



# Adobe Experience Manager

Draft Integration Guide

## Table of Contents

1	Introduction.....	1
2	Direct Connect.....	1
2.1	Security Configuration Settings.....	2
2.2	Content Template Requirements .....	2
2.2.1	JCR Node Path .....	2
2.2.2	JCR Node Structure and Properties .....	3
2.2.3	Skyword Content Template Items .....	4
2.3	Publication URL .....	5
2.4	API Testing Tool.....	5
3	XML Feed.....	6
3.1	How it Works .....	6
3.1.1	API Key .....	6
3.2	Skyword Java Integration Kit for AEM.....	6
3.2.1	Getting Started .....	6
3.2.2	Customizing SkywordAdobe.java .....	7
3.2.3	Compiling the Kit .....	8
3.2.4	Scheduling a Cron Job .....	8
4	Tracking Tag.....	8

# 1 Introduction

The Skyword Platform integrates with the Adobe Experience Manager (AEM) content management system, previously known as CQ5. Contributors, editors, program managers, and reviewers can use the Skyword Platform to create, edit, and review content. After content is approved, the Skyword Platform can communicate directly with AEM to publish the content.

There are two ways to integrate AEM with Skyword:

- **Direct Connect:** When content is approved for release, the Skyword Platform will automatically push it to AEM using the JCR (Java Content Repository) API. AEM fully implements the JCR standard, which makes cross platform integrations easier. This option is easier, but may require changes to your organization's firewall configuration.
- **XML Feed:** A batch/cron program will download approved content from the Skyword Platform via an XML feed from a URL that Skyword will provide. Skyword provides a Java-based development kit for the integration, which will kick-start much of the implementation. However, it is the responsibility of your organization to customize this to your specific AEM implementation and install it within your infrastructure.

This document describes both of the Skyword integration options for AEM.

# 2 Direct Connect

The Skyword Platform can use the JCR protocol to communicate directly with your AEM instance. This option requires little to no technical work on your part but requires that you permit Skyword to access your AEM instance. You must provide Skyword with the following information regarding your AEM instance:

- **Publish Instance URL:** This is the endpoint of your site, and the URL where your site is viewable by the public. By default, the publish instance will run on port 4503.
- **Author Instance URL:** This is the administrative URL for your AEM instance and is where your internal staff typically interacts with AEM content management. This URL requires authentication credentials. By default, the author instance will run on port 4502.

- **Authentication Credentials:** An AEM username and password is required for authentication. This username should have sufficient access rights to create, update, or delete content that the Skyword Platform will provide.
- **Content Template:** You will need to inform Skyword of your Adobe/JCR content template structure. All content produced within Skyword must match your expected content template and layout patterns. See Section 2.2 for more information.

## 2.1 Security Configuration Settings

Direct connect provides the following security configuration settings:

- **IP Address Restriction:** You should configure an IP address restriction within your firewall to allow *inbound* HTTP requests to your Author instance URL *only* from Skyword's IP range, which is 74.201.33.\*.
- **HTTPS:** You may optionally configure your server to operate on SSL.

We strongly recommend that you configure both IP address restriction and SSL encryption. Doing so will help to insure that access to your AEM Author instance is protected. All authentication credentials stored within the Skyword Platform are encrypted.

## 2.2 Content Template Requirements

For the Skyword Platform to transmit and store content in your AEM instance, you will need to inform Skyword of your Adobe Content Repository mappings. Content repository mappings should include the full root JCR path and node structure/properties.

### 2.2.1 JCR Node Path

Skyword will transmit and store all content relative to a chosen JCR root path. To obtain that path, log in to your CRXDE Lite administration console and navigate to the root JCR path where you will store content from Skyword, as shown in Figure 1.

