12345</file>
    <alt><![CDATA[This is the image alt text]]></alt>
  </image>
</entry>
</entries>
```

The format and schema of the XML will closely follow 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 the Skyword Platform. All of your content will be delivered in the same XML stream.

In addition to your content templates field information, Skyword will also send the following nodes for each and every content item:

- <id> — This is the Skyword 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 Skyword ID of the article transmitted to Skyword. This is useful when checking for duplicate content as well as future debugging.

You have full control over what actions you may permit Skyword to take. Skyword requires that you implement the “create” functionality. However, update and delete are optionally implemented. Skyword can be configured to permit only the actions you choose, but we do encourage permitting update and deletes.

If you choose to permit updates, Skyword will transmit 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.

## 5 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 `<file>` node is sent that contains the identifier of the file in the Skyword system. For example, your XML stream may contain:

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

The `<file>` node is a placeholder with the ID that you must use to download the binary information using this URL: <https://api.skyword.com/file?key=XXXXXXXXXX&file=12345>. 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. Skyword 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 Skyword 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.

## 5.1 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, Skyword provides the following choices when processing them:

- No special processing. Image references in the HTML are simply sent along to your CMS as is.
- Skyword can download and store temporarily the image on its servers and will transmit to you the binary data with the file identifier for you to download.

If you require Skyword to download these images, the HTML `<img>` in the article body will be referenced as follows:

```

```

It is required that you 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 Skyword does not know the final URL of the image at the time the article feed is transmitted.

## 5.2 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>
```

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

## 6 Skyword Tracking Tag

Every piece of content transmitted by Skyword contains a small snippet of JavaScript code that is used by Skyword 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 Skyword.

## 7 Publication Acknowledgement/Detection

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

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

### 7.1 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 Skyword identifier sent in the XML feed as `<id>`
- The “url” parameter is the encoded, fully qualified URL of the content item.



When the client pings this URL, Skyword will return a success message or error message to the client. Upon notification of a successful publish Skyword will *no longer* transmit the respective article or content.

## 7.2 Automated Publication Detection by Skyword

The Skyword tracking tag can be used to inform Skyword that content has been published. Every article that Skyword 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, Skyword can detect when an article gets initial public traffic and will mark the content as published in the Skyword database.

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

- Skyword will continue to transmit the same content (with the same id) until it can detect 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 Skyword to detect the publication.
- You must provide Skyword a valid regex to match your public URL structure.

### 7.2.1 Regular Expression Matching

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

To learn more about regular expressions, we recommend using these resources:

- <http://www.regular-expressions.info/javascriptexample.html>
- <http://www.regular-expressions.info/tutorial.html>
- [http://en.wikipedia.org/wiki/Regular\\_expression](http://en.wikipedia.org/wiki/Regular_expression)
- <http://regexlib.com/>