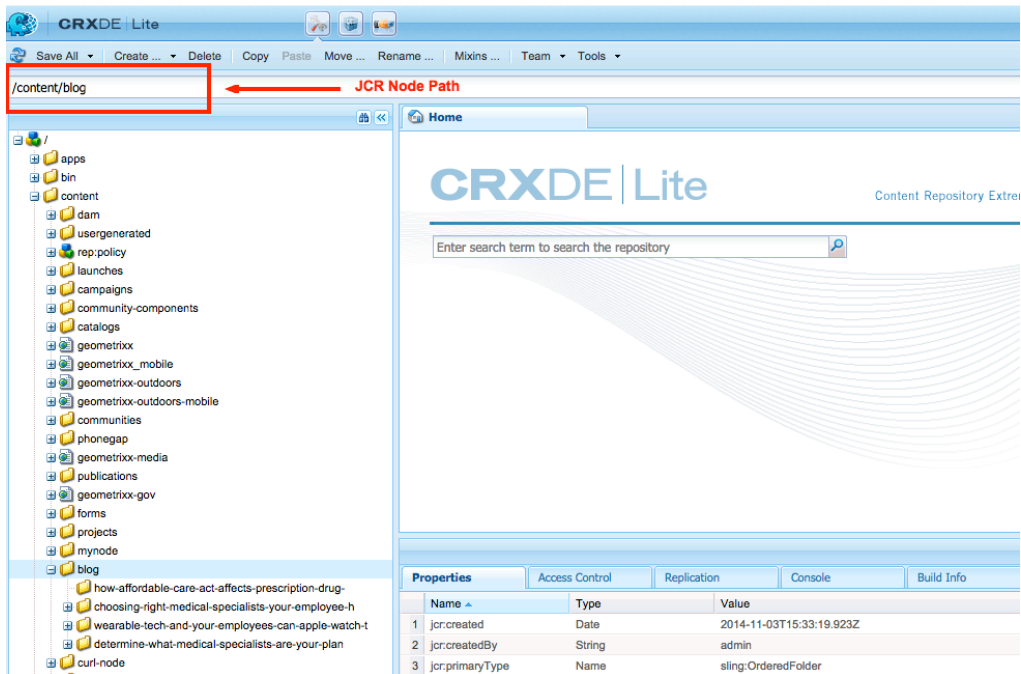


Figure 1. Locate the JCR node path.

## 2.2.2 JCR Node Structure and Properties

Content produced by the Skyword Platform can have many different rich media components, similar to AEM. This essentially defines your data model of the content you want to manage within the Skyword Platform. For example, content fields may include a title, lead in, large body, multiple attached images, and metadata descriptions.

The Skyword Platform supports a wide variety of content template fields and UX configurations, including:

- Single line text
- Large body HTML text
- Dropdowns/Tagging based upon fixed or expandable taxonomies
- Multi-selects
- Image upload
- File attachments (PDFs, Word documents, etc.)
- Checkboxes

In addition to the wide variety of UX control type fields, Skyword can also enforce mandatory requirements, cardinality, and REGEXP like validations.

You will need to inform Skyword of your content template node structure and properties. The easiest way to inform Skyword of your node structure is to take a screenshot within CRXDE Lite, as shown in Figure 2. Send that screenshot to your Skyword Program Manager and Skyword will configure your template within the Skyword Platform.

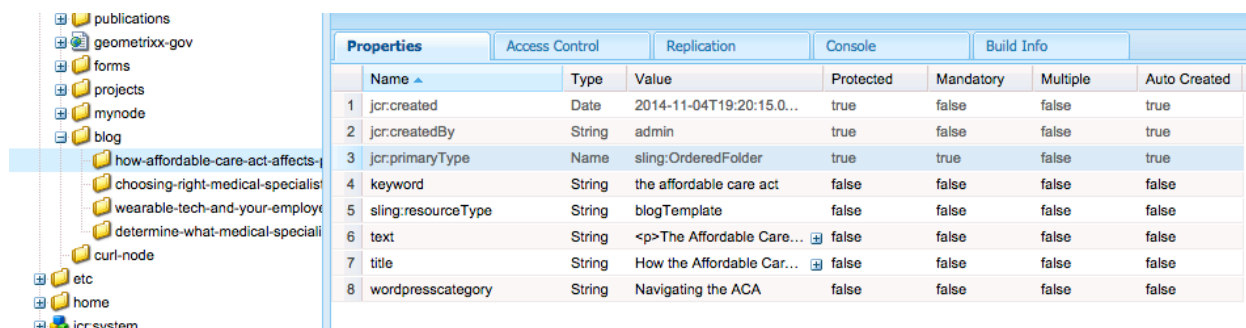


Figure 2. Take a screenshot within CRXDE Lite.

### 2.2.3 Skyword Content Template Items

Skyword may additionally transmit the following data elements, or you may choose to use the Skyword standard content template.

**Note:** All data is encoded in UTF-8.

Name	Description	Required
Title	The title of the article. Maximum length is 255 characters. Simple plaintext characters only, no HTML.	Yes
ID	The Skyword internal identifier for the article. This will be a plaintext 20-digit numeric. This is the primary way of identifying a Skyword article.	No
Content Body	The full content body of the article. Maximum length is 65,535 characters. This element will contain HTML markup in escaped entity encoding format.	Yes
Author Name	The byline of the writer. Maximum length is 255 characters. Simple plaintext characters only.	No

Name	Description	Required
Category	The category name of the article. All articles are classified into one and only one client defined category. Maximum length is 60 characters. Plaintext characters only.	No
Title Tag	The HTML <title> of the article. This can be different than the formal title of the article in order to more fully SEO optimize the content.	No
Meta Description	The HTML meta description (<meta name="description" content="">)	No
Meta Keywords	The HTML meta keywords (meta name="keyword" content="">)	No
Keyword	The keyword used to SEO optimize the article content.	No

## 2.3 Publication URL

The Skyword Platform will automatically generate the final public publication URL for the content when it transmitted to your AEM instance. The public URL will be based on the publication URL you specified above, added with the following in order:

- The JCR node path (see Section 2.2.1)
- With year/month in /YYYY/MM form
- With a slugified value of the title of the content
- With the postfix ".html"

For example: <http://www.yoursite.com/content/blogs/2014/12/this-the-title.html>

## 2.4 API Testing Tool

Skyword maintains a testing and integration tool to assist you in developing your server's portion of the API implementation. This testing tool is accessed through the Skyword interface at <https://create.skyword.com>. We recommend that you publish to a staging server when using this tool. You will need to contact Skyword for setting up an account to access this tool.

## 3 XML Feed

If your organization decides not to use the Direct Connection integration method, you can use an XML feed to integrate Skyword with AEM. If you choose this integration option, Skyword will provide your organization with a customizable Java-based implementation development kit, and a URL that outputs an XML feed of content approved in the Skyword system. You are responsible for customizing the Java kit for your organization's AEM instance.

### 3.1 How it Works

Skyword will provide your organization with a URL to retrieve the XML feed. The content of the feed will be an XML document that can include multiple approved content items. The XML format can be customized to a large degree depending on your content template. The structure of the URL is as follows:

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

This URL is accessible via HTTPS.

#### 3.1.1 API Key

The “key” parameter is a 20-digit API key that is uniquely generated for every client. Please contact Skyword to obtain an API key. For additional security, Skyword can restrict access for your API key to an IP address range that you specify.

### 3.2 Skyword Java Integration Kit for AEM

Skyword provides a Java integration development kit for AEM that you can use to integrate AEM with the Skyword Platform. This kit contains code examples and a framework for downloading and parsing the XML stream for AEM. Please contact Skyword for the download link.

#### 3.2.1 Getting Started

Once you have downloaded the Skyword Java integration kit for AEM, you can unzip the archive on a server of your choice—it does not need to be the machine that hosts your AEM instance. You will need to customize the code provided and build the final



implementation Java application. There are several files in the Java kit, but you should only have to modify the settings in SkywordAdobe.java.

### 3.2.2 Customizing SkywordAdobe.java

Once you receive the Java integration kit for AEM, you'll need to customize the SkywordAdobe.java file. This file contains settings that control how content items are stored in AEM. Customize the following settings for your AEM instance:

- **key:** The API key that Skyword generated for your organization.
- **nodePathBase:** The JCR node path. See Section 2.2.1 of this document for more information.
- **publishDomain:** The URL of the endpoint of your site and where your site is publically viewable. By default, the publish instance will run on port 4503.
- **authorURL:** The URL of the AEM author instance, which is used to make changes to your site. This instance requires authentication credentials. By default, the author instance will run on port 4502.
- **Authentication Credentials:** An AEM username and password is required for authentication. This username should have sufficient access rights to create, update, or delete content.

Note that by default, the code stores content with the `/year/month/title_slug` path. You may need to modify this code for your AEM instance.

```
String nodePathBase = "content/geometrixx-  
outdoors/en/company/unlimited-blog";  
ArrayList<String> nodePathAppend = new  
ArrayList<String>(Arrays.asList("{year}", "{month}"));  
  
String templateName = "/libs/social/blog/templates/page";  
String slingPageResourceType = "geometrixx-  
outdoors/components/social/journal/page-company";  
String slingFoundationType = "foundation/components/";  
String slingEntryType = "social/blog/components/entrytext";  
String publishDomain = "http://123.456.78.9:4503";  
String authorURL = "http://123.456.78.9:4502/crx/server";  
  
String username = "admin";  
String password = "admin";
```

You will also need to customize the following section for your AEM instance. This example makes use of the *parsys* (*paragraph system*) component. If you use another

component, for example the `textImage` component, you will need to modify this code to match the requirements of that component.

```
Node testParNode;

try {
    testParNode = testContentNode.getNode("par");
} catch (Exception e) {
    testParNode = testContentNode.addNode("par", "nt:unstructured");
}

testParNode.setProperty("sling:resourceType", slingFoundationType +
    "parsys");

Node testEntryNode;
try {
    testEntryNode = testParNode.getNode("entry");
} catch (Exception e) {
    testEntryNode = testParNode.addNode("entry", "nt:unstructured");
}

testEntryNode.setProperty("sling:resourceType", slingEntryType);
testEntryNode.setProperty("text", (String)
    articleContents.get("body"));
```

### 3.2.3 Compiling the Kit

After you configure the settings in `SkywordAdobe.java`, you'll need to compile the kit. You can do this by navigating to the kit directory, and then typing `ant all` at the command line.

### 3.2.4 Scheduling a Cron Job

The last step is scheduling a cron job. To initiate a pull request, the cron job should run the `batch.sh` file. We recommend that you schedule the cron job to run at least once an hour.

## 4 Tracking Tag

Skyword uses an analytics tracking tag to record performance information on published content. Skyword uses the tracking information for real-time performance reporting, contributor compensation, trend detection enhancement, and fraud prevention. For both of the integration options for AEM, the Skyword Platform includes the tracking tag in the body of the content item. For more information about the tracking tag, see this document: <https://help.skyword.com/hc/en-us/articles/200565144>